VIDO: A Robust and Consistent Monocular Visual-Inertial-Depth Odometry

Yuanxi Gao, Jing Yuan, *Member, IEEE*, Jingqi Jiang, Qinxuan Sun, and Xuebo Zhang, *Senior Member, IEEE*

***Abstract*— Multi-sensor fusion is a mainstream method for localization of unmanned systems. How to achieve 6-degrees of freedom (DOF) pose estimation of the system is challenging in GPS-denied environments. Although map-aided localization methods normally perform well on intelligent transportation systems, prior maps are unavailable in some GPS-denied scenes (e.g., dense forests, tunnels, and underground parking lots). In this paper, we present a robust and consistent monocular visual-inertial-depth odometry (VIDO) to perform 6-DOF pose estimation without the need of prior information. The system contains a visual-inertial subsystem (VIS) based on tightly coupled optimization in a sliding window and a depth subsystem (DS) based on the iterative closest point (ICP) estimation using 3D point clouds obtained by a LiDAR or depth camera. The uncertainties of the estimation results in VIS and DS are rigorously calculated to consider measurement noises of the sensors. The obtained uncertainty estimates are fed into a covariance intersection (CI) filter for pose fusion, and the fused pose is further refined in the mapping process. We perform experiments on public datasets, as well as in various real-world outdoor and indoor scenes to verify the performance on localization and mapping in urban areas with buildings and cars, off-road environments with rugged terrains, as well as indoor structured environments. The results show that the proposed method can provide both a robust 6-DOF pose estimate and a precise 3D map for fully autonomous navigation in different scenes without a prior map, which presents an attractive complement to map-aided automated driving.**

***Index Terms*— Covariance propagation, depth subsystem, multi-sensor fusion, visual-inertial subsystem.**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) provides a solution to navigation of intelligent transportation systems in unknown environments, such as automated driving of unmanned ground vehicles (UGVs) in urban areas and autonomous exploration of robots in off-road scenes when a prior map and the global navigation satellite system (GNSS) are unavailable [1], [2], [3]. A prerequisite of automated driving and autonomous exploration is the accurate pose estimation of the vehicles and robots. In [4], [5], and [6], assuming a flat ground, 3-degrees of freedom (DOF) pose was estimated for the vehicle in urban road environments. However, in large-scale urban environments, the road surface is inevitably rough. Up and down slopes always exist in urban environments. Thus, 3-DOF pose estimation is insufficient for many realistic urban applications. For motorway scenes, the studies in [6] and [7] proposed methods of pose estimation for UGVs based on semantic information. Nevertheless, scene-specific semantic information lacks generalization in other environments, especially in off-road environments. In off-road areas, the ground is also quite bumpy. Therefore, 6-DOF pose estimation is necessary for navigation of UGVs. The work in [8] proposed a multi-camera-based visual SLAM for off-road environments with sparse texture. However, visual SLAM is easily affected by illumination change. In [9], a tightly coupled LiDAR-inertial measurement unit (IMU) SLAM method was proposed for UGVs in off-road areas. Although the tightly coupled framework ensures the accuracy of localization, the robustness of this method is poor in structure-less scenes. The study in [10] proposed a GNSS-aided LiDAR SLAM system for navigation in off-road scenes. However, GNSS satellite signals are blocked in dense forests, mountains, tunnels, and underground parking lots. Specifically, in forests and mountains, UGVs are often used to perform tasks such as transportation and inspection. In these cases, GNSS-based localization is likely to fail because huge trees and mountains block the reception of satellite signals. In recent years, more and more large cities in the world have relieved the urban traffic pressure through underground tunnels. Autonomous localization without GNSS and additional communication devices is challenging for UGVs in underground tunnels. Likewise, large railway stations are generally equipped with large-scale underground parking lots. Localization in such environments with similar structures and appearances is difficult. In addition, in inner-city and urban canyons, multipath propagation and shadowing effects commonly make GNSS unreliable. Despite the lack of the GNSS information in the above scenes, SLAM techniques using on-board sensors can perform 6-DOF pose estimation, which provide a feasible solution for localization and navigation of vehicles. On the other hand, map-aided SLAM is a typical localization technique for autonomous driving [11], [12], [13]. For instance, the work in [13] proposed a tightly-coupled monocular map-matching localization

Manuscript received 4 October 2021; revised 17 April 2022 and 2 September 2022; accepted 11 November 2022. This work was supported in part by the Natural Science Foundation of China under Grant U21A20486, Grant 62073178, and Grant 61873327; in part by the Tianjin Science Fund for Distinguished Young Scholars under Grant 20JCJQJC00140; in part by the Major Basic Research Projects of the Natural Science Foundation of Shandong Province under Grant ZR2019ZD07; and in part by the Tianjin Natural Science Foundation under Grant 20JCYBJC01470. The Associate Editor for this article was Z. He. *(Corresponding author: Jing Yuan.)*

Yuanxi Gao, Jing Yuan, Qinxuan Sun, and Xuebo Zhang are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China (e-mail: nkyuanjing@gmail.com).

Jingqi Jiang is with Meituan, Beijing 100102, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2022.3226719, provided by the authors.

Digital Object Identifier 10.1109/TITS.2022.3226719

method for autonomous vehicles, which estimated the vehicle pose with an abundance of visual features and multi-frame high-definition (HD) map landmarks. Although map-aided SLAM can achieve high-precision localization in most of the cases, maps need to be constructed in advance, which limits the application of map-aided SLAM methods in unknown environments. In addition, as mentioned in [14], in the place that has a limited number of landmarks or has dramatic changes, map-aided SLAM methods could make false decisions. Moreover, challenges of managing the large size data of HD maps and updating the HD map need to be considered in real-time localization of autonomous vehicles. Hence, the robust 6-DOF pose estimation utilizing the on-board sensors, such as the monocular camera, IMU and depth sensor is required for successful navigation in urban environments and off-road scenes.

Specifically, in the monocular vision based localization, visual features are triangulated through consecutive observations to estimate the motion of the robots or vehicles [15], [16], [17]. However, visual features in the monocular vision-based localization largely depend on the texture information of the environment. Combination of the monocular visual system and the IMU yields a visual-inertial navigation system (VINS) [18], [19], [20], which helps the monocular visual system to directly acquire the scale as well as the roll and pitch angles. However, the visual-inertial-based odometry is unreliable under long-term motion or in the environment lacking stable visual features. On the other hand, depth sensors (such as a depth camera and 3D LiDAR) can directly obtain the structural information of the environment, and are not dependent on the illumination and texture. Hence, complementary characteristics are shown between monocular visual-inertial and depth sensors. The combination of them provides a possibility for the accurate and robust localization and mapping of robots or vehicles in challenging scenes.

The fusion methods of visual-inertial information and depth point cloud are different. For example, in some studies on fusion of the RGBD camera and IMU [21], [22], the depth value of each pixel in the image is directly extracted from the point cloud, which can omit the step of triangulation. Although these methods are simple, the structural information of the point cloud was not used. LOAM [23] is the most widely used LiDAR odometery, which extracted edge points and planar points from the point cloud, and calculated the pose of the vehicle by scan matching. In the works of [24] and [25], the visual (-inertial) odometry was used to provide an initial value for scan matching of the point cloud. In these frameworks, the visual information was only used to initialize the point cloud registration of the LiDAR, while the feature association between the camera and LiDAR was not fully utilized. In recent years, some visual-inertial-LiDAR tightly coupled fusion frameworks have been proposed. LIC-Fusion [26], [27] combined IMU measurements, visual features, and LiDAR features within the multi-state constraint Kalman filter (MSCKF [19]) framework. Nevertheless, it is difficult to overcome the effects of linearization errors on the estimation result. LVI-SAM [28] coupled the visual-inertial system and LIO-SAM [29]-based LiDAR-inertial system using factor graphs. Although this optimization-based method can achieve satisfying accuracy, high computational cost and strong dependence on hardware are common problems. In addition, the accuracy and robustness of the tightly coupled framework may be largely affected once one of the sensors fails.

In recent years, nonlinear stochastic filters [30], [31], [32], [33], [34] have been successfully applied to the pose estimation of UGVs. The nonlinear stochastic filters focused on how to design a state observer to make the localization error of the robots globally asymptotically stable. In these studies, the pose estimation of the robot was achieved by the developed error state observer and theoretical analysis was also presented. However, they excessively simplified the environment perception problem in SLAM. Specifically, they did not take the issue of feature estimation into account. Instead, they assumed that positions of all landmarks/features are known. As a result, map building cannot be involved in the navigation process of the robot. For instance, in [30], [32], [33], and [34], landmarks needed to be manually marked in advance. On one hand, manual marking is obviously not suitable for practical applications in automated driving and autonomous exploration. On the other hand, the number of these landmarks is too small, such that they cannot fully represent the environmental map.

In this paper, we propose a robust and consistent monocular visual-inertial-depth odometry (VIDO) for UGVs. The monocular visual-inertial information and the depth point cloud are used to perform the pose estimation, respectively, and then a covariance intersection (CI) filter-based fusion framework is presented to fuse the estimation results. Finally, the scan-to-map registration is used to achieve more accurate and robust localization of UGVs. VIDO can achieve the accurate and robust 6-DOF pose estimation for navigation of UGVs on complex terrains. VIDO can be applied to both on-road and off-road environments under GPS-denied conditions, including self-driving vehicles in the city, rescue robots in the field, weeding robots in farming, and reconnaissance robots in the military area. In addition to localization in the GPS-denied environments, VIDO provides an alternative way to build a 3D map for navigation of UGVs. Moreover, even if the GNSS or a prior HD map is available, VIDO can be combined with GNSS-based or HD map-based navigation to achieve higher-precision localization and to address uncertainty. Therefore, VIDO potentially provides a common SLAM technology module for navigation of UGVs. Besides the different applications of UGVs, for logistics transportation of unmanned aerial vehicles (UAVs), the accurate 6-DOF pose estimation is extremely important for navigation and flight control in large-scale environments. When VIDO is applied in specific application scenes, it can be combined with other techniques, such as lane detection [35], pedestrian and vehicle detection [36], occlusion removal [37], path planning and motion control [38]. In a word, VIDO can serve as an essential component in a complete intelligent transportation system for UGVs, UAVs and even air-ground cooperation. The main contributions of this work are as follows:

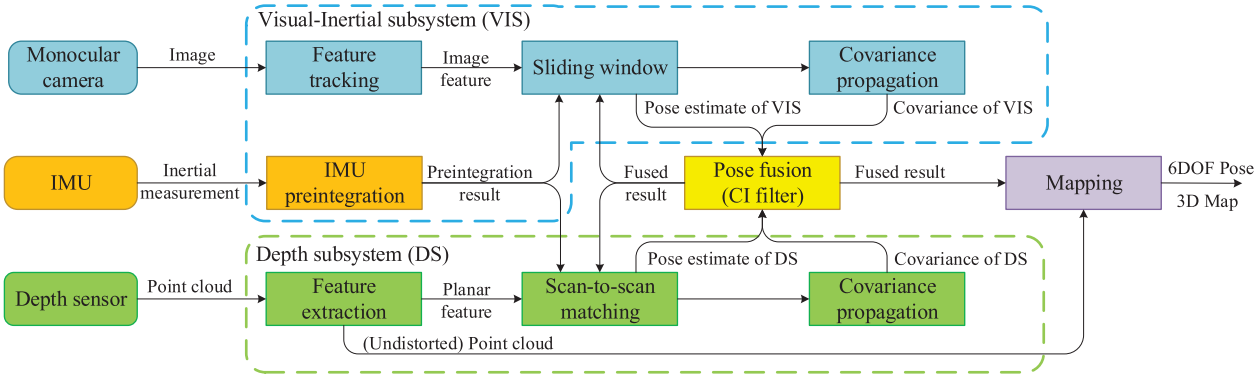


Fig. 1. Block diagram of VIDO. VIS and DS process the measurements from the monocular camera, IMU, and depth sensor, respectively, and estimate the pose of the UGV based on an optimization-based method. After deriving the closed-form covariance matrices, the pose fusion based on the CI filter is performed. The fused information is returned to VIS and DS as the prior information of the next iteration and matching. Finally, the fused result participates in the mapping process.

1) We rigorously derive closed-form covariance matrices for pose estimation results of visual-inertial subsystem (VIS) and depth subsystem (DS) in the optimization-based framework. In particular, to the best of our knowledge, this paper is the first attempt to compute a closed-form covariance matrix for visual-inertial SLAM based on the VINS-Mono framework, which provides an important supplement for VINS-Mono. With the covariance matrices, accuracy of the localization results of the two subsystems can be evaluated online, which offers a prerequisite for optimal and consistent fusion of the pose estimates.
2) We propose a loosely coupled visual-inertial-depth localization framework, which contains two subsystems, i.e., VIS and DS. On one hand, two subsystems can be fused by the CI filter to yield a consistent and optimal pose estimate of the UGV. On the other hand, two subsystems can work separately, such that VIDO is robust against the case that each of the two subsystems fails in challenging environments.

II. System Overview

The structure of VIDO is shown in Fig. 1. VIS processes images and IMU measurements and DS processes point clouds from the depth sensor. VIS extracts and tracks visual features in the image, and optimizes the joint residuals of the visual reprojection and IMU preintegration in a sliding window. In DS, planar features are extracted from point clouds for scan-to-scan matching. The point-to-plane iterative closest point (ICP) is used to optimize the pose, which uses the IMU preintegration as the initial value in iteration. If the point cloud is obtained from 3D LiDARs, the system uses the IMU measurements to remove the point cloud distortion. In both VIS and DS subsystems, the pose uncertainties (covariance matrices) are estimated in closed forms by propagating the uncertainties of the sensor's measurement. Based on the resultant uncertainties, pose estimates of VIS and DS are fused via the CI filter [39], [40] to yield a consistent and optimal localization result. The fused pose is returned to VIS and DS as the prior information of the next iteration and matching, respectively. In the mapping module, the system selects key-frame point clouds. And then, key-frame point clouds are matched with the local map, and added into the map.

For a clear description of the proposed method, we define the notations used in this paper. The world frame is represented by $(\cdot)^w$, and its z-axis is aligned with the direction of gravity. The body frame is $(\cdot)^b$, which is the same as the IMU frame. $(\cdot)^c$ is the camera frame and $(\cdot)^d$ is the depth sensor frame. The transformation from the frame B to the frame A is denoted by $\mathbf{T}_B^A = \left[\mathbf{R}_B^A | \mathbf{p}_B^A\right] \in \mathrm{SE}(3)$, where $\mathbf{p}_B^A \in \mathbb{R}^3$ represents the translation and $\mathbf{R}_B^A \in \mathrm{SO}(3)$ represents the rotation matrix. The rotation can also be represented by a rotation vector $\boldsymbol{\phi}_B^A \in \mathbb{R}^3$ and a Hamilton quaternion $\mathbf{q}_B^A$. The mapping with a rotation vector and a rotation matrix are exponential map and logarithm map [41]:

$$\begin{aligned} &\mathrm{Exp}: \mathbb{R}^3 \to \mathrm{SO}(3); \phi \mapsto \exp\left(\phi^\wedge\right) \\ &\mathrm{Log}: \mathrm{SO}(3) \to \mathbb{R}^3; \mathbf{R} \mapsto \log(\mathbf{R})^\vee \end{aligned} \tag{1}$$

where $(\cdot)^\wedge$ is defined as a mapping from a vector in $\mathbb{R}^3$ to a skew symmetric matrix, while $(\cdot)^\vee$ is the inverse function of $(\cdot)^\wedge$. In addition,

$$\exp\left(\phi^\wedge\right) = \mathbf{I}_{3\times 3} + \frac{\sin\left(\|\phi\|\right)}{\|\phi\|}\phi^\wedge + \frac{1-\cos\left(\|\phi\|\right)}{\|\phi\|^2}\left(\phi^\wedge\right)^2 \tag{2}$$

$$\log\left(\mathbf{R}\right) = \frac{\varphi \cdot \left(\mathbf{R} - \mathbf{R}^T\right)}{2\sin\left(\varphi\right)}, \ \varphi = \cos^{-1}\left(\frac{\mathrm{tr}\left(\mathbf{R}\right)-1}{2}\right). \tag{3}$$

When $\|\phi\| \approx 0$,

$$\exp\left(\phi^\wedge\right) \approx \mathbf{I}_{3\times 3} + \phi^\wedge. \tag{4}$$

It should be noted that when $\|\phi\| \approx 0$, i.e., ϕ is infinitesimal rotation, the exponential map can be approximated by (4). In this paper, this approximation is only used for deriving the Jacobian w.r.t. rotation, where the limit operation is involved when rotation tends to $\mathbf{0}_{3\times 1}$. In fact, at this time, the second term in the right side of (2) tends to $\phi^\wedge$ and the third term, i.e., the second order term, tends to $\frac{1}{2}\left(\phi^\wedge\right)^2$. Because ϕ tends to $\mathbf{0}_{3\times 1}$, $\phi^\wedge$ tends to $\mathbf{0}_{3\times 3}$. Hence, the second order term of (2) can be ignored. In addition, by ordinary series expansion of the

exponential map and Rodrigues' rotation formula, the rotation matrix $\mathbf{R} = \mathbf{I}_{3\times3}$ at $\phi = \mathbf{0}_{3\times1}$. When ϕ is not infinitesimal rotation, we do not approximate the exponential map as in (4).

In addition, variables with $\hat{}$ and $\check{}$, e.g., $\hat{\mathbf{T}}_B^A$ and $\check{\mathbf{T}}_B^A$, are used to indicate the estimation results in VIS and DS, respectively. The variable with $\bar{}$, e.g., $\bar{\mathbf{T}}_B^A$, represents the fusion result of VIS and DS. We assume that both the intrinsic and extrinsic parameters of the sensors are given through calibration and the sensors are time-synchronized.

III. Robust and Consistent Monocular Visual-Inertial-Depth Odometry

A. Visual-Inertial Subsystem and Its Uncertainty Estimation

Shi-Tomasi features are detected in the visual image and tracked by the Kanade-Lucas-Tomasi sparse optical flow algorithm [42]. We preintegrate IMU measurements between two consecutive visual key-frames b_i and b_{i+1}, yielding the preintegrated result $\mathbf{z}_{b_{i+1}}^{b_i} \in \mathbb{R}^9$, including translation $\mathbf{p}_{b_{i+1}}^{b_i} \in \mathbb{R}^3$, rotation $\boldsymbol{\phi}_{b_{i+1}}^{b_i} \in \mathbb{R}^3$ and linear velocity $\mathbf{v}_{b_{i+1}}^{b_i} \in \mathbb{R}^3$, as well as the covariance matrix $\Sigma_{b_{i+1}}^{b_i} \in \mathbb{R}^{9\times9}$ of $\mathbf{z}_{b_{i+1}}^{b_i}$. Details about the IMU preintegration can be found in Appendix I. VIS performs bundle adjustment (BA) in a sliding window, in which a total of n body states and m features are optimized. The system state vector considered in the sliding window is

$$\begin{aligned}\hat{\mathbf{x}}_{vis} &= \left[\hat{\mathbf{x}}_0^T, \hat{\mathbf{x}}_1^T, \cdots, \hat{\mathbf{x}}_n^T, \hat{\lambda}_0, \hat{\lambda}_1, \cdots, \hat{\lambda}_m\right]^T \\ \hat{\mathbf{x}}_k &= \left[\hat{\boldsymbol{\xi}}_{b_k}^{w\,T}, \hat{\mathbf{v}}_{b_k}^{w\,T}, \hat{\mathbf{b}}_{a_k}^T, \hat{\mathbf{b}}_{g_k}^T\right]^T \in \mathbb{R}^{15}, k \in \{0, n\} \\ \hat{\boldsymbol{\xi}}_{b_k}^{w} &= \left[\hat{\mathbf{p}}_{b_k}^{w\,T}, \hat{\boldsymbol{\phi}}_{b_k}^{w\,T}\right]^T \in \mathbb{R}^6,\end{aligned}$$

where $\hat{\mathbf{x}}_k$ is the body state when the k-th image is obtained. It includes the position $\hat{\mathbf{p}}_{b_k}^w \in \mathbb{R}^3$, orientation $\hat{\boldsymbol{\phi}}_{b_k}^w \in \mathbb{R}^3$, and velocity $\hat{\mathbf{v}}_{b_k}^w \in \mathbb{R}^3$ in the world frame, as well as the biases $\hat{\mathbf{b}}_{a_k} \in \mathbb{R}^3$ and $\hat{\mathbf{b}}_{g_k} \in \mathbb{R}^3$ of the acceleration and gyroscope. The IMU biases are modeled as a random walk. $\hat{\lambda}_l$ is the inverse depth of the feature l when it was first observed. Similar to the front-end of VINS-Mono, the visual-inertial optimization in the VIS is to minimize the sum of the prior and the Mahalanobis norm of all measurement residuals

$$\min_{\hat{\mathbf{x}}_{vis}} \left\{ \left\| \mathbf{r}_p - \mathbf{H}_p \hat{\mathbf{x}}_{vis} \right\|^2 + \sum_{k\in\mathbb{B}} \left\| \mathbf{r}_b(\mathbf{z}_{b_{k+1}}^{b_k}, \hat{\mathbf{x}}_{vis}) \right\|_{\Omega_{imu}^k}^2 + \sum_{(l,j)\in\mathbb{C}} \rho_k\left(\left\| \mathbf{r}_c(\mathbf{z}_l^{c_j}, \hat{\mathbf{x}}_{vis}) \right\|_{\Omega_{cam}^{l,j}}^2 \right) \right\}. \tag{5}$$

where $\{\mathbf{r}_p, \mathbf{H}_p\}$ is the prior information from marginalization. $\mathbb{B}$ is the set of all body states in the sliding window. $\mathbf{r}_b \in \mathbb{R}^{15}$ is the residual of the preintegrated IMU measurements between the k-th and (k+1)-th frames. $\Omega_{imu}^k \in \mathbb{R}^{15\times15}$ is the corresponding information matrix, which is the inverse of the covariance matrix $\Sigma_{imu}^k \in \mathbb{R}^{15\times15}$ of the IMU preintegration. $\mathbb{C}$ is the set of all features tracked in the sliding window. $\rho_k(\cdot)$ is the robust kernel. $\mathbf{r}_c \in \mathbb{R}^2$ is the reprojection error of the feature l in the j-th frame and $\mathbf{z}_l^{c_j}$ is its visual observation. Specifically, for the feature l that is observed in the j-th frame, its pixel coordinates are $P_l^{c_j} = (u_l^{c_j}, v_l^{c_j})^T \in \mathbb{R}^2$. $\Omega_{cam}^{l,j} \in \mathbb{R}^{2\times2}$ is the information matrix of $P_l^{c_j}$.

The covariance of the VIS is propagated from the measurement uncertainties of the camera and IMU, as well as the prior covariance. The propagation process is given as follows.

Consider all the visual features observed in the sliding window, denoted as $\mathbf{z}_{cam} = \left\{P_1^{c_1}, \cdots, P_l^{c_j}, \cdots, P_m^{c_n}\right\}$, and the corresponding uncertainty is $\Sigma_{cam} = \text{diag}\left\{\Sigma_{cam}^{1,1}, \cdots, \Sigma_{cam}^{l,j}, \cdots, \Sigma_{cam}^{m,n}\right\}$. m is the number of feature points in the sliding window. n is the length of the sliding window. Then the covariance of VIS originated from visual observations is

$$\Sigma_{vis,cam} = \left(\frac{\partial \hat{\mathbf{x}}_{vis}}{\partial \mathbf{z}_{cam}}\right) \Sigma_{cam} \left(\frac{\partial \hat{\mathbf{x}}_{vis}}{\partial \mathbf{z}_{cam}}\right)^T. \tag{6}$$

Due to the lack of an explicit expression of $\hat{\mathbf{x}}_{vis}(\mathbf{z}_{cam})$, the Jacobian of $\hat{\mathbf{x}}_{vis}$ w.r.t. $\mathbf{z}_{cam}$ is difficult to be computed directly. The visual residual in the cost function (5) provides the corresponding relationship between $\hat{\mathbf{x}}_{vis}$ and $\mathbf{z}_{cam}$, therefore the Jacobian matrix can be obtained indirectly based on the implicit function theorem [43].

$$\frac{\partial \hat{\mathbf{x}}_{vis}}{\partial \mathbf{z}_{cam}} = -\left(\frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial \hat{\mathbf{x}}_{vis}^T}\right)^{-1} \left(\frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial \mathbf{z}_{cam}^T}\right)^T \tag{7}$$

where

$$E_c = \sum_{(l,j)\in\mathbb{C}} e_c = \sum_{(l,j)\in\mathbb{C}} \left\| \mathbf{r}_c(\mathbf{z}_l^{c_j}, \hat{\mathbf{x}}_{vis}) \right\|_{\Omega_{cam}^{l,j}}^2. \tag{8}$$

Assuming the feature l is observed for the first time in the i-th frame, its pixel coordinates are $P_l^{c_i}$. The normalized image coordinates corresponding to $P_l^{c_i}$ are denoted by $\tilde{P}_l^{c_i} \in \mathbb{R}^3$.

$$\tilde{P}_l^{c_i} = \pi_c^{-1}\left(P_l^{c_i}\right) \tag{9}$$

where $\pi_c^{-1}(\cdot)$ is the back projection function of the pinhole camera model. The feature l observed in the i-th frame is projected into the j-th frame, and its coordinates are denoted by $\hat{P}_l^{c_j} = \left[x_{j,l}, y_{j,l}, z_{j,l}\right]^T \in \mathbb{R}^3$.

$$\hat{P}_l^{c_j} = \mathbf{R}_b^c\left(\hat{\mathbf{R}}_w^{b_j}\left(\hat{\mathbf{R}}_{b_i}^w \tilde{P}_l^{b_i} + \hat{\mathbf{p}}_{b_i}^w - \hat{\mathbf{p}}_{b_j}^w\right) - \mathbf{p}_c^b\right) \tag{10}$$

$$\tilde{P}_l^{b_i} = \mathbf{R}_c^b \frac{1}{\hat{\lambda}_l} \tilde{P}_l^{c_i} + \mathbf{p}_c^b \tag{11}$$

where $\left\{\mathbf{R}_c^b, \mathbf{p}_c^b\right\}$ are the external parameters between the camera and the IMU. Then the visual residual in (8) is

$$\mathbf{r}_c(\mathbf{z}_l^{c_j}, \hat{\mathbf{x}}_{vis}) = \left[\frac{1}{z_{j,l}} \hat{P}_l^{c_j} - \tilde{P}_l^{c_j}\right]_{1:2}. \tag{12}$$

In order to compute the Jacobian of $\mathbf{z}_{cam}$ w.r.t. $\hat{\mathbf{x}}_{vis}$ in (7), we firstly consider the following Jacobian

$$\frac{\partial E_c}{\partial \hat{\mathbf{x}}_{vis}} = \sum_{(l,j)\in\mathbb{C}} \frac{\partial e_c}{\partial \hat{\mathbf{x}}_{vis}} \tag{13}$$

$$\frac{\partial e_c}{\partial \hat{\mathbf{x}}_{vis}} = 2\mathbf{r}_c^T \Omega_{cam}^{l,j} \mathbf{J}_c. \tag{14}$$

$\mathbf{J}_c$ is the Jacobian of $\mathbf{r}_c$ w.r.t. $\hat{\mathbf{x}}_{vis}$

$$\mathbf{J}_c = \frac{\partial \mathbf{r}_c}{\partial \hat{\mathbf{x}}_{vis}} = \frac{\partial \mathbf{r}_c}{\partial \hat{P}_l^{c_j}} \frac{\partial \hat{P}_l^{c_j}}{\partial \hat{\mathbf{x}}_{vis}} \tag{15}$$

$$\frac{\partial \mathbf{r}_c}{\partial \hat{P}_l^{c_j}} = \begin{bmatrix} \frac{1}{z_{j,l}} & 0 & -\frac{x_{j,l}}{z_{j,l}^2} \\ 0 & \frac{1}{z_{j,l}} & -\frac{y_{j,l}}{z_{j,l}^2} \end{bmatrix} \tag{16}$$

$$\frac{\partial \hat{P}_l^{c_j}}{\partial \hat{\mathbf{p}}_{b_i}^{w}} = \mathbf{R}_b^c \hat{\mathbf{R}}_w^{b_j} \tag{17}$$

$$\frac{\partial \hat{P}_l^{c_j}}{\partial \hat{\boldsymbol{\phi}}_{b_i}^{w}} = -\mathbf{R}_b^c \hat{\mathbf{R}}_w^{b_j} \left(\hat{\mathbf{R}}_{b_i}^{w} \tilde{P}_l^{b_i} \right)^{\wedge} \mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_i}^{w}) \tag{18}$$

$$\frac{\partial \hat{P}_l^{c_j}}{\partial \hat{\mathbf{p}}_{b_j}^{w}} = -\mathbf{R}_b^c \hat{\mathbf{R}}_w^{b_j} \tag{19}$$

$$\frac{\partial \hat{P}_l^{c_j}}{\partial \hat{\boldsymbol{\phi}}_{b_j}^{w}} = \mathbf{R}_b^c \hat{\mathbf{R}}_w^{b_j} \left(\hat{\mathbf{R}}_{b_i}^{w} \tilde{P}_l^{b_i} + \hat{\mathbf{p}}_{b_i}^{w} - \hat{\mathbf{p}}_{b_j}^{w} \right)^{\wedge} \mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_j}^{w}) \tag{20}$$

$$\frac{\partial \hat{P}_l^{c_j}}{\partial \hat{\lambda}_l} = -\mathbf{R}_b^c \hat{\mathbf{R}}_w^{b_j} \hat{\mathbf{R}}_{b_i}^{w} \mathbf{R}_c^b \frac{1}{\hat{\lambda}_l^2} \tilde{P}_l^{c_i}. \tag{21}$$

For $\mathbf{J}_c$, the rotation in $\hat{\mathbf{x}}_{vis}$ is derived in the Lie Algebra space, and the detailed derivation process is given in Appendix II-A. The term $\mathbf{J}_l(\phi) \in \mathbb{R}^{3\times3}$ is the left Jacobian of SO(3).

$$\mathbf{J}_l(\phi) = \mathbf{I}_{3\times3} + \frac{1-\cos(\|\phi\|)}{\|\phi\|^2}\phi^{\wedge} + \frac{\|\phi\| - \sin(\|\phi\|)}{\|\phi\|^3}(\phi^{\wedge})^2 \tag{22}$$

$$\mathbf{J}_l^{-1}(\phi) = \mathbf{I}_{3\times3} - \frac{1}{2}\phi^{\wedge} + \left(\frac{1}{\|\phi\|^2} + \frac{1+\cos(\|\phi\|)}{2\|\phi\|\sin(\|\phi\|)} \right)(\phi^{\wedge})^2. \tag{23}$$

$\mathbf{J}_l(\phi)$ relates a *global* additive increment in the tangent space to a multiplicative increment applied on the *left-hand side*. The Hessian matrix in (7) is

$$\begin{aligned} \frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial \hat{\mathbf{x}}_{vis}^T} &= \sum_{(l,j)\in\mathbb{C}} \frac{\partial}{\partial \hat{\mathbf{x}}_{vis}^T} \left(\frac{\partial e_c}{\partial \hat{\mathbf{x}}_{vis}} \right) \\ &= 2 \sum_{(l,j)\in\mathbb{C}} \left(\left(\frac{\partial \mathbf{J}_c^T}{\partial \hat{\mathbf{x}}_{vis}} \Omega_{cam}^{l,j} \mathbf{r}_c \right)^T + \mathbf{J}_c^T \Omega_{cam}^{l,j} \mathbf{J}_c \right), \end{aligned} \tag{24}$$

its calculation process is similar to (15). Another term in (7) is

$$\frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial \mathbf{z}_{cam}^T} = \frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial \left[P_1^{c_1 T}, \cdots P_l^{c_j T}, \cdots P_m^{c_n T} \right]}, \tag{25}$$

where

$$\frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial P_l^{c_j T}} = \frac{\partial}{\partial P_l^{c_j T}} \left(2\mathbf{r}_c^T \Omega_{cam}^{l,j} \mathbf{J}_c \right) = 2\mathbf{T}\Omega_{cam}^{l,j} \mathbf{J}_c. \tag{26}$$

$\mathbf{T}$ is the back projection matrix corresponding to $\pi_c^{-1}(\cdot)$ in (9). Then, the calculation process of $\frac{\partial^2 E_c}{\partial \hat{\mathbf{x}}_{vis} \partial P_l^{c_i T}}$ is similar to (26).

The covariance of VIS from preintegrated IMU measurements in (5) is similar to that from visual observations

$$\Sigma_{vis,imu} = \left(\frac{\partial \hat{\mathbf{x}}_{vis}}{\partial \mathbf{z}_{imu}} \right) \Sigma_{imu} \left(\frac{\partial \hat{\mathbf{x}}_{vis}}{\partial \mathbf{z}_{imu}} \right)^T \tag{27}$$

$$\frac{\partial \hat{\mathbf{x}}_{vis}}{\partial \mathbf{z}_{imu}} = -\left(\frac{\partial^2 E_b}{\partial \hat{\mathbf{x}}_{vis} \partial \hat{\mathbf{x}}_{vis}^T} \right)^{-1} \left(\frac{\partial^2 E_b}{\partial \hat{\mathbf{x}}_{vis} \partial \mathbf{z}_{imu}^T} \right)^T \tag{28}$$

$$E_b = \sum_{k\in\mathbb{B}} e_b = \sum_{k\in\mathbb{B}} \left\| \mathbf{r}_b(\mathbf{z}_{b_{k+1}}^{b_k}, \hat{\mathbf{x}}_{vis}) \right\|_{\Omega_{imu}^k}^2. \tag{29}$$

$\mathbf{z}_{imu} = \left\{ \mathbf{z}_{b_{k+1}}^{b_k}, k \in \mathbb{B} \right\}$ contains preintegrated IMU measurement terms between two consecutive image frames. Its covariance is $\Sigma_{imu} = \text{diag}\left\{ \Sigma_{imu}^k, k \in \mathbb{B} \right\}$. The residual of preintegrated IMU measurements is denoted by

$$\begin{aligned} &\mathbf{r}_b(\mathbf{z}_{b_{k+1}}^{b_k}, \hat{\mathbf{x}}_{vis}) \\ &= \begin{bmatrix} \delta\mathbf{p}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\phi}_{b_{k+1}}^{b_k} \\ \delta\mathbf{v}_{b_{k+1}}^{b_k} \\ \delta\mathbf{b}_a \\ \delta\mathbf{b}_g \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{R}}_w^{b_k} \left(\hat{\mathbf{p}}_{b_{k+1}}^{w} - \hat{\mathbf{p}}_{b_k}^{w} + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 - \hat{\mathbf{v}}_{b_k}^{w} \Delta t_k \right) - \mathbf{p}_{b_{k+1}}^{b_k} \\ \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k} \hat{\mathbf{R}}_{b_k}^{w} \hat{\mathbf{R}}_w^{b_{k+1}} \right) \\ \hat{\mathbf{R}}_w^{b_k} \left(\hat{\mathbf{v}}_{b_{k+1}}^{w} + \mathbf{g}^w \Delta t_k - \hat{\mathbf{v}}_{b_k}^{w} \right) - \mathbf{v}_{b_{k+1}}^{b_k} \\ \hat{\mathbf{b}}_{a_{b_{k+1}}} - \hat{\mathbf{b}}_{a_{b_k}} \\ \hat{\mathbf{b}}_{\omega_{b_{k+1}}} - \hat{\mathbf{b}}_{\omega_{b_k}} \end{bmatrix}. \end{aligned} \tag{30}$$

The Jacobian of E_b w.r.t. $\hat{\mathbf{x}}_{vis}$ is

$$\frac{\partial E_b}{\partial \hat{\mathbf{x}}_{vis}} = \sum_{k\in\mathbb{B}} \frac{\partial e_b}{\partial \hat{\mathbf{x}}_{vis}} = 2 \sum_{k\in\mathbb{B}} \mathbf{r}_b^T \Omega_{imu}^k \mathbf{J}_b \tag{31}$$

where $\mathbf{J}_b$ is the Jacobian of $\mathbf{r}_b$ w.r.t. $\hat{\mathbf{x}}_{vis}$.

1) Jacobians of $\delta\mathbf{p}_{b_{k+1}}^{b_k}$: The preintegrated translation $\mathbf{p}_{b_{k+1}}^{b_k}$ is represented by the first-order approximations w.r.t. the biases $\hat{\mathbf{b}}_{a_{b_k}}$ and $\hat{\mathbf{b}}_{\omega_{b_k}}$ if the estimates of biases change minorly.

$$\begin{aligned} &\mathbf{p}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{a_{b_k}} + \Delta\mathbf{b}_{a_{b_k}}, \hat{\mathbf{b}}_{\omega_{b_k}} + \Delta\mathbf{b}_{\omega_{b_k}}) \\ &\approx \mathbf{p}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{a_{b_k}}, \hat{\mathbf{b}}_{\omega_{b_k}}) + \mathbf{J}_{\mathbf{b}_a}^{\mathbf{p}} \Delta\mathbf{b}_{a_{b_k}} + \mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{p}} \Delta\mathbf{b}_{\omega_{b_k}} \end{aligned} \tag{32}$$

with

$$\mathbf{J}_{\mathbf{b}_a}^{\mathbf{p}} = \frac{\partial \mathbf{p}_{b_{k+1}}^{b_k}}{\partial \mathbf{b}_{a_k}}, \quad \mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{p}} = \frac{\partial \mathbf{p}_{b_{k+1}}^{b_k}}{\partial \mathbf{b}_{\omega_k}}. \tag{33}$$

Jacobians $\mathbf{J}_{\mathbf{b}_a}^{\mathbf{p}}$ and $\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{p}}$ describe how the preintegrated result changes due to the changes in the bias estimates [44]. Then, Jacobians of $\delta\mathbf{p}_{b_{k+1}}^{b_k}$ are

$$\frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\boldsymbol{\phi}}_{b_k}^{w}} = \hat{\mathbf{R}}_w^{b_k} \alpha_1^{\wedge} \mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w}), \quad \frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}} = \mathbf{0}_{3\times3} \tag{34}$$

$$\frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{p}}_{b_k}^{w}} = -\hat{\mathbf{R}}_w^{b_k}, \quad \frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{p}}_{b_{k+1}}^{w}} = \hat{\mathbf{R}}_w^{b_k} \tag{35}$$

$$\frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{v}}_{b_k}^{w}} = -\hat{\mathbf{R}}_w^{b_k} \Delta t_k, \quad \frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{v}}_{b_{k+1}}^{w}} = \mathbf{0}_{3\times3} \tag{36}$$

$$\frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{b}}_{a_{b_k}}} = -\mathbf{J}_{\mathbf{b}_a}^{\mathbf{p}}, \quad \frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{b}}_{a_{b_{k+1}}}} = \mathbf{0}_{3\times3} \tag{37}$$

$$\frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{b}}_{\omega_{b_k}}} = -\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{p}}, \quad \frac{\partial \delta\mathbf{p}_{b_{k+1}}^{b_k}}{\partial \hat{\mathbf{b}}_{\omega_{b_{k+1}}}} = \mathbf{0}_{3\times3} \tag{38}$$

with $\alpha_1 = \hat{\mathbf{p}}_{b_{k+1}}^{w} - \hat{\mathbf{p}}_{b_k}^{w} + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 - \hat{\mathbf{v}}_{b_k}^{w} \Delta t_k$.

2) Jacobians of $\delta\mathbf{v}_{b_{k+1}}^{b_k}$: Similar to $\mathbf{p}_{b_{k+1}}^{b_k}$, the preintegrated linear velocity $\mathbf{v}_{b_{k+1}}^{b_k}$ can also be linearized by

$$\begin{aligned}&\mathbf{v}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{a_{b_k}}+\Delta\mathbf{b}_{a_{b_k}},\hat{\mathbf{b}}_{\omega_{b_k}}+\Delta\mathbf{b}_{\omega_{b_k}})\\&\quad\approx\mathbf{v}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{a_{b_k}},\hat{\mathbf{b}}_{\omega_{b_k}})+\mathbf{J}_{\mathbf{b}_a}^{\mathbf{v}}\Delta\mathbf{b}_{a_{b_k}}+\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{v}}\Delta\mathbf{b}_{\omega_{b_k}}.\end{aligned}\tag{39}$$

Definitions of Jacobians $\mathbf{J}_{\mathbf{b}_a}^{\mathbf{v}}$ and $\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{v}}$ are similar to $\mathbf{J}_{\mathbf{b}_a}^{\mathbf{p}}$ and $\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{p}}$ in (32), respectively. Then, Jacobians of $\delta\mathbf{v}_{b_{k+1}}^{b_k}$ are

$$\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\boldsymbol{\phi}}_{b_k}^{w}}=\hat{\mathbf{R}}_{w}^{b_k}\alpha_2^{\wedge}\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w}),\quad\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}}=\mathbf{0}_{3\times3}\tag{40}$$

$$\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{p}}_{b_k}^{w}}=\mathbf{0}_{3\times3},\quad\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{p}}_{b_{k+1}}^{w}}=\mathbf{0}_{3\times3}\tag{41}$$

$$\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{v}}_{b_k}^{w}}=-\hat{\mathbf{R}}_{w}^{b_k},\quad\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{v}}_{b_{k+1}}^{w}}=\hat{\mathbf{R}}_{w}^{b_k}\tag{42}$$

$$\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{a_{b_k}}}=-\mathbf{J}_{\mathbf{b}_a}^{\mathbf{v}},\quad\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{a_{b_{k+1}}}}=\mathbf{0}_{3\times3}\tag{43}$$

$$\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{\omega_{b_k}}}=-\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{v}},\quad\frac{\partial\delta\mathbf{v}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{\omega_{b_{k+1}}}}=\mathbf{0}_{3\times3}\tag{44}$$

with $\alpha_2=\hat{\mathbf{v}}_{b_{k+1}}^{w}+\mathbf{g}^{w}\Delta t_k-\hat{\mathbf{v}}_{b_k}^{w}$.

3) Jacobians of $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$: the preintegrated rotation is only affected by the change of a gyroscope bias.

$$\mathbf{R}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{\omega_{b_k}}+\Delta\mathbf{b}_{\omega_{b_k}})\approx\mathbf{R}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{\omega_{b_k}})\mathrm{Exp}(\mathbf{J}_{\mathbf{b}_\omega}^{\boldsymbol{\phi}}\Delta\mathbf{b}_{\omega_{b_k}})\tag{45}$$

where the definition of $\mathbf{J}_{\mathbf{b}_\omega}^{\boldsymbol{\phi}}$ is similar to $\mathbf{J}_{\mathbf{b}_\omega}^{\mathbf{p}}$. $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ is only related to the rotation and gyroscope bias, hence, the corresponding Jacobians are computed as

$$\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\boldsymbol{\phi}}_{b_k}^{w}}=\mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w}),\quad\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}}=\alpha_3\tag{46}$$

$$\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{\omega_{b_k}}}=\mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_{\mathbf{b}_\omega}^{\boldsymbol{\phi}},\quad\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{\omega_{b_{k+1}}}}=\mathbf{0}\tag{47}$$

with $\alpha_3=-\mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathrm{Exp}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w})$. Computation of Jacobians of $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ are relatively complicated, thus the detailed derivation processes of Jacobians w.r.t. the rotation and the gyroscope bias are shown in Appendixes II-A and II-B, respectively. Remaining Jacobians are zero.

4) Jacobians of $\delta\mathbf{b}_a$ and $\delta\mathbf{b}_\omega$:

$$\frac{\partial\delta\mathbf{b}_a}{\partial\hat{\mathbf{b}}_{a_{b_k}}}=\frac{\partial\delta\mathbf{b}_\omega}{\partial\hat{\mathbf{b}}_{\omega_{b_k}}}=-\mathbf{I}_{3\times3},\quad\frac{\partial\delta\mathbf{b}_a}{\partial\hat{\mathbf{b}}_{a_{b_{k+1}}}}=\frac{\partial\delta\mathbf{b}_\omega}{\partial\hat{\mathbf{b}}_{\omega_{b_{k+1}}}}=\mathbf{I}_{3\times3}.\tag{48}$$

$\delta\mathbf{b}_a$ and $\delta\mathbf{b}_\omega$ are irrelevant to $\hat{\boldsymbol{\phi}}_{b_k}^{w}$, $\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}$, $\hat{\mathbf{p}}_{b_k}^{w}$, $\hat{\mathbf{p}}_{b_{k+1}}^{w}$, $\hat{\mathbf{v}}_{b_k}^{w}$ and $\hat{\mathbf{v}}_{b_{k+1}}^{w}$, hence, the corresponding Jacobians are zero.

The Jacobian of $\hat{\mathbf{x}}_{vis}$ w.r.t. $\mathbf{z}_{imu}$ in (28) is obtained by further differentiating (31).

In the sliding window, marginalization in [18] is used to convert the variables that do not need to be optimized into the priori information. We use $\hat{\mathbf{x}}_m$ to represent the marginalized state vector in the sliding window, and $\hat{\mathbf{x}}_p$ to represent the preserved state vector. Specifically, if the second latest frame is a key-frame, $\hat{\mathbf{x}}_m$ represents the body state and visual measurements corresponding to the oldest frame. If the second latest frame is not a key-frame, $\hat{\mathbf{x}}_m$ represents the body state corresponding to the second latest frame. $\mathbf{H}_p$ represents the constraints between the marginalized state vector and the preserved state vector. The uncertainty Σ_m of $\hat{\mathbf{x}}_m$ has been calculated in the previous optimization process and current sliding window. It does not change because $\hat{\mathbf{x}}_m$ will not be updated. The uncertainty of $\hat{\mathbf{x}}_m$ is propagated to $\hat{\mathbf{x}}_p$ by

$$\Sigma_p=\left(\frac{\partial\hat{\mathbf{x}}_p}{\partial\hat{\mathbf{x}}_m}\right)\Sigma_m\left(\frac{\partial\hat{\mathbf{x}}_p}{\partial\hat{\mathbf{x}}_m}\right)^T\tag{49}$$

$$\frac{\partial\hat{\mathbf{x}}_p}{\partial\hat{\mathbf{x}}_m}=-\left(\frac{\partial^2\mathrm{e}_p}{\partial\hat{\mathbf{x}}_p\partial\hat{\mathbf{x}}_p^T}\right)^{-1}\left(\frac{\partial^2\mathrm{e}_p}{\partial\hat{\mathbf{x}}_p\partial\hat{\mathbf{x}}_m^T}\right)^T\tag{50}$$

$$\mathrm{e}_p=\left(\mathbf{r}_p-\mathbf{H}_p\hat{\mathbf{x}}_{vis}\right)^T\left(\mathbf{r}_p-\mathbf{H}_p\hat{\mathbf{x}}_{vis}\right).\tag{51}$$

Finally, we obtain the pose uncertainty Σ_{VIS} by

$$\Sigma_{VIS}=\Sigma_{vis,cam}+\Sigma_{vis,imu}+\Sigma_p.\tag{52}$$

B. Depth Subsystem and Its Uncertainty Estimation

The point-to-plane ICP is used to optimize the pose in DS. ICP is one of the widely used algorithms in aligning 3D models, 2D or 3D surfaces reconstruction and robot localization. Point-to-plane ICP, as a variant of ICP, usually exhibits excellent performance in the environment with a large number of planes [45], [46]. Specifically, in urban environments, roads as well as buildings contain a large number of planes. In off-road scenes, local planes can be extracted from the ground, trees and other objects. Thus, in this paper, point-to-plane ICP is adopted to achieve the pose estimation. Note that, classical LO (e.g., LOAM and LeGO-LOAM [47]) and LIO (e.g., LIO-SAM and FAST-LIO [48]) use both planar and edge features, because planar and edge features are complementary in many cases. However, it has been pointed out in [49] that for edges with the depth jumps between foreground objects and background objects, the extracted edge features are neither reliable nor accurate. Especially for an RGBD camera, the edge features extracted from the point clouds have large observation noises. In [50] and [51], theoretical and experimental analyses have shown that edges extracted from RGBD cameras have low accuracy and stability. Therefore, to guarantee the robustness of DS and to make VIDO more general, we only use planar features in VIDO.

For the point cloud obtained by the LiDAR, the IMU measurement is used to correct the distortion of the LiDAR scan. Then, the curvature of each point is computed. The points with a small curvature are selected as the planar points and pushed into a set $\mathbb{H}$. A planar point $\boldsymbol{P}_s^{d_{i-1}}\in\mathbb{H}^{d_{i-1}}$ in the $(i-1)$-th frame (last frame) is projected into the world frame [23] with the fused pose estimate $\{\bar{\mathbf{R}}_{b_{i-1}}^{w},\bar{\mathbf{p}}_{b_{i-1}}^{w}\}$, which is given in Section III-C.

$$\bar{\boldsymbol{P}}_s^{w}=\bar{\mathbf{R}}_{b_{i-1}}^{w}\left(\mathbf{R}_d^b\boldsymbol{P}_s^{d_{i-1}}+\mathbf{p}_d^b\right)+\bar{\mathbf{p}}_{b_{i-1}}^{w}\tag{53}$$

where $\{\mathbf{R}_d^b, \mathbf{p}_d^b\}$ are the external parameters between the depth sensor and the IMU. The corresponding uncertainty $\Sigma_{\boldsymbol{P}_s}^{d_{i-1}} \in \mathbb{R}^{3\times3}$ is propagated into the world frame by

$$\bar{\Sigma}_{\boldsymbol{P}_s}^{w} = \bar{\mathbf{R}}_{b_{i-1}}^{w}\mathbf{R}_d^b\Sigma_{\boldsymbol{P}_s}^{d_{i-1}}\mathbf{R}_b^d\bar{\mathbf{R}}_w^{b_{i-1}} + \frac{\partial\bar{\boldsymbol{P}}_s^w}{\partial\bar{\boldsymbol{\phi}}_{b_{i-1}}^w}\bar{\Sigma}_{\bar{\boldsymbol{\phi}}}\left(\frac{\partial\bar{\boldsymbol{P}}_s^w}{\partial\bar{\boldsymbol{\phi}}_{b_{i-1}}^w}\right)^T + \frac{\partial\bar{\boldsymbol{P}}_s^w}{\partial\bar{\mathbf{p}}_{b_{i-1}}^w}\bar{\Sigma}_{\bar{\mathbf{p}}}\left(\frac{\partial\bar{\boldsymbol{P}}_s^w}{\partial\bar{\mathbf{p}}_{b_{i-1}}^w}\right)^T \quad (54)$$

where

$$\frac{\partial\bar{\boldsymbol{P}}_s^w}{\partial\bar{\boldsymbol{\phi}}_{b_{i-1}}^w} = -\left(\bar{\mathbf{R}}_{b_{i-1}}^w\left(\mathbf{R}_d^b\boldsymbol{P}_s^{d_{i-1}} + \mathbf{p}_d^b\right)\right)^\wedge\mathbf{J}_l(\bar{\boldsymbol{\phi}}_{b_{i-1}}^w) \quad (55)$$

$$\frac{\partial\bar{\boldsymbol{P}}_s^w}{\partial\bar{\mathbf{p}}_{b_{i-1}}^w} = \mathbf{I}_{3\times3}. \quad (56)$$

$\bar{\Sigma}_{\bar{\boldsymbol{\phi}}} \in \mathbb{R}^{3\times3}$ and $\bar{\Sigma}_{\bar{\mathbf{p}}} \in \mathbb{R}^{3\times3}$ represent the submatrices corresponding to $\bar{\boldsymbol{\phi}}_{b_i}^w$ and $\bar{\mathbf{p}}_{b_i}^w$ in the covariance matrix of $\bar{\Sigma} \in \mathbb{R}^{6\times6}$, respectively, which can also be found in Section III-C. Correspondingly, the point set after projection is denoted as $\bar{\mathbb{H}}^w$. The planar points in the i-th frame (current frame) are projected into the world frame through the transform provided by the IMU preintegration. For the point $\boldsymbol{P}_l^{d_i} \in \mathbb{R}^3$, several non-collinear nearest neighbors in $\bar{\mathbb{H}}^w$ are found to fit a plane by the KD tree. $\bar{\boldsymbol{P}}_l^w \in \mathbb{R}^3$ is the nearest neighbor of the projection point of $\boldsymbol{P}_l^{d_i}$, and $\bar{\boldsymbol{n}}_l^w \in \mathbb{R}^3$ is the normal of the corresponding plane. The state vector of DS includes the position $\check{\mathbf{p}}_{b_k}^w$ and orientation $\check{\boldsymbol{\phi}}_{b_k}^w$

$$\check{\mathbf{x}}_{ds} = \check{\boldsymbol{\xi}}_{b_i}^w = \left[\check{\mathbf{p}}_{b_i}^{w\,T}, \check{\boldsymbol{\phi}}_{b_i}^{w\,T}\right]^T \in \mathbb{R}^6.$$

$\mathbf{z}_{pc} = \left\{\mathbf{z}_1^{d_i}, \cdots, \mathbf{z}_l^{d_i}, \cdots, \mathbf{z}_{n_l}^{d_i}\right\}$ denotes the observation of the depth sensor, where $\mathbf{z}_l^{d_i} = \left[\boldsymbol{P}_l^{d_i\,T}, \bar{\boldsymbol{P}}_l^{w\,T}, \bar{\boldsymbol{n}}_l^{w\,T}\right]^T \in \mathbb{R}^9$. n_l is the length of $\mathbb{H}$. Then, the depth residual is computed by

$$r_d(\mathbf{z}_l^{b_i}, \check{\mathbf{x}}_{ds}) = \left(\check{\mathbf{R}}_{b_i}^w\boldsymbol{P}_l^{b_i} + \check{\mathbf{p}}_{b_i}^w - \bar{\boldsymbol{P}}_l^w\right)\cdot\bar{\boldsymbol{n}}_l^w \quad (57)$$

$$\boldsymbol{P}_l^{b_i} = \mathbf{R}_d^b\boldsymbol{P}_l^{d_i} + \mathbf{p}_d^b. \quad (58)$$

We minimize the Mahalanobis norm of the depth residual

$$\min_{\check{\mathbf{x}}_{ds}}\sum_{l\in\mathbb{H}}\left\|r_d(\mathbf{z}_l^{d_i}, \check{\mathbf{x}}_{ds})\right\|_{1/\sigma_d^2}^2 \quad (59)$$

where σ_d^2 is

$$\sigma_d^2 = \text{trace}(\Sigma_l^{d_i}) \quad (60)$$

where $\Sigma_l^{d_i} = \text{diag}\left\{\Sigma_{\boldsymbol{P}_l}^{d_i}, \Sigma_{\bar{\boldsymbol{P}}_l}^w, \Sigma_{\bar{\boldsymbol{n}}_l}^w\right\} \in \mathbb{R}^{9\times9}$.

The calculation process of the covariance in DS is similar to that in VIS, i.e.,

$$\Sigma_{ds} = \left(\frac{\partial\check{\mathbf{x}}_{ds}}{\partial\mathbf{z}_{pc}}\right)\Sigma_{pc}\left(\frac{\partial\check{\mathbf{x}}_{ds}}{\partial\mathbf{z}_{pc}}\right)^T \quad (61)$$

$$\frac{\partial\check{\mathbf{x}}_{ds}}{\partial\mathbf{z}_{pc}} = -\left(\frac{\partial^2E_d}{\partial\check{\mathbf{x}}_{ds}\partial\check{\mathbf{x}}_{ds}^T}\right)^{-1}\left(\frac{\partial^2E_d}{\partial\check{\mathbf{x}}_{ds}\partial\mathbf{z}_{pc}^T}\right)^T \quad (62)$$

where

$$E_d = \sum_{l\in\mathbb{H}}e_d = \sum_{l\in\mathbb{H}}\frac{1}{\sigma_d^2}r_d^2(\mathbf{z}_l^{d_i}, \check{\mathbf{x}}_{ds}). \quad (63)$$

$\Sigma_{pc} = \text{diag}\left\{\Sigma_l^{d_i}, l \in \mathbb{H}\right\}$ denotes the uncertainty of $\mathbf{z}_{pc}$. The Jacobian of E_d w.r.t $\check{\mathbf{x}}_{ds}$ is computed by

$$\frac{\partial E_d}{\partial\check{\mathbf{x}}_{ds}} = 2\sum_{l\in\mathbb{H}}\frac{1}{\sigma_d^2}r_d\frac{\partial r_d}{\partial\check{\mathbf{x}}_{ds}} = 2\sum_{l\in\mathbb{H}}\frac{1}{\sigma_d^2}r_d\mathbf{J}_d, \quad (64)$$

where

$$\mathbf{J}_d = \frac{\partial r_d}{\partial\begin{bmatrix}\check{\mathbf{p}}_{b_i}^w\\ \check{\boldsymbol{\phi}}_{b_i}^w\end{bmatrix}} = \left[\bar{\boldsymbol{n}}_l^{w\,T}, -\bar{\boldsymbol{n}}_l^{w\,T}\left(\check{\mathbf{R}}_{b_i}^w\boldsymbol{P}_l^{b_i}\right)^\wedge\mathbf{J}_l(\check{\boldsymbol{\phi}}_{b_i}^w)\right]. \quad (65)$$

Then, differentiating (64) w.r.t $\check{\mathbf{x}}_{ds}$ and $\mathbf{z}_{pc}$ yields

$$\frac{\partial^2E_d}{\partial\check{\mathbf{x}}_{ds}\partial\check{\mathbf{x}}_{ds}^T} = 2\sum_{l\in\mathbb{H}}\frac{1}{\sigma_d^2}\mathbf{J}_d^T\mathbf{J}_d + 2\sum_{l\in\mathbb{H}}\frac{1}{\sigma_d^2}r_d\left(\frac{\partial\mathbf{J}_d^T}{\partial\check{\mathbf{x}}_{ds}}\right)^T \quad (66)$$

$$\frac{\partial^2E_d}{\partial\check{\mathbf{x}}_{ds}\partial\mathbf{z}_{pc}^T} = 2\sum_{l\in\mathbb{H}}\frac{1}{\sigma_d^2}\left(\frac{\partial\mathbf{J}_d^Tr_d}{\partial\left[\mathbf{z}_1^{d_i\,T}, \cdots\mathbf{z}_l^{d_i\,T}, \cdots\mathbf{z}_{n_l}^{d_i\,T}\right]}\right)^T, \quad (67)$$

and their calculation process are similar to (15) and (26). Further, continue to use the chain rule for (67) to calculate the partial derivatives of the observation $\left[\boldsymbol{P}_l^{d_{i-1}}, \boldsymbol{P}_l^{d_{i-1}}, \boldsymbol{n}_l^{d_{i-1}}\right]$ and the fused pose estimate of the last frame. Then, Σ_{pc} is also adjusted accordingly.

C. Pose Fusion Based On the CI Filter

In Section III-A, (5) and (52) give the pose and covariance matrix $\left\{\hat{\boldsymbol{\xi}}_{b_i}^w, \Sigma_{vis}\right\}$ obtained by VIS, where $\Sigma_{vis} \in \mathbb{R}^{6\times6}$ is the submatrix of Σ_{VIS} corresponding to $\hat{\boldsymbol{\xi}}_{b_i}^w$. In Section III-B, (59) and (61) give the pose and covariance matrix $\left\{\check{\boldsymbol{\xi}}_{b_i}^w, \Sigma_{ds}\right\}$ obtained by DS. Since DS uses the IMU preintegration of VIS as the initial value, estimate results of the two subsystems are not strictly independent. Therefore, inconsistency will occur if the extended Kalman filter (EKF), the unscented Kalman filter (UKF) or the particle filter is used to fuse them. This is because a premise of the EKF, UKF and particle filter is that the state variables are independent. If a correlation occurs between two states, these filters will yield an inconsistent or over-confident fusion result. In order to solve the problem, we use the CI filter to obtain a consistent and optimal fusion result. The CI filter can perform information fusion for two or more state variables when the correlation between them is unknown [52], [53]. A main advantage of the CI filter is that, it produces a consistent and convergent fused estimate regardless of the correlation between state variables. In fact, it gives a common upper bound of actual covariances, which has robustness with respect to unknown correlations [54], [55]. The covariance intersection fusion estimator is calculated by

$$\bar{\boldsymbol{\xi}}_{b_i}^w = \bar{\Sigma}\left[\rho\,\Sigma_{vis}^{-1}\hat{\boldsymbol{\xi}}_{b_i}^w \oplus (1-\rho)\,\Sigma_{ds}^{-1}\check{\boldsymbol{\xi}}_{b_i}^w\right] \quad (68)$$

where $\oplus$ represents the addition of two poses. $\bar{\Sigma} \in \mathbb{R}^{6\times6}$ is defined as

$$\bar{\Sigma} = \left[\rho\,\Sigma_{vis}^{-1} + (1-\rho)\,\Sigma_{ds}^{-1}\right]^{-1} \quad (69)$$

Fig. 2. UGV equipped with a self-developed sensor suite.

where $\rho \in [0, 1]$ is obtained by minimizing the objective function

$$\min_{\rho} tr\,\bar{\Sigma}. \tag{70}$$

A premise that the CI filter can perform consistent fusion is that an optimal ρ can be solved from the nonlinear optimization problem (70). ρ reflects the contributions of VIS and DS in the pose fusion. The most credible and consistent pose fusion is achieved when the uncertainty in (69) is minimal. In this paper, the quantum particle swarm optimization (QPSO) method [56] is adopted to solve (70). QPSO is a modified particle swarm optimization (PSO) algorithm. PSO is a stochastic population-based optimization method. It seeks the optimal solution by information transferring and sharing between individuals, which is a highly efficient parallel search algorithm. However, PSO has too much reliance on personal best position and global best position of the particles, which may limit its searching ability. In contrast, QPSO can find a solution quickly in the global space. Since update of the particle position is not related to the previous motion of the particle, the global searching ability of QPSO is better than that of PSO. In order to speed up the convergence rate of QPSO, we first introduce an EKF to preliminarily fuse two estimation results. Then, using the result of EKF as the initial value, QPSO is adopted to obtain the optimal solution to (70). Finally, the optimal weight ρ is applied to the CI filter in (68) and (69) to generate an accurate and consistent pose estimate. It should be pointed out that although the result of EKF is inconsistent, further optimization through QPSO can greatly increase the possibility of obtaining the global optimal solution to (70).

It is worth to note that, when the estimation results of both subsystems are relatively accurate, the fusion result is more accurate in a statistical sense. If one of the both subsystems is inaccurate or even fails, its uncertainty is large. As a result, the corresponding contribution to pose fusion is small. At this time, the fusion result is biased towards the estimation result of the accurate subsystem. In this way, adaptive fusion of the two subsystems is achieved, and a consistent and optimal pose estimate is obtained.

D. Mapping

In the mapping process, the system selects key-frame point clouds for scan-to-map matching to reduce the accumulated drift of the odometry. A key-frame is selected when the change of the pose of the UGV exceeds a threshold compared with the pose of previous LiDAR key-frame. The system extracts a local map from the map, and the points with *small uncertainties* in the local map are selected for feature association with key-frame point clouds. Then, the point-to-plane ICP in Section III-B is applied to refine the pose estimate. Finally, we use (53) to project the key-frame point cloud to the world frame and build the map incrementally. The covariance of the points in the LiDAR key-frame is also propagated by (54).

IV. Experiments

To fully evaluate the proposed method, we perform experiments on the Multi Vehicle Stereo Event Camera (MVSEC) [57] dataset and an UGV equipped with a self-developed sensor suite (Fig. 2), respectively. In Section IV-A, we evaluate VIDO on the MVSEC dataset. This dataset provides a test bed for automated driving, which is collected from a vehicle at 12 m/s on urban roads in residential areas. In the MVSEC dataset, two sequences are collected in day settings and three in night settings, and each sequence is over 1.2 kilometers in length. The on-board sensors included two experimental mDAVIS-346B cameras (output events, APS grayscale images and IMU measurements), two Skybotix integrated VI-sensors and a Velodyne VLP-16 LiDAR. Although the dataset contains event images, we do not use them in VIDO. We only use the images and IMU measurements from one Skybotix integrated VI-sensor, as well as the point cloud data from the VLP-16 LiDAR. There are vehicles, pedestrians, buildings, trees, as well as up and down slopes in the sequences. The ground truth is obtained by fusing LiDAR information with IMU and GPS. In Section IV-B, we evaluate VIDO on outdoor datasets collected by a Clearpath Husky UGV. The self-developed sensor suite is composed of a Velodyne VLP-16 LiDAR, an Xsens MTi 30 IMU and a Microsoft Kinect 2.0 RGBD camera. The outdoor experiments are conducted in different scenes, including the lawn, the forest and two parks, respectively, which contain rugged terrains and repetitive texture. It should be noted that we only use the RGB image of the Kinect 2.0 camera, rather than its depth information to implement the VIDO system. In Section IV-C, to verify generality of VIDO in terms of sensor selection, we use the Kinect 2.0 camera to collect the depth data. It needs to be pointed out that the depth camera cannot work well in outdoor environments. Therefore, we have to choose an indoor scene. All the methods are executed on an INTEL NUC7i7BNH using the robot operating system (ROS) in Ubuntu Linux.

A. Evaluation on MVSEC Dataset

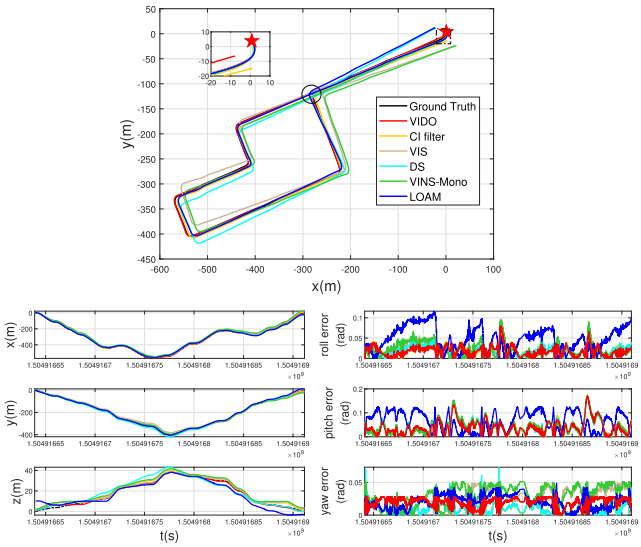
In this section, we perform experiments on the MVSEC dataset. The image resolution of the MVSEC dataset is 752×480. For an image, it is necessary to extract enough feature points, and their distribution on the image should be reasonable. Extracting too few features makes it difficult to form sufficient visual constraints. Extracting too many features increases the computational load. At this time, it has a possibility that the points with insufficient gray gradients are also

TABLE I
RMSEs OF ABSOLUTE TRAJECTORY ERROR (ATE) AND ORIENTATION ERROR

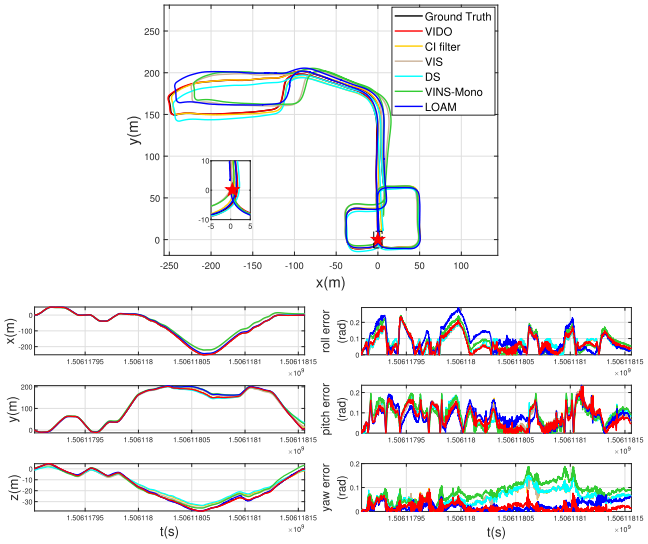
Sequence		VIDO	CI filter	VIS	DS	VINS-Mono	LOAM	VILO
Outdoor_Day_1	ATE / m	**0.666**	2.087	7.312	6.539	9.860	2.028	2.407
(261s, 1207m)	Orientation Error / rad	**0.104**	0.112	0.111	0.113	0.116	0.122	/
Outdoor_Day_2	ATE / m	**2.047**	4.028	6.983	11.347	7.489	20.598	2.326
(653s, 3467m)	Orientation Error / rad	**0.115**	0.126	0.129	0.126	0.130	0.139	/
Outdoor_Night_1	ATE / m	**1.046**	2.510	15.926	2.552	21.082	2.590	1.940
(262s, 1217m)	Orientation Error / rad	**0.099**	0.101	0.120	0.104	0.122	0.139	/
Outdoor_Night_2	ATE / m	3.293	3.942	8.008	9.198	8.378	5.198	**2.673**
(374s, 2109m)	Orientation Error / rad	**0.107**	0.110	0.112	0.109	0.108	0.145	/
Outdoor_Night_3	ATE / m	**1.423**	2.133	5.831	5.679	6.521	4.678	1.585
(276s, 1613m)	Orientation Error / rad	**0.097**	0.100	0.106	0.101	0.107	0.129	/

regarded as feature points, resulting in a large number of errors in feature association. On the other hand, uneven distribution of feature points also affects the accuracy of feature association. Hence, for VIS, in order to extract sufficient and evenly distributed feature points, the maximum feature number is set to be 140, and the minimum distance between two features is set to be 25 pixels. In addition, since the uncertainties of all states in the sliding window need to be calculated, in order to make a tradeoff between utilizing adequate information and reducing the computational complexity, the length of the sliding window is set as 5. For DS, we use the same parameters as [23], i.e., the number of non-collinear planar points that are closest to the projected point and used to fit a plane is selected as 3. In the mapping process, the number of planar points to fit a plane is selected as 5, and it is also the same as [23]. Finally, in order to build the point cloud map, the keyframe point cloud is downsampled and added to the map. The downsampled grid size is 0.6m.

We quantitatively compare VIDO with VINS-Mono, LOAM and VILO [25]. VILO is also a loosely coupled visual-inertial-depth odometry system. It used the front-end of VINS-Mono to obtain the inter-frame pose estimate, which was used to remove the distortion of LiDAR point clouds and provide an initial prediction for point cloud matching. In VILO, scan-to-scan matching and scan-to-map matching processes are similar to those in LOAM. In other words, VILO only used visual and inertial observations for initialization, and then used the LiDAR data to optimize the pose. In all the experiments, loop closure in VINS-Mono is not run to verify the performance of the odometry system. Five sequences in the MVSEC dataset are chosen to compare four methods. To reduce the impact of random factors, we run all sequences 5 times each. The average absolute trajectory error (ATE) and orientation error are used to evaluate their performances. The results are shown in Fig. 3, Table I and Video 1. The mapping result of VIDO of sequence *Outdoor_Day_1* is shown in Fig. 4. In addition, the duration and the trajectory length of each sequence are also shown in Table I. Since VILO is not open-source, the results of ATE of VILO in Table I comes from [25]. Note that, the results of the orientation angles of VILO were not given in [25]. Therefore, we do not present them in Table I. Fig. 3 shows trajectories, position estimation results and orientation estimation errors in sequences *Outdoor_Night_3* and



(a) Experimental results in sequence *Outdoor_Night_3*.



(b) Experimental results in sequence *Outdoor_Day_1*.

Fig. 3. Experimental results on the MVSEC dataset. The red star indicates the starting point.

Outdoor_Day_1. The results of ATE show that the superiority of VIDO is over the visual-inertial and LiDAR systems in all

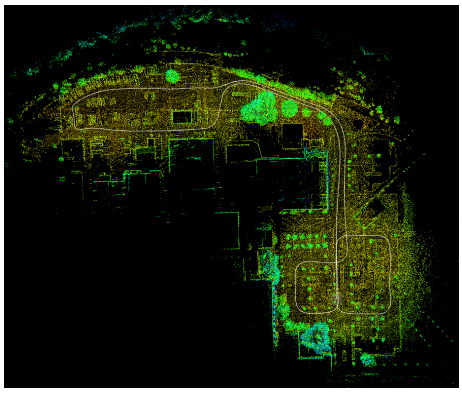


Fig. 4. Mapping result of VIDO of sequence *Outdoor_Day_1*.

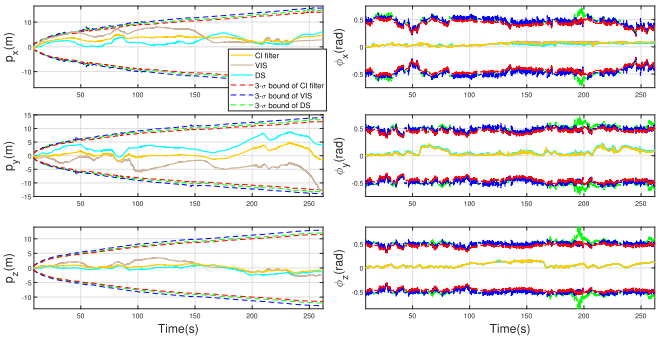


Fig. 5. Pose uncertainties of VIS, DS and the CI fusion result.

cases (the accuracy is improved by 72.7%∼95.0% compared with VINS-Mono and 36.6%∼90.1% compared with LOAM). In addition, VIDO outperforms VILO in most cases (the accuracy is improved by 10.2%∼52.7%). Moreover, VIDO has the highest accuracy in terms of the orientation estimation. The results about pose estimates of VIS, DS and the pose fusion result of the CI filter are also given in Table I and Fig. 3. The results show that the accuracy of the pose fusion result is better than that of VIS and DS. Furthermore, the 3-σ bounds derived from the estimated covariance of VIS, DS and the CI fusion result in sequence *Outdoor_Day_1* is shown are Fig. 5, respectively. It is obvious that the fusion result of the CI filter can decrease the pose uncertainties of VIS and DS.

It should be noted that the IMU constraints can assist to maintain the orientation accuracy of the system, especially the roll and pitch angles (because they are observable). In addition, the camera and LiDAR are able to extract sufficient constraint information from the environment to further improve the accuracy of the pose estimation and map building. Due to the rich environmental features, both VIS and DS can establish enough feature constraints, which ensure the accuracy of the pose estimation of the two subsystems. The pose fusion process further refines the localization results. For example, at the intersection marked in Fig. 3(a) and Fig. 6, the accuracy of VIDO is significantly higher than that of VINS-Mono and LOAM. On one hand, compared with VINS-Mono and LOAM, VIDO makes full use of the environmental information. On the other hand, when the vehicle reaches the intersection where it has

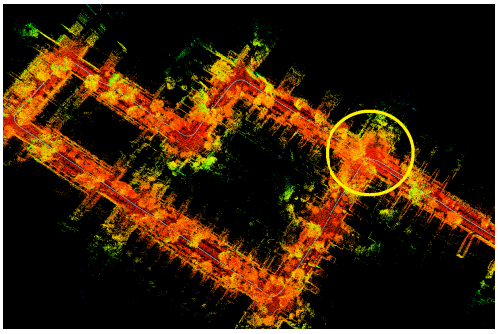


Fig. 6. Point cloud map constructed by VIDO when the vehicle passes the intersection for the second time. The scene marked by a circle is the intersection.

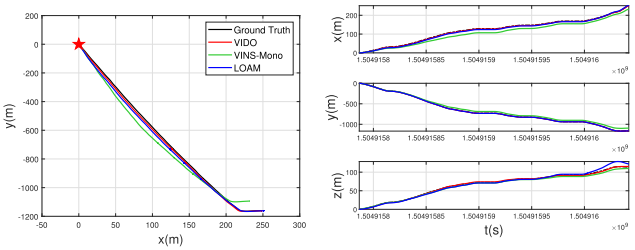


Fig. 7. Experimental results in sequence *Outdoor_Night_1*. The red star indicates the starting point.

previously visited, the mapping process of VIDO utilizes the historical information by matching the current frame with the local map around the intersection to further constrain the pose. In contrast, although LOAM also performs frame-to-map matching, large accumulated errors make it unable to correctly associate the current frame with the local map around the intersection. As a result, LOAM fails to achieve local map matching and pose correction in the current frame. Accordingly, the quality of the map constructed by LOAM is worse than that by VIDO in this area. As for VINS-Mono, its performance heavily depends on loop-closure and global BA. When VINS-Mono runs without loop-closure, localization errors will largely accumulate since VINS-Mono does not maintain map points and employ the historical information.

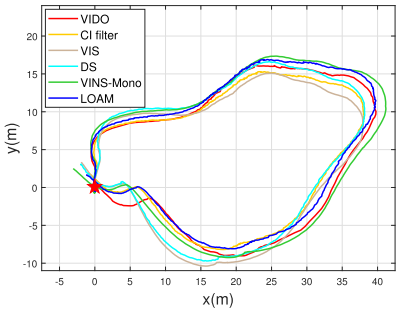
Fig. 7 intuitively shows the impact of a large initialization error on the systems. In the *Outdoor_Night_1*, the vehicle travels straight on an urban road for more than one kilometer. It only performs translation motion, and has no motion in roll, pitch and yaw direction. That is to say, in this sequence, the initialization of VINS is very insufficient. Therefore, estimation of the scalar factor is inaccurate, which makes initialization errors large. As can be seen from Fig. 7, the localization result of VINS-Mono has a large error. In comparison, although localization accuracy of VIS in VIDO may be affected by large initialization errors, DS cannot be affected, since the pose estimate in DS is directly computed by point cloud matching. As a result, the pose fusion of VIS and DS is robust against large initialization errors.

B. Evaluation on Outdoor Datasets Collected by the UGV

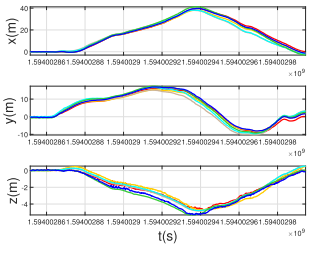
In this section, we perform outdoor experiments on an UGV equipped with a self-developed sensor suite. The image

TABLE II
START-TO-END DRIFT IN THE OUTDOOR TESTS

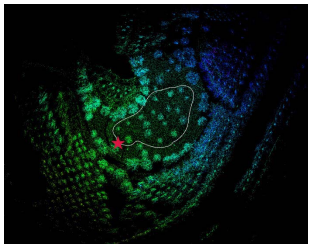
Dataset (Length)	VIDO	CI filter	VIS	DS	VINS-Mono	LOAM
Park 1 (143.02m)	**0.513m(0.4%)**	2.022m(1.4%)	3.847m(2.7%)	3.730m(2.6%)	3.887m(2.7%)	0.982m(0.7%)
Park 2 (434.19m)	**0.653m(0.2%)**	4.494m(1.0%)	10.649m(2.3%)	5.210m(1.2%)	5.564m(1.3%)	16.767m(3.9%)
Forest (302.47m)	**0.589m(0.2%)**	6.514m(2.2%)	8.123m(2.7%)	9.816m(3.3%)	18.251m(6.0%)	31.784m(10.5%)
Lawn (178.82m)	**0.103m(0.1%)**	6.142m(3.4%)	12.000m(6.7%)	6.382m(3.6%)	12.138m(6.8%)	29.514m(16.6%)
Long-distance (2223.91m)	**5.250m(0.2%)**	26.985m(1.2%)	37.696m(1.6%)	24.937m(1.1%)	41.555m(1.9%)	103.484m(4.6%)



(a) Trajectories in *Park 1* dataset.

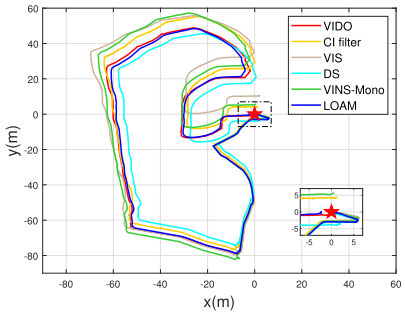


(b) Position estimation results.

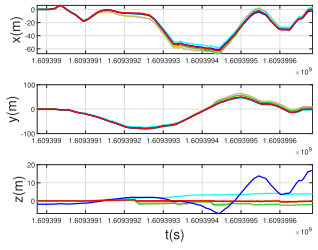


(c) Mapping result of VIDO.

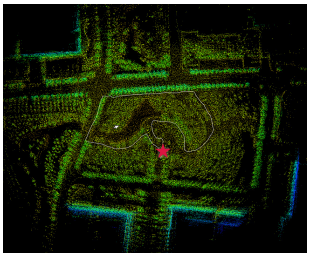
Fig. 8. Experimental results on *Park 1* dataset. The red star indicates the starting point.



(a) Trajectories in *Park 2* dataset.



(b) Position estimation results.



(c) Mapping result of VIDO.

Fig. 9. Experimental results on *Park 2* dataset. The red star indicates the starting point.

resolution of the self-collected dataset is 960 × 540, larger than that of the MVSEC dataset. Hence, for VIS, both the maximum feature number and the minimum distance between two features are set to be larger than those of the MVSEC dataset, i.e., 150 and 30, respectively. In addition, the length of the sliding window is 5, which is the same as that in Section IV-A. Similar to the MVSEC dataset, we also use the LiDAR to obtain point cloud and thus the corresponding parameters are the same as in Section IV-A.

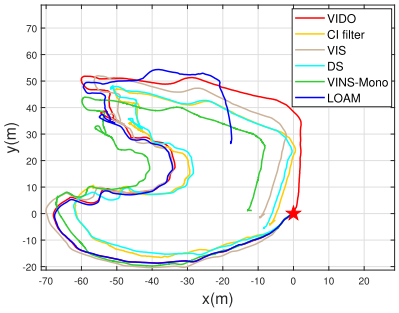
We compare the accuracy of VIDO with that of VIS, DS, the CI fusion result, VINS-Mono and LOAM in term of the start-to-end drift, and the results (drift amount and percentage) are shown in Table II. Similar to Section IV-A, we run all sequences 5 times each. The trajectory length in each dataset is given in brackets.

First, in the *Park 1* dataset, the pose estimation result of each method and the point cloud map built by VIDO are shown in Fig. 8. Due to the small scale, as well as rich texture and structure information, the ending point of VIDO approximately coincides with the given starting point, while the other two methods perform poorly.

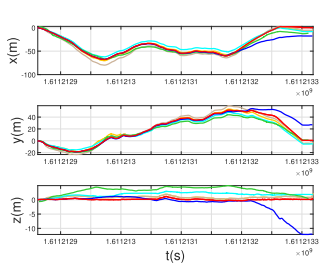
Second, compared with the *Park 1* dataset, the *Park 2* dataset has a larger scene scale, which is more challenging for odometry systems. Note that on the *Park 2* dataset, VIDO performs significantly better than VINS-Mono and LOAM, as shown in Fig. 9.

Third, for the *Forest* dataset, the pose estimation and mapping results are shown in Fig. 10. In this experiment, the UGV first enters a forest, which is represented by a red box in Fig. 10(c). In this area, due to the rich scene information, the trajectories estimated by three methods are not much different. After a period of time, the UGV comes to the grass, which is represented by a blue box in Fig. 10(c). This area has less scene information and a higher appearance similarity. As a result, both visual-based and LiDAR-based odometeries are difficult to extract reliable features, thus their pose estimation results are inaccurate. Nevertheless, the proposed method can obtain statistically better results, even if both VINS-Mono and LOAM have a large drift.

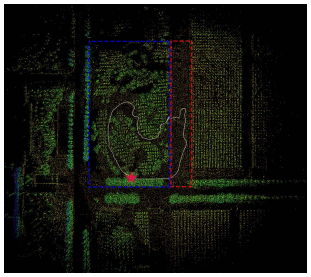
Fourth, for the *Lawn* dataset, the results are shown in Fig. 11 and Video 1. Similar to the *Forest* dataset, the lawn scene lacks structural and texture information. As a result, the accuracy of LOAM is very poor. Because of the motion constraint provided by IMU, VINS-Mono can roughly estimate the motion trajectory, but there is still an obvious drift. In comparison, the performance of VIDO on this dataset is still the best.



(a) Trajectories in the *Forest* dataset.

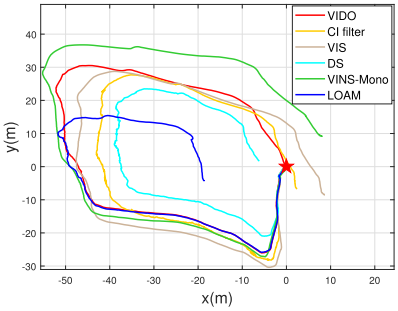


(b) Position estimation results.

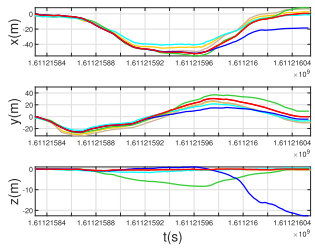


(c) Mapping result of VIDO.

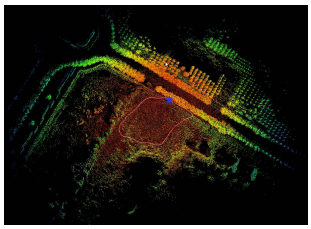
Fig. 10. Experimental results on the *Forest* dataset. The red star indicates the starting point.



(a) Trajectories in the *Lawn* dataset.



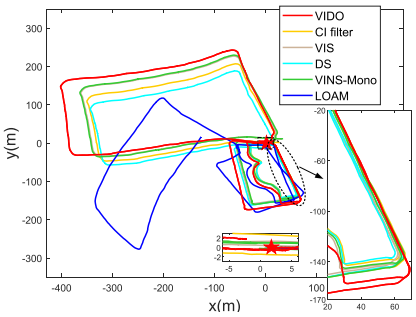
(b) Position estimation results.



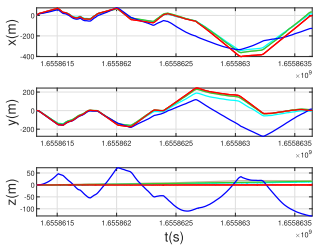
(c) Mapping result of VIDO.

Fig. 11. Experimental results on the *Lawn* dataset. The star indicates the starting point.

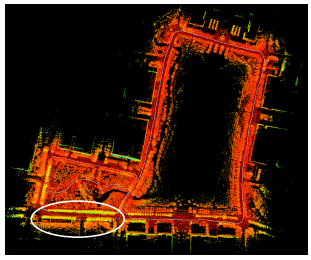
At last, to further verify the performance of VIDO, we conduct a long-term and long-distance experiment. In this experiment, we adopt a PointGrey BFLY-PGE-31S4M-C camera. The image resolution is 2048×1536, and the maximum feature number and the minimum distance between two features are set to be 160 and 120, respectively. The UGV travels for about 40 minutes. The total length of the trajectory is 2224m. The experimental environment is composed of a field scene (about 232s) and an urban traffic flow scene (about 2178s).



(a) Trajectories in the *Long-distance* dataset.



(b) Position estimation results.



(c) Mapping result of VIDO.

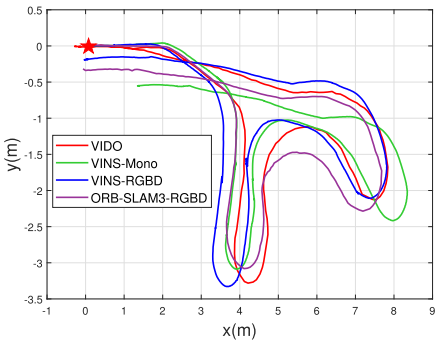
Fig. 12. Estimated trajectory and mapping result in the long-distance experiment. The scene marked by a circle is the road that the vehicle repeatedly passes through.

It includes various terrains such as gravel pavements, forests, grass, steps, and urban roads. There are vehicles, bicycles and pedestrians in the urban traffic flow scene. The pose estimation and mapping results are shown in Fig. 12 and Video 2. The results show that for a long-term and long-distance localization problem in a multi-terrain environment, VIDO can still achieve more accurate and robust pose estimation compared with other methods. Due to the long running time, LOAM accumulates a large amount of errors, resulting in a large drift of the trajectory. In the interval of 495s to 682s, the scene and trajectory of the vehicle are the same as those in the interval of 0s to 186s (marked by a circle in Fig. 12). Since VIDO builds the point cloud map in real time, in the interval of 495s to 682s, VIDO can be regarded as performing matching between the current frame and the previously built map. As can be seen from Fig. 12, such a frame-to-map matching process ensures a high degree of coincidence of the two trajectories. In contrast, since VINS-Mono without loop-closure does not maintain map points, it cannot use the historical information to reduce the accumulated error even though the vehicle has reached the place where it has previously visited. Thus, it presents larger localization errors than VIDO.

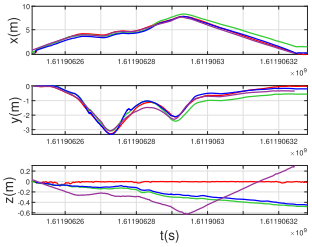
Note that in the *Park 2*, *Forest*, *Lawn* and *Long-distance* datasets, cumulative errors of LOAM in the z-axis are very large. The reason is that the ground of the environment is rugged, which brings a lot of disturbances in the z-axis direction of the UGV. In addition, the cluttered weeds on the ground make it difficult for the LiDAR odometery to extract accurate planar features. Thus, it is unlikely to provide sufficient constraints in the z-axis direction. In VIDO, the IMU preintegration module provides a relatively accurate initial value for DS, which reduces the scan-to-scan matching errors.

TABLE III
START-TO-END DRIFT IN THE INDOOR TEST

Dataset (Length)	VIDO	VINS-Mono	VINS-RGBD	ORB-SLAM3 -RGBD
Indoor (21.8m)	**0.231m (1.6%)**	1.546m (7.3%)	0.484m (2.3%)	0.387m (1.8%)



(a) Trajectories in the *Indoor* dataset.



(b) Position estimation results.

(c) Mapping result of VIDO.

Fig. 13. Experimental results in the *Indoor* scene. The star indicates the starting point.

Furthermore, the optimization result of DS is fused with the pose estimate of the VIS, which further improves the accuracy and robustness of the system. The results in the *Park 2* and *Long-distance* demonstrate that VIDO can work well on complicated terrains, which provides an accurate 6-DOF pose estimation technique for the UGV navigation in off-road environments.

C. Evaluation on Indoor Dataset Collected by the UGV

In the indoor experiment, since the same Kinect 2.0 camera is used as in Section IV-B, the parameters of VIS are the same as that in Section IV-B. However, in this experiment, point clouds are collected with an RGBD camera, instead of a LiDAR. Point clouds from the RGBD camera are denser than those acquired by the LiDAR. Hence, the downsampled grid size of a keyframe point cloud is 0.08m, which is smaller than that in Section IV-A and Section IV-B.

We compare VIDO with VINS-Mono, VINS-RGBD [58] and ORB-SLAM3-RGBD in terms of the drift. VINS-RGBD is an RGBD-inertial SLAM method based on VINS-Mono. It integrated depth measurements into the visual-inertial initialization process of VINS-Mono. ORB-SLAM3 [21] added inertial measurements of the IMU into ORB-SLAM2 [59], and supported monocular, stereo and RGBD cameras. ORB-SLAM3-RGBD refers to ORB-SLAM3 using an RGBD camera. It directly used the depth measurement instead of the triangularization result to obtain the 3D position of map points. Loop closure in all the comparison methods is not run. Similar to outdoor experiments, the UGV travels around the environment with a size of 12m $\times$ 8m and goes back to the starting position. We compare the accuracy of the four methods in term of the start-to-end drift, and the results (drift amount and percentage) are shown in Table III. The pose estimation and mapping results are shown in Fig. 13 and Video 1.

The drift of VINS-Mono is larger than that of VINS-RGBD and ORB-SLAM3-RGBD, indicating that depth measurements have indeed improved the localization accuracy. However, the measurement range of the depth camera is very limited, resulting in that a large number of feature points have no depth measurement. Therefore, direct use of depth measurements cannot yield significant improvements on the accuracy of the VIO systems. In VIDO, the structural information of the environment is obtained from the depth point cloud, which provides constraints for the pose estimation. In this experiment, the UGV has relatively little motion in the z-axis. As shown in Fig. 11(b), all the comparison methods have large drifts in the z-axis, while VIDO has no obvious drift in the z-axis due to the consideration of plane constraints. In summary, VIDO performs the best in the indoor experiment.

V. CONCLUSION AND FUTURE WORKS

In this paper, to achieve the accurate and robust 6-DOF pose estimation for navigation of UGVs on complex terrains, we have proposed VIDO, a robust and consistent monocular visual-inertial-depth odometry based on a loosely coupled framework. For both the visual-inertial and depth subsystems, closed-form covariance matrices of the pose estimate have been rigorously computed based on the implicit function theorem. The impacts of different sensor noises are fully involved into the pose estimation of the UGV, which can significantly improve accuracy and robustness of the system. Then, the localization results of VIS and DS subsystems are further fused with the CI filter to yield a consistent and optimal visual-inertial-depth odometry system. In order to smooth the estimated trajectory and reduce the accumulated drift, the system also introduces the scan-to-map registration algorithm, and the pose uncertainty is further propagated to map points for maintaining an accurate map. Extensive experiments on public datasets and in real-world environments have been presented, which demonstrate the superiority of the proposed method over the state-of-the-art ones.

In a word, the proposed method has two main merits. First, we derive the closed-form covariance matrices for pose estimation results of two subsystems in the optimization-based framework. To the best of our knowledge, this is the first work that presents a closed-form covariance matrix for VINS-Mono such that a more complete visual-inertial odometry system can be achieved. The obtained uncertainty estimates for VIS and DS provide an important foundation for online evaluation of the odometry systems and multi-sensor fusion. In addition, the

closed-form uncertainty facilitates motion prediction of active SLAM. In particular, VIDO also estimates the uncertainty of map points. It can be used for quantitative evaluation of map quality, and further, for keyframe selection and map management. Second, a robust loosely coupled visual-inertial-depth odometry system is developed. The VIS and DS subsystems not only can be fused by the CI filter to provide a more accurate pose estimate result, but also can work independently. The proposed loosely coupled framework is also applicable to other sensors, such as infrared cameras, stereo cameras, and heterogeneous LiDARs. As long as the pose and uncertainty of the UGV can be estimated from data of these sensors, they can be easily integrated into VIDO.

Future work involves loop closure detection and global optimization to achieve a more complete SLAM system. In addition, we will focus on how to combine VIDO with occlusion detection and moving object detection algorithms to further improve the applicability of our method in dynamic urban environments. Moreover, VIDO has to face the degeneracy problem in some extreme situations, which is a common challenge in the SLAM field. We will integrate degeneracy detection into VIDO and use the IMU measurements in unconstrained motion subspaces to address the degeneracy problem.

APPENDIX I
IMU PREINTEGRATION

Inertial observations include the linear acceleration and angular velocity of the vehicle. They are expressed as

$$\begin{aligned}\hat{\mathbf{a}}_t &= \mathbf{a}_t + \mathbf{b}_{a_t} + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{n}_a \\ \hat{\boldsymbol{\omega}}_t &= \boldsymbol{\omega}_t + \mathbf{b}_{\omega_t} + \mathbf{n}_\omega,\end{aligned} \tag{71}$$

where $\hat{\mathbf{a}}_t \in \mathbb{R}^3$ and $\hat{\boldsymbol{\omega}}_t \in \mathbb{R}^3$ are the raw IMU measurements in the body frame. $\mathbf{a}_t \in \mathbb{R}^3$ and $\boldsymbol{\omega}_t \in \mathbb{R}^3$ are their respective true values. $\mathbf{b_{a_t}} \in \mathbb{R}^3$ and $\mathbf{b}_{\omega_t} \in \mathbb{R}^3$ are acceleration and gyroscope biases modeled as random walk, whose derivatives are Gaussian white noises, i.e., $\dot{\mathbf{b}}_{a_t} = \mathbf{n}_{b_a}$, $\dot{\mathbf{b}}_{\omega_t} = \mathbf{n}_{b_\omega}$. $\mathbf{n}_a \in \mathbb{R}^3$ and $\mathbf{n}_\omega \in \mathbb{R}^3$ are Gaussian white noises in acceleration and gyroscope measurements. $\mathbf{g^w} \in \mathbb{R}^3$ is the constant gravity vector in the world frame.

For two consecutive frames b_k and b_{k+1}, assuming that IMU has n_k sets of measurements in the time interval $[t_k, t_{k+1}]$, and the sampling period is Δt_k, which is constant. The position, velocity and orientation states can be propagated by IMU measurements in the time interval $[t_k, t_{k+1}]$ in the world frame.

$$\begin{aligned}\mathbf{p}_{b_{k+1}}^w &= \mathbf{p}_{b_k}^w + \sum_{t=t_k}^{n_k-1}[\mathbf{v}_t^w \Delta t_k + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 \\ &\quad + \frac{1}{2}\mathbf{R}_t^w(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a)\Delta t_k^2] \\ \mathbf{v}_{b_{k+1}}^w &= \mathbf{v}_{b_k}^w + \mathbf{g}^w \Delta t_k + \sum_{t=t_k}^{n_k-1} \mathbf{R}_t^w(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \mathbf{n}_a)\Delta t_k \\ \mathbf{R}_{b_{k+1}}^w &= \mathbf{R}_{b_k}^w \prod_{t=t_k}^{n_k-1} \exp[((\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \mathbf{n}_\omega)\Delta t_k)^\wedge].\end{aligned} \tag{72}$$

When initial states of the position, velocity and rotation of the frame b_k change in the time interval $[t_k, t_{k+1}]$, we do not need to propagate IMU measurements repeatedly since preintegration is performed. We change the reference frame from the world frame to the local frame b_k, and (72) can be rewritten by

$$\begin{aligned}\mathbf{p}_{b_{k+1}}^w &= \mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w \Delta t_k + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 + \mathbf{R}_{b_k}^w(\Delta \mathbf{p}_{b_{k+1}}^{b_k} - \delta \mathbf{p}_{b_{k+1}}^{b_k}) \\ \mathbf{v}_{b_{k+1}}^w &= \mathbf{v}_{b_k}^w + \mathbf{g}^w \Delta t_k + \mathbf{R}_{b_k}^w(\Delta \mathbf{v}_{b_{k+1}}^{b_k} - \delta \mathbf{v}_{b_{k+1}}^{b_k}) \\ \mathbf{R}_{b_{k+1}}^w &= \mathbf{R}_{b_k}^w \Delta \mathbf{R}_{b_{k+1}}^{b_k} \exp(-\delta \boldsymbol{\phi}_{b_{k+1}}^{b_k\ \wedge})\end{aligned} \tag{73}$$

where $\Delta \mathbf{p}_{b_{k+1}}^{b_k} \in \mathbb{R}^3$, $\Delta \mathbf{v}_{b_{k+1}}^{b_k} \in \mathbb{R}^3$ and $\Delta \boldsymbol{\phi}_{b_{k+1}}^{b_k} \in \mathbb{R}^3$ are preintegrated IMU measurements

$$\begin{aligned}\Delta \mathbf{p}_{b_{k+1}}^{b_k} &= \sum_{t=t_k}^{n_k-1}[\mathbf{v}_t^w \Delta t_k + \frac{1}{2}\mathbf{R}_t^w(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})\Delta t_k^2] \\ \Delta \mathbf{v}_{b_{k+1}}^{b_k} &= \sum_{t=t_k}^{n_k-1} \mathbf{R}_t^w(\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})\Delta t_k \\ \Delta \mathbf{R}_{b_{k+1}}^{b_k} &= \prod_{t=t_k}^{n_k-1} \exp[((\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t})\Delta t_k)^\wedge].\end{aligned} \tag{74}$$

$\delta \mathbf{z}_{b_{k+1}}^{b_k} = [\delta \mathbf{p}_{b_{k+1}}^{b_k\ T}, \delta \mathbf{v}_{b_{k+1}}^{b_k\ T}, \delta \boldsymbol{\phi}_{b_{k+1}}^{b_k\ T}]^T \in \mathbb{R}^9$ is a zero-mean white Gaussian noise vector corresponding to the preintegrated IMU measurements, and it is obtained by

$$\begin{aligned}\delta \mathbf{p}_{b_{k+1}}^{b_k} &= \sum_{t=t_k}^{n_k-1}[\delta \mathbf{v}_t^{b_k} \Delta t_k \\ &\quad + \frac{1}{2}\Delta \mathbf{R}_t^{b_k}(\mathbf{n}_a - (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})^\wedge \delta \boldsymbol{\phi}_t^{b_k})\Delta t_k^2] \\ \delta \mathbf{v}_{b_{k+1}}^{b_k} &= \sum_{t=t_k}^{n_k-1} \Delta \mathbf{R}_t^{b_k}(\mathbf{n}_a - (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t})^\wedge \delta \boldsymbol{\phi}_t^{b_k})\Delta t_k \\ \delta \boldsymbol{\phi}_{b_{k+1}}^{b_k} &= \sum_{t=t_k}^{n_k-1} \Delta \mathbf{R}_{t+1}^{b_{k+1}} \mathbf{J}_r^t((\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t})\Delta t)\mathbf{n}_\omega \Delta t_k\end{aligned} \tag{75}$$

where $\mathbf{J}_r^t$ is the right Jacobian of SO(3). Then, $\delta \mathbf{z}_t^{b_k}$ can be computed in an iterative form

$$\begin{aligned}\begin{bmatrix}\delta \mathbf{p}_t^{b_k} \\ \delta \mathbf{v}_t^{b_k} \\ \delta \boldsymbol{\phi}_t^{b_k}\end{bmatrix} &= \begin{bmatrix}\mathbf{I}_{3\times3} & \Delta t_k \mathbf{I}_{3\times3} & -\frac{1}{2}\Delta \mathbf{R}_{t-1}^{b_k}(\hat{\mathbf{a}}_{t-1} - \mathbf{b}_{a_t})^\wedge \Delta t_k^2 \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & -\Delta \mathbf{R}_{t-1}^{b_k}(\hat{\mathbf{a}}_{t-1} - \mathbf{b}_{a_t})^\wedge \Delta t_k \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{R}_{t-1}^t\end{bmatrix} \\ &\quad \times \begin{bmatrix}\delta \mathbf{p}_{t-1}^{b_k} \\ \delta \mathbf{v}_{t-1}^{b_k} \\ \delta \boldsymbol{\phi}_{t-1}^{b_k}\end{bmatrix} + \begin{bmatrix}\frac{1}{2}\Delta \mathbf{R}_{t-1}^{b_k}\Delta t_k^2 & \mathbf{0}_{3\times3} \\ \Delta \mathbf{R}_{t-1}^{b_k}\Delta t_k & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}_r^{t-1}\Delta t_k\end{bmatrix}\begin{bmatrix}\mathbf{n}_a \\ \mathbf{n}_\omega\end{bmatrix} \\ &= \mathbf{F}_{t-1}\delta \mathbf{z}_{t-1}^{b_k} + \mathbf{G}_{t-1}\mathbf{n}_{t-1}.\end{aligned} \tag{76}$$

The covariance $\Sigma_{b_{k+1}}^{b_k} \in \mathbb{R}^{9\times9}$ is computed iteratively with the initial covariance $\Sigma_{b_k}^{b_k} = \mathbf{0_{9\times9}}$

$$\Sigma_t^{b_k} = \mathbf{F}_{t-1}\Sigma_{t-1}^{b_k}\mathbf{F}_{t-1}^T + \mathbf{G}_{t-1}\Sigma_{\mathbf{n}_{t-1}}\mathbf{G}_{t-1}^T \tag{77}$$

where $\Sigma_{\mathbf{n}_{t-1}} \in \mathbb{R}^{6\times6}$ is the covariance of $\mathbf{n}_{t-1} \in \mathbb{R}^6$. In addition, IMU biases are propagated by

$$\mathbf{b}_{a_{t+1}} = \mathbf{b}_{a_t} + \delta \mathbf{b}_{a_t}$$

$$\mathbf{b}_{\omega_{t+1}} = \mathbf{b}_{\omega_t} + \delta\mathbf{b}_{\omega_t} \tag{78}$$

where $\delta\mathbf{b}_{a_t}$ and $\delta\mathbf{b}_{\omega_t}$ are discrete noises corresponding to IMU biases. They have zero mean and covariance $\Sigma_{b_a} = \Delta t_k \text{cov}(\mathbf{n}_{b_a}) \in \mathbb{R}^3$ and $\Sigma_{b_\omega} = \Delta t_k \text{cov}(\mathbf{n}_{b_\omega}) \in \mathbb{R}^3$, respectively.

APPENDIX II
COMPLETE DERIVATIONS FOR THE JACOBIAN MATRICES

A. Jacobians w.r.t. Rotation

A small change in rotation is expressed as a small disturbance $\delta\boldsymbol{\phi}$ in the Lie Algebra space. Correspondingly, a small disturbance on the SO(3) is calculated by

$$\text{Exp}\left(\boldsymbol{\phi} + \delta\boldsymbol{\phi}\right) \approx \text{Exp}\left(\mathbf{J}_l(\boldsymbol{\phi})\delta\boldsymbol{\phi}\right)\text{Exp}\left(\boldsymbol{\phi}\right). \tag{79}$$

It is the first-order approximation of SO(3). Similarly, the first-order expansion of the logarithmic map is

$$\text{Log}\left(\text{Exp}\left(\delta\boldsymbol{\phi}\right)\text{Exp}\left(\boldsymbol{\phi}\right)\right) \approx \boldsymbol{\phi} + \mathbf{J}_l^{-1}(\boldsymbol{\phi})\delta\boldsymbol{\phi}. \tag{80}$$

On the basis of (79), the small changes in $\hat{P}_l^{c_j}$ w.r.t. $\hat{\boldsymbol{\phi}}_{b_i}^{w}$ in (18) is

$$\begin{aligned}
&\hat{P}_l^{c_j}\left(\hat{\boldsymbol{\phi}}_{b_i}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_i}^{w}\right)\\
&= \mathbf{R}_b^c\left(\hat{\mathbf{R}}_w^{b_j}\left(\text{Exp}(\hat{\boldsymbol{\phi}}_{b_i}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_i}^{w})\tilde{P}_l^{b_i} + \hat{\mathbf{p}}_{b_i}^{w} - \hat{\mathbf{p}}_{b_j}^{w}\right) - \mathbf{p}_c^b\right)\\
&\overset{(79)}{\approx} \mathbf{R}_b^c\left(\hat{\mathbf{R}}_w^{b_j}\left(\text{Exp}(\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_i}^{w})\delta\hat{\boldsymbol{\phi}}_{b_i}^{w})\hat{\mathbf{R}}_{b_i}^{w}\tilde{P}_l^{b_i} + \hat{\mathbf{p}}_{b_i}^{w} - \hat{\mathbf{p}}_{b_j}^{w}\right) - \mathbf{p}_c^b\right)\\
&\overset{(4)}{\approx} \mathbf{R}_b^c\left(\hat{\mathbf{R}}_w^{b_j}\left(\left(\mathbf{I} + \left(\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_i}^{w})\delta\hat{\boldsymbol{\phi}}_{b_i}^{w}\right)^\wedge\right)\hat{\mathbf{R}}_{b_i}^{w}\tilde{P}_l^{b_i} + \hat{\mathbf{p}}_{b_i}^{w} - \hat{\mathbf{p}}_{b_j}^{w}\right) - \mathbf{p}_c^b\right)\\
&= \hat{P}_l^{c_j}\left(\hat{\boldsymbol{\phi}}_{b_i}^{w}\right) + \mathbf{R}_b^c\hat{\mathbf{R}}_w^{b_j}\left(\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_i}^{w})\delta\hat{\boldsymbol{\phi}}_{b_i}^{w}\right)^\wedge\hat{\mathbf{R}}_{b_i}^{w}\tilde{P}_l^{b_i}\\
&= \hat{P}_l^{c_j}\left(\hat{\boldsymbol{\phi}}_{b_i}^{w}\right) - \mathbf{R}_b^c\hat{\mathbf{R}}_w^{b_j}\left(\hat{\mathbf{R}}_{b_i}^{w}\tilde{P}_l^{b_i}\right)^\wedge\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_i}^{w})\delta\hat{\boldsymbol{\phi}}_{b_i}^{w}.
\end{aligned} \tag{81}$$

Then, the Jacobian of $\hat{P}_l^{c_j}$ w.r.t. $\hat{\boldsymbol{\phi}}_{b_i}^{w}$ in (18) is calculated by

$$\begin{aligned}
\frac{\partial\hat{P}_l^{c_j}}{\partial\hat{\boldsymbol{\phi}}_{b_i}^{w}} &= \lim_{\delta\hat{\boldsymbol{\phi}}_{b_i}^{w}\to\mathbf{0}}\frac{\hat{P}_l^{c_j}\left(\hat{\boldsymbol{\phi}}_{b_i}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_i}^{w}\right) - \hat{P}_l^{c_j}\left(\hat{\boldsymbol{\phi}}_{b_i}^{w}\right)}{\delta\hat{\boldsymbol{\phi}}_{b_i}^{w}}\\
&= -\mathbf{R}_b^c\hat{\mathbf{R}}_w^{b_j}\left(\hat{\mathbf{R}}_{b_i}^{w}\tilde{P}_l^{b_i}\right)^\wedge\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_i}^{w}).
\end{aligned} \tag{82}$$

The calculation processes of (20), (34), (40), (55) and (65) are similar. In comparison, the derivation of the Jacobian matrices of $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ is different. A useful property of the exponential mapping is

$$\begin{aligned}
&\mathbf{R}\text{Exp}(\boldsymbol{\phi})\mathbf{R}^T = \exp(\mathbf{R}\boldsymbol{\phi}^\wedge\mathbf{R}^T) = \text{Exp}(\mathbf{R}\boldsymbol{\phi})\\
&\Leftrightarrow \mathbf{R}\text{Exp}(\boldsymbol{\phi}) = \text{Exp}(\mathbf{R}\boldsymbol{\phi})\mathbf{R}.
\end{aligned} \tag{83}$$

Then, the small changes in $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ w.r.t. $\hat{\boldsymbol{\phi}}_{b_k}^{w}$ and $\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}$ are

$$\begin{aligned}
&\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}\left(\hat{\boldsymbol{\phi}}_{b_k}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_k}^{w}\right)\\
&= \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k}\text{Exp}\left(\hat{\boldsymbol{\phi}}_{b_k}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_k}^{w}\right)\hat{\mathbf{R}}_w^{b_{k+1}}\right)\\
&\overset{(79)}{\approx} \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k}\text{Exp}\left(\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w})\delta\hat{\boldsymbol{\phi}}_{b_k}^{w}\right)\hat{\mathbf{R}}_{b_k}^{w}\hat{\mathbf{R}}_w^{b_{k+1}}\right)\\
&\overset{(83)}{=} \text{Log}\left(\text{Exp}\left(\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w})\delta\hat{\boldsymbol{\phi}}_{b_k}^{w}\right)\mathbf{R}_{b_{k+1}}^{b_k}\hat{\mathbf{R}}_{b_k}^{w}\hat{\mathbf{R}}_w^{b_{k+1}}\right)\\
&\overset{(80)}{\approx} \delta\boldsymbol{\phi}_{b_{k+1}}^{b_k} + \mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w})\delta\hat{\boldsymbol{\phi}}_{b_k}^{w}
\end{aligned} \tag{84}$$

$$\begin{aligned}
&\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}\left(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}\right)\\
&= \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k}\hat{\mathbf{R}}_{b_{k+1}}^{w}\text{Exp}\left(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w} + \delta\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}\right)^{-1}\right)\\
&\overset{(79)}{\approx} \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k}\hat{\mathbf{R}}_{b_k}^{w}\hat{\mathbf{R}}_{w}^{b_{k+1}}\text{Exp}\left(-\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w})\delta\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}\right)\right)\\
&\overset{(83)}{=} \text{Log}\left(\text{Exp}\left(-\text{Exp}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w})\delta\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}\right)\right.\\
&\quad\left.\cdot\,\text{Exp}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\right)\\
&\overset{(80)}{\approx} \delta\boldsymbol{\phi}_{b_{k+1}}^{b_k} - \mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\text{Exp}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w})\delta\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}.
\end{aligned} \tag{85}$$

Finally, Jacobians of $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ in (46) are computed by

$$\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\boldsymbol{\phi}}_{b_k}^{w}} = \mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_k}^{w}) \tag{86}$$

$$\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}} = \mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\text{Exp}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{J}_l(\hat{\boldsymbol{\phi}}_{b_{k+1}}^{w}). \tag{87}$$

B. Jacobians w.r.t. Gyroscope Biases

The equation (45) shows the small change in the gyroscope bias $\hat{\mathbf{b}}_{\omega_{b_k}}$. Hence, in (47), the small changes in $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ w.r.t. $\hat{\mathbf{b}}_{\omega_{b_k}}$ is

$$\begin{aligned}
&\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}\left(\hat{\mathbf{b}}_{\omega_{b_k}} + \Delta\mathbf{b}_{\omega_{b_k}}\right)\\
&\quad= \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k}(\hat{\mathbf{b}}_{\omega_{b_k}} + \Delta\mathbf{b}_{\omega_{b_k}})\hat{\mathbf{R}}_{b_k}^{w}\hat{\mathbf{R}}_w^{b_{k+1}}\right)\\
&\quad\overset{(45)}{\approx} \text{Log}\left(\mathbf{R}_{b_{k+1}}^{b_k}\text{Exp}\left(\mathbf{J}_{\mathbf{b}_\omega}^{\boldsymbol{\phi}}\Delta\mathbf{b}_{\omega_{b_k}}\right)\hat{\mathbf{R}}_{b_k}^{w}\hat{\mathbf{R}}_w^{b_{k+1}}\right)\\
&\quad\overset{(80)}{\approx} \delta\boldsymbol{\phi}_{b_{k+1}}^{b_k} + \mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_{\mathbf{b}_\omega}^{\boldsymbol{\phi}}\Delta\mathbf{b}_{\omega_{b_k}}.
\end{aligned} \tag{88}$$

Then, Jacobians of $\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}$ w.r.t. $\hat{\mathbf{b}}_{\omega_{b_k}}$ in (47) is computed by

$$\frac{\partial\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k}}{\partial\hat{\mathbf{b}}_{\omega_{b_k}}} = \mathbf{J}_l^{-1}(\delta\boldsymbol{\phi}_{b_{k+1}}^{b_k})\mathbf{R}_{b_{k+1}}^{b_k}\mathbf{J}_{\mathbf{b}_\omega}^{\boldsymbol{\phi}}. \tag{89}$$

REFERENCES

[1] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, p. 2068, Apr. 2020.

[2] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[3] D. M. Rosen, K. J. Doherty, A. T. Espinoza, and J. J. Leonard, "Advances in inference and representation for simultaneous localization and mapping," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 4, no. 1, pp. 215–242, May 2021.

[4] M. Zhang, Y. Chen, and M. Li, "Vision-aided localization for ground robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2455–2461.

[5] F. Zheng and Y.-H. Liu, "Visual-odometric localization and mapping for ground vehicles using SE(2)-XYZ constraints," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3556–3562.

[6] Z. Youji, C. Qijun, Z. Hao, L. Dairong, and W. Penghao, "A SLAM method based on LOAM for ground vehicles in the flat ground," in *Proc. IEEE Int. Conf. Ind. Cyber Phys. Syst. (ICPS)*, May 2019, pp. 546–551.

[7] J. Jeong, Y. Cho, and A. Kim, "Road-SLAM: Road marking based SLAM with lane-level accuracy," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1736–1743.

[8] Y. Yang, D. Tang, D. Wang, W. Song, J. Wang, and M. Fu, "Multi-camera visual SLAM for off-road navigation," *Robot. Auto. Syst.*, vol. 128, Jun. 2020, Art. no. 103505.

[9] Z. Zhang, H. Liu, J. Qi, K. Ji, G. Xiong, and J. Gong, "A tightly coupled LiDAR-IMU SLAM in off-road environment," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Sep. 2019, pp. 1–6, doi: 10.1109/ICVES.2019.8906489.

[10] T. Su, H. Zhu, P. Zhao, Z. Li, S. Zhang, and H. Liang, "A robust LiDAR-based SLAM for autonomous vehicles aided by GPS/INS integrated navigation system," in *Proc. 6th Int. Conf. Autom., Control Robot. Eng. (CACRE)*, Jul. 2021, pp. 351–358.

[11] S. Yang, R. Jiang, H. Wang, and S. S. Ge, "Road constrained monocular visual localization using Gaussian–Gaussian cloud model," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3449–3456, Dec. 2017, doi: 10.1109/TITS.2017.2685436.

[12] S. Cho, C. Kim, M. Sunwoo, and K. Jo, "Robust localization in map changing environments based on hierarchical approach of sliding window optimization and filtering," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3783–3789, Apr. 2022, doi: 10.1109/TITS.2020.3035801.

[13] T. Wen, K. Jiang, B. Wijaya, H. Li, M. Yang, and D. Yang, "TM^{3}Loc: Tightly-coupled monocular map matching for high precision vehicle localization," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20268–20281, Nov. 2022, doi: 10.1109/TITS.2022.3176914.

[14] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 683–711, Feb. 2021, doi: 10.1109/TITS.2019.2958352.

[15] 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, doi: 10.1109/TPAMI.2007.1049.

[16] J. Yuan, S. Zhu, K. Tang, and Q. Sun, "ORB-TEDM: An RGB-D SLAM approach fusing ORB triangulation estimates and depth measurements," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022, doi: 10.1109/TIM.2022.3154800.

[17] J. H. Jung et al., "Monocular visual-inertial-wheel odometry using low-grade IMU in urban areas," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 925–938, Feb. 2022.

[18] 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.

[19] 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.

[20] J. Jiang, J. Yuan, X. Zhang, and X. Zhang, "DVIO: An optimization-based tightly coupled direct visual-inertial odometry," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11212–11222, Nov. 2021, doi: 10.1109/TIE.2020.3036243.

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

[22] H. Zhang and C. Ye, "DUI-VIO: Depth uncertainty incorporated visual inertial odometry based on an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5002–5008.

[23] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," *Robot., Sci. Syst.*, vol. 2, no. 9, pp. 1–9, 2014.

[24] J. Zhang and S. Singh, "Visual-LiDAR odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2174–2181.

[25] Z. Wang, J. Zhang, S. Chen, C. Yuan, J. Zhang, and J. Zhang, "Robust high accuracy visual-inertial-laser SLAM system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6636–6641.

[26] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: LiDAR-inertial-camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5848–5854.

[27] X. Zuo et al., "LIC-Fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5112–5119.

[28] 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.

[29] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.

[30] H. A. Hashim and A. E. E. Eltoukhy, "Landmark and IMU data fusion: Systematic convergence geometric nonlinear observer for SLAM and velocity bias," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3292–3301, Apr. 2022.

[31] H. A. Hashim, "GPS-denied navigation: Attitude, position, linear velocity, and gravity estimation with nonlinear stochastic observer," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 1149–1154.

[32] H. A. Hashim, L. J. Brown, and K. McIsaac, "Nonlinear pose filters on the special Euclidean group SE(3) with guaranteed transient and steady-state performance," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 5, pp. 2949–2962, May 2021.

[33] R. Mahony and T. Hamel, "A geometric nonlinear observer for simultaneous localisation and mapping," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 2408–2415.

[34] H. A. Hashim and F. L. Lewis, "Nonlinear stochastic estimators on the special Euclidean group SE(3) using uncertain IMU and vision measurements," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 12, pp. 7587–7600, Dec. 2021.

[35] C. Bila, F. Sivrikaya, M. A. Khan, and S. Albayrak, "Vehicles of the future: A survey of research on safety issues," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1046–1065, May 2017, doi: 10.1109/TITS.2016.2600300.

[36] S.-W. Kim et al., "Autonomous campus mobility services using driverless taxi," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3513–3526, Dec. 2017, doi: 10.1109/TITS.2017.2739127.

[37] A. Hata and D. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer LiDAR," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420–429, Feb. 2016, doi: 10.1109/TITS.2015.2477817.

[38] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 2022, doi: 10.1109/TITS.2020.3024655.

[39] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proc. 5th Int. Conf. Inf. Fusion*, vol. 2, 2002, pp. 901–904.

[40] G. Hao, Y. Li, M. Zhao, H. Li, and Y. Dou, "Covariance intersection fusion particle filter for nonlinear systems," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 5501–5504.

[41] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM Rev.*, vol. 6, no. 4, pp. 422–430, 1964.

[42] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 24–28.

[43] R. Valencia, "Mapping, planning and exploration with pose SLAM," Ph.D. dissertation, Polytechnic Univ. Catalonia, Barcelona, Spain, 2013.

[44] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[45] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," Chapel Hill, Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep., TR04–004, 2004, vol. 4, no. 10, pp. 1–3.

[46] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auto. Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.

[47] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[48] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[49] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution LiDAR and camera in targetless environments," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7517–7524, Jul. 2021.

[50] Q. Sun, J. Yuan, X. Zhang, and F. Duan, "Plane-edge-SLAM: Seamless fusion of planes and edges for SLAM in indoor environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 2061–2075, Oct. 2021.

[51] Q. Sun, J. Yuan, X. Zhang, and Y. Gao, "PLVO: Plane-line-based RGB-D visual odometry," *Acta Autom. Sinica*, May 2021, doi: 10.16383/j.aas.c200878.

[52] S. J. Julier and J. K. Uhlmann, "Using covariance intersection for SLAM," *Robot. Auto. Syst.*, vol. 55, no. 1, pp. 3–20, Jan. 2007.

[53] L. Chen, P. O. Arambel, and R. K. Mehra, "Fusion under unknown correlation—Covariance intersection as a special case," in *Proc. 5th Int. Conf. Inf. Fusion. (FUSION)*, vol. 2, 2002, pp. 905–912.
[54] W. Li, Z. Wang, G. Wei, L. Ma, J. Hu, and D. Ding, "A survey on multisensor fusion and consensus filtering for sensor networks," *Discrete Dyn. Nature Soc.*, vol. 2015, pp. 1–12, Oct. 2015.
[55] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao, "Sequential covariance intersection fusion Kalman filter," *Inf. Sci.*, vol. 189, pp. 293–309, Apr. 2012.
[56] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. Congr. Evol. Comput.*, vol. 1, Portland, OR, USA, Jun. 2004, pp. 325–331.
[57] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, Jul. 2018.
[58] Z. Shan, R. Li, and S. Schwertfeger, "RGBD-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, p. 2251, May 2019.
[59] R. Mur-Artal and J. D. Tardós, "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.

Yuanxi Gao received the B.Eng. degree in intelligence science and technology from Nankai University, Tianjin, China, in 2019, where he is currently pursuing the Ph.D. degree in control science and engineering with the Institute of Robotics and Automatic Information System.

His current research interests include multisensor fusion, mobile robot navigation, and SLAM.

Jing Yuan (Member, IEEE) received the B.S. degree in automatic control and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2002 and 2007, respectively.

He was with the College of Computer and Control Engineering, Nankai University, from 2007 to 2018. He is currently a Professor with the College of Artificial Intelligence, Nankai University. His current research interests include robotic control, motion planning, and simultaneous localization and mapping.

Jingqi Jiang received the B.Eng. degree in intelligence science and technology from Nankai University, Tianjin, China, in 2018, and the M.S. degree in control science and engineering from the Institute of Robotics and Automatic Information System, Nankai University, in 2021.

He is currently an Algorithm Engineer with Meituan, Beijing, China. His current research interests include state estimation, multisensor fusion, visual-inertial localization, and mapping in complex environments.

Qinxuan Sun received the B.Sc. degree in electronic information engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2013, and the M.Sc. and Ph.D. degrees in control theory and control engineering from Nankai University, Tianjin, China, in 2016 and 2021, respectively.

Her current research interests include mobile robot navigation and SLAM.

Xuebo Zhang (Senior Member, IEEE) received the B.Eng. degree in automation from Tianjin University, Tianjin, China, in 2002, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2011.

He is currently a Professor with the Institute of Robotics and Automatic Information System and also the Tianjin Key Laboratory of Intelligent Robotics, Nankai University. His research interests include mobile robotics, motion planning, visual servoing, visual sensor networks, and localization and mapping.

Dr. Zhang is an Technical Editor of IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor of the *ASME Journal of Dynamic Systems, Measurement, and Control*.