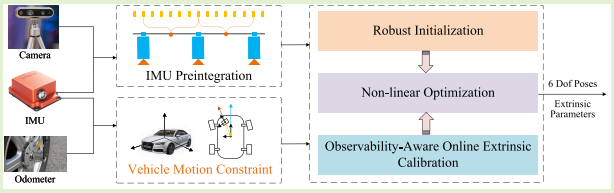
Vehicle-Motion-Constraint-Based Visual-Inertial-Odometer Fusion With Online Extrinsic Calibration

Hang Zhao, Xinchun Ji, and Dongyan Wei, *Member, IEEE*

***Abstract*—Autonomous positioning in complex urban areas is a challenging problem and has attracted increasing attention in academia. Multisensor fusion of camera, inertial measurement unit (IMU), and wheel odometer has been a prevailing solution that is not only low-cost but easy-to-build. Most of the related methods fuse the odometer measurements into the visual-inertial framework by preintegrating the wheel speed into the displacement. However, they are not very suitable for some scenarios where the vehicle velocity changes frequently. Instead of preintegrating the odometer measurements, in this article, we present an efficient and practical visual-inertial-odometer approach that fuses the wheel speed into the visual-inertial framework *directly* based on the vehicle motion constraint. Specifically, we use the nonholonomic constraint (NHC) and lever-arm compensation to introduce the vehicle velocity measurement into the fusion framework to limit the drift. Meanwhile, the proposed framework also allows to online calibration of the IMU-odometer extrinsic parameters (EPs) explicitly. Moreover, we develop an observability-aware method to enhance the system stability and the performance of online extrinsic calibration. We also develop a robust initialization method to obtain the initial values of the visual-inertial-odometer system in one-step optimization. Our approach is validated extensively in simulation environments, autonomous driving public datasets, and real-world experiments. The simulation results show the extrinsic calibration error is within 0.1°/0.02 m for rotation and translation. The public dataset and real-world ground robot experiments show a 0.3% position error in both the 4-km long urban area route and the 403-m long park route.**



***Index Terms*— Extrinsic calibration, robust initialization, vehicle motion constraint, visual-inertial-odometer fusion.**

I. Introduction and Related Work

AUTONOMOUS positioning of ground vehicles in complex urban environments has attracted increasing attention in academia in recent years. Global Navigation Satellite System (GNSS) can provide accurate and continuous absolute position information in the open areas of the city, but cannot work well in the GNSS degraded environments such as urban valleys, tunnels, and underground garages. The method of multisensor fusion is a prevailing solution to improve the positioning accuracy and robustness of ground vehicles in such environments. Particularly, the fusion of a camera, inertial measurement unit (IMU), and odometer (also known as wheel encoder) is regarded as a low-cost and easy-to-build solution for ground vehicles. In such a fusion framework, a good initialization for the parameters is significant since the system is nonlinear and sensitive to the initial values. In addition, high-performance positioning relies on the precise extrinsic parameters (EPs) between the sensors. Otherwise, the additional errors would be introduced and the overall performance would be degraded. A lot of research on the fusion of the above-mentioned sensors has been developed.

The visual-inertial fusion framework is proposed first in academia and the research on it has been maturing in recent years. In general, there are two main technical routes in academia for visual-inertial fusion: filtering-based methods such as MSCKF [1], [2], ROVIO [3], [4] are developed based on the Kalman Filter, optimization-based methods such as OKVIS [5], VINS-Mono [6], ORB-SLAM3 [7], VI-DSO [8] usually adopt the form of maximum a posteriori (MAP) and nonlinear least square to estimate the variables. Typically, the filtering-based methods have better computational efficiency

Manuscript received 7 August 2023; revised 17 September 2023; accepted 19 September 2023. Date of publication 4 October 2023; date of current version 14 November 2023. The associate editor coordinating the review of this article and approving it for publication was Prof. Bin Gao. *(Corresponding author: Xinchun Ji.)*

Hang Zhao is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China, and also with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: zhh2005757@hotmail.com).

Xinchun Ji is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China, and also with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: jixc@aircas.ac.cn).

Dongyan Wei is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: weidy@aircas.ac.cn).

Digital Object Identifier 10.1109/JSEN.2023.3319345

while the optimization-based methods have better accuracy. Since the optimization methods are sensitive to the initial values, a good initialization is necessary before entering the optimization process. To obtain good initial values of the estimator, VINS-Mono proposes an efficient step-by-step initialization method [9] whereas ORB-SLAM3 proposes a MAP-based initialization algorithm which has a higher performance but a lower success rate [10]. Research on camera-IMU extrinsic calibration has also been developed. Methods that use the calibration board can improve the accuracy of calibration dramatically, where the open source calibration tool *Kalibr* can be considered as the representative of a great integration of work [11], [12], [13], [14], [15]. Considering the use of a calibration board is complex and time-consuming, online calibration methods [16], [17] are developed to estimate the EPs when the system is running, which does not require a calibration board and has been prevailing in academia.

For ground vehicles, such as self-driving cars and ground robots, it is a good way to improve positioning performance by coupling the measurement of the odometer into a visual-inertial system. By referring to VINS-Mono, a lot of research adopts the optimization-based method and preintegration framework to fuse the odometer measurement as just IMU does. Some of them preintegrate both the forward velocity and the angular velocity from the odometer measurements [18], [19], [20], [21]. These approaches require that the vehicle runs on a smooth plane, otherwise, the turning measurement may not be accurate [22]. Others use only the forward velocity output of the wheel encoder and preintegrate it with gyroscope measurements [23], [24]. They have more sensitive orientation measurements from the gyroscope but are affected more seriously by the gyroscope bias instead. Based on the preintegration framework, the odometer measurement can be used to improve the performance of system initialization [20], [23]. Furthermore, many approaches utilize the preintegration framework to solve the online odometer extrinsic calibration problem such as [20], [23], and [25].

However, since the preintegration theory is implemented by numerical integration methods such as midpoint, it requires the integration variable (linear acceleration for IMU preintegration and wheel speed for odometer preintegration) to change relatively slowly, or a high sample rate of the wheel encoder. Otherwise, the numerical integration error can not be neglected. It means that the preintegration of the odometer measurement would introduce more errors than that of IMU, especially in circumstances where the vehicle drives at a series of frequently changing speeds such as starting and stopping. We study the change of the odometer preintegration error with respect to the actual driving scenarios in the urban areas from Kaist dataset sequence urban26 [26] as Fig. 1. It can be seen obviously that the odometer preintegration error increases sharply with the vehicle starting and stopping, which will introduce extra errors into the fusion system.

To solve this problem, we focus on the vehicle motion constraint, which is commonly referred to as nonholonomic

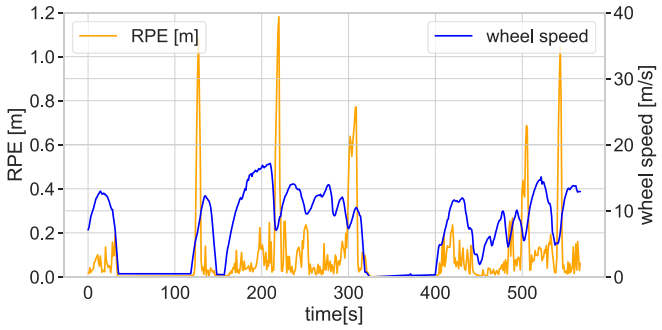


Fig. 1. Relative pose error of the odometer preintegration in every second with the change of the wheel speed.

constraint (NHC) and lever-arm compensation and adopted widely in IMU/odometer system [27] and GNSS/IMU system [28] in the filtering-based framework. NHC means that the lateral and vertical velocity of the vehicle can be regarded as *zero* if there is no slippery or bumping, and the correct utilization of NHC requires compensation of lever-arm between the IMU and the vehicle body [28]. It is noted that the vehicle motion constraint manages to build a measurement model based on the velocity, which means that it can limit the velocity estimation error directly and does not need to integrate into the displacement. Inspired by this, different from the use of preintegration, in this article, we utilize the velocity measurements from vehicle motion constraint and odometer to build an optimization-based visual-inertial odometer framework, which is concise and effective. Based on our proposed framework, robust initialization, and online IMU-odometer calibration are also studied to improve the performance. Our major contribution lies as follows.

1) A tightly-coupled visual-inertial-odometer framework with robust initialization based on vehicle motion constraint is proposed. This framework couples the odometer velocity directly and does not require to preintegrate the odometer measurements, which can improve the performance in scenarios where the speed changes frequently.
2) A coarse-to-fine IMU-odometer extrinsic calibration method is designed based on the proposed framework. It allows us to obtain the initial IMU-odometer EPs without any prior knowledge of the initialization procedure and finely calibrate them online in back-end optimization. The calibration method is explicit and does not require the linearization in preintegration framework, resulting in a better performance.
3) An observability-aware online extrinsic calibration process is proposed. It allows us to determine the beneficial conditions of the IMU-odometer extrinsic calibration by vehicle motion observability judgment which can improve the convergency of parameters and system robustness.

Finally, a variety of experiments in a simulation environment, autonomous driving public dataset, and real-world are performed to validate the proposed method with respect to

the performance of initialization, extrinsic calibration, and trajectory accuracy.

II. Preliminaries

A. Notations and Frames

For the clarity of the article, we now introduce the notations and frame definitions first. The world frame is denoted as $(\cdot)^w$ where the gravity is aligned with its z-axis. It coincides with the coordinate system of the first camera keyframe as [6] dose. $(\cdot)^b$ represents the IMU frame and $(\cdot)^c$ is the camera frame. $(\cdot)_c^v$ represents the camera frame in the visual reference frame. The odometer frame is denoted as $(\cdot)^o$. We define that the odometer frame adopts the right-hand coordinate in which the x-axis is forward, y-axis is left and z-axis is up in this article. We use Hamilton quaternions $\mathbf{q}$ to represent rotation as well as rotation matrices $\mathbf{R}$. We denote $\mathbf{q}_c^b$ and $\mathbf{p}_c^b$ as the rotation and translation from the camera frame to the IMU frame, whereas $\mathbf{q}_o^b$ and $\mathbf{p}_o^b$ are the rotation and translation from the odometer frame to IMU frame, which are known as EPs. o_k represents the odometer frame while taking the kth image. c_k and b_k are in a similar way. $\hat{(\cdot)}$ represent the noisy measurement or estimation of a certain quantity. Finally, our methods assume that the camera-IMU EPs $\mathbf{q}_c^b$ and $\mathbf{p}_c^b$ are calibrated beforehand, thus we focus only on the calibration problem of $\mathbf{q}_o^b$ and $\mathbf{p}_o^b$ between IMU and odometer.

B. NHC and Lever-Arm Compensation

As mentioned above, NHC can be modeled as $\mathbf{v}_y^o = 0$, $\mathbf{v}_z^o = 0$. Actually, the real value $\hat{\mathbf{v}}_y^o = 0, \hat{\mathbf{v}}_z^o = 0$ cannot be exactly zero when driving on the roads with different conditions. The Gaussian noise $\delta\mathbf{v}_y^o$, $\delta\mathbf{v}_z^o$ are introduced to model such a kind of variety, thus we have

$$\begin{aligned} \mathbf{v}_y^o &= \hat{\mathbf{v}}_y^o - \mathbf{n}_y \\ \mathbf{v}_z^o &= \hat{\mathbf{v}}_z^o - \mathbf{n}_z \end{aligned} \tag{1}$$

where $\mathbf{n}_y \sim \mathcal{N}(0, \sigma_y^2)$, $\mathbf{n}_z \sim \mathcal{N}(0, \sigma_z^2)$. Besides, NHC also means that the velocity at the origin of the vehicle body frame is always consistent with the body forward velocity (wheel speed in this article). For differential steering vehicles, the origin of the body frame is at the center of the body, while that of Ackermann steering vehicles is at the midpoint of the rear wheels. Furthermore, the velocity of other sensors (e.g., IMU) that are not positioned at the origin of the body frame would be influenced by the translation of the EPs, which is referred to as the lever-arm effect. Lever-arm compensation indicates the coordinate transformation between the IMU frame and odometer frame, which is shown in Fig. 2. Considering the odometer output is exactly the forward velocity of the vehicle, we regard the odometer frame as the vehicle body frame itself. Based on the NHC and lever-arm effect compensation, we can derive the relationship of the velocity between the IMU frame and odometer frame

$$\mathbf{v}^b = \mathbf{R}_o^b \mathbf{v}^o - \boldsymbol{\omega}^b \times \mathbf{p}_o^b \tag{2}$$

where $\mathbf{v}^b$ is the IMU velocity in the IMU frame, $\mathbf{v}^o = \{\mathbf{v}_x^o, \mathbf{v}_y^o, \mathbf{v}_z^o\}$ is the velocity of the vehicle, and $\mathbf{v}_x^o$ is the nominal value of odometer measurements which is given by

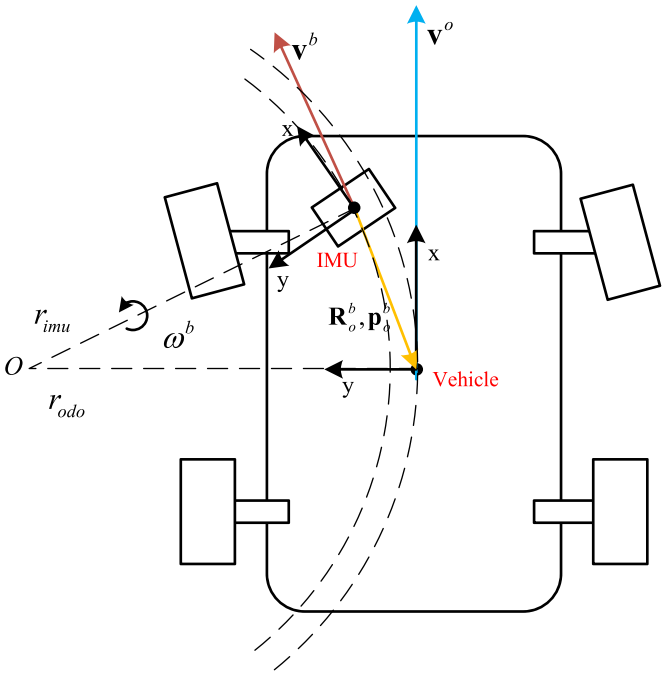


Fig. 2. Lever-arm compensation between IMU and the vehicle body. When the vehicle turns, the IMU and odometer have different velocities $\mathbf{v}^b$ and $\mathbf{v}^o$. Therefore, the odometer velocity should be transformed from the vehicle frame to the IMU frame by their EPs $(\mathbf{R}_o^b, \mathbf{p}_o^b)$ to realize the fusion.

$$\mathbf{v}_x^o = \hat{\mathbf{v}}_x^o - \mathbf{n}_v \tag{3}$$

where $\hat{\mathbf{v}}_x^o$ is the raw output of the odometer, $\mathbf{n}_v$ is assumed as Gaussian noise, $\mathbf{n}_v \sim \mathcal{N}(0, \sigma_v^2)$. $\boldsymbol{\omega}^b$ is the nominal value of gyroscope measurements which can be obtained from its raw measurements $\hat{\boldsymbol{\omega}}^b$

$$\boldsymbol{\omega}^b = \hat{\boldsymbol{\omega}}^b - \mathbf{b}_\omega - \mathbf{n}_\omega \tag{4}$$

where $\mathbf{n}_\omega$ is the Gaussian white noise modeled as $\mathbf{n}_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$. $\mathbf{b}_\omega$ is the gyroscope bias whose derivative is Gaussian white noise, $\dot{\mathbf{b}}_\omega = \mathbf{n}_{b_\omega}$, where $\mathbf{n}_{b_\omega} \sim \mathcal{N}(0, \sigma_{b_\omega}^2)$.

III. Methods

The system structure is shown in Fig. 3. First of all, the system starts with the initialization. In static initialization, we consider that the vehicle usually parks on an approximate flat surface. Thus, we use raw IMU measurements to initialize the gravity direction, gyroscope bias, and pitch component of the EPs in the static state of the vehicle. After the vehicle starts, the dynamic initialization is performed. Visual features are extracted and tracked to perform the vision-only structure from motion to obtain the visual up-to-scale poses. Meanwhile, IMU measurements are preintegrated at the time of the visual frames. Then, the visual, inertial, and odometer measurements are aligned to estimate the visual scale and the yaw component of the EPs by a nonlinear least square estimator (in Section III-A). When the system enters the back-end optimization, we construct the visual reprojection residual, IMU preintegration residual, and vehicle

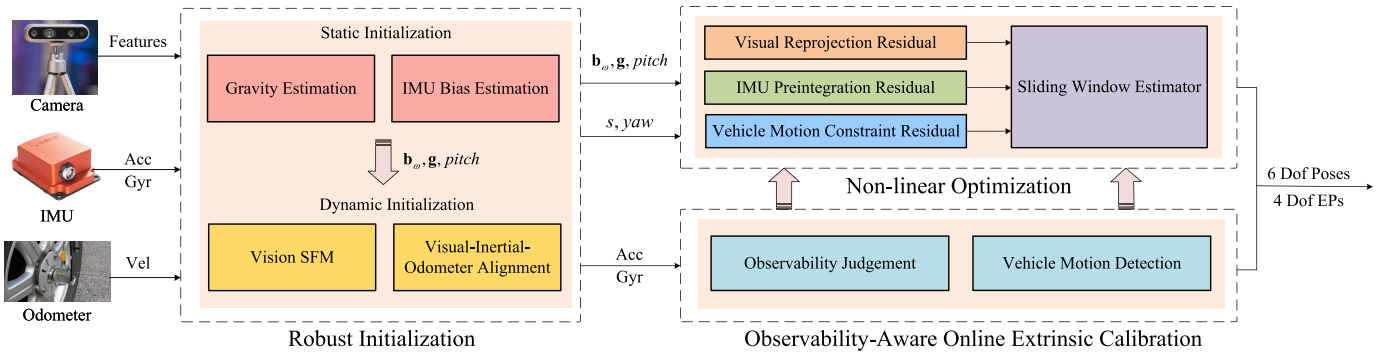


Fig. 3. System overview. Acc represents the linear acceleration, Gyr represents the angular velocity, and Vel represents the odometer velocity. EPs means IMU-odometer EPs, including *pitch* and *yaw* components of the orientation as well as *x*–*y* of the translation. *s* represents the monocular visual scale.

motion constraint residual to build a nonlinear sliding window estimator so as to estimate 6 DoF poses and 4 DoF EPs iteratively, including pitch, yaw, and x- and y-axis translation (in Section III-B). In the optimization process, to improve the performance of online extrinsic calibration, we propose an observability-aware method (in Section III-C1) based on the observability judgment and vehicle motion detection algorithm (in Section III-C2).

A. Robust Initialization

We propose a visual-inertial-odometer joint initialization method to estimate initial parameters, including the monocular visual scale, gravity, IMU bias, velocities, and pitch and yaw components of the EPs. Thanks to the fusion with odometer measurement, the monocular visual scale recovery and the initial estimation of other parameters can be improved well. Considering two typical scenarios of ground vehicles on the road, parking, and driving, we propose a step-by-step initialization method by utilizing both the static and dynamic conditions in the above scenarios well to improve the robustness and effectiveness.

1) Static Initialization for Gravity, Gyroscope Bias, and Pitch of EP: The static state of the vehicle when parking is utilized to estimate the gravity vector as well as the rotation matrix from IMU frame to world frame $\mathbf{R}_b^w$. We collect the accelerometer data in the parking state and use its average value as the initial gravity vector in the IMU frame, which is denoted as $\mathbf{g}_{\text{init}}^b$. The gravity vector in the world frame is defined as $\mathbf{g}^w = [0, 0, G]^T$, where G is the magnitude of the local gravity. By Rodrigues' rotation formula [29], $\mathbf{R}_b^w$ can be calculated by $\mathbf{g}_{\text{init}}^b$ and $\mathbf{g}^w = [0, 0, G]^T$, thus, we have

$$\mathbf{R}_b^w = \cos\boldsymbol{\theta} I + (1 - \cos\boldsymbol{\theta})\mathbf{u}\mathbf{u}^T + \sin\boldsymbol{\theta}\mathbf{u}^\wedge \tag{5}$$

where

$$\mathbf{u} = \frac{\mathbf{g}_{\text{init}}^b \times \mathbf{g}^w}{\left\|\mathbf{g}_{\text{init}}^b \times \mathbf{g}^w\right\|}, \quad \boldsymbol{\theta} = \text{atan2}\left(\left\|\mathbf{g}_{\text{init}}^b \times \mathbf{g}^w\right\|, \mathbf{g}_{\text{init}}^b \cdot \mathbf{g}^w\right). \tag{6}$$

Then, the actual gravity vector $\mathbf{g}^b$ in the IMU frame can be calculated by

$$\mathbf{g}^b = \mathbf{R}_b^{wT}\mathbf{g}^w. \tag{7}$$

Furthermore, we assume that the vehicle parks on an approximately flat surface in general in this article, thus the roll and pitch components of $\mathbf{R}_b^w$ can be regarded as the same part of the IMU-odometer EPs. As a result, $\mathbf{R}_o^b$ can be written as

$$\mathbf{R}_o^b = \mathbf{R}_b^{wT}\mathbf{R}_o^w \tag{8}$$

where $\mathbf{R}_o^w$ represents the yaw component of $\mathbf{R}_o^b$ which will be estimated in the next steps.

In addition, we initially estimate $\mathbf{b}_\omega$ along with the gravity by obtaining its average value $\bar{\omega}$ since the output of the gyroscope should be zero in the static state

$$\mathbf{b}_\omega = \bar{\omega}. \tag{9}$$

Then, we use the estimated $\mathbf{b}_\omega$ to preintegrate IMU measurements in the subsequent process. Compared with gyroscope bias initialization from VINS-Mono, our method is more suitable for the vehicle because the rotation movement is uncommon when the vehicle starts on a straight line but it is necessary in VINS-Mono for robust estimation.

2) Dynamic Initialization for Visual Scale, IMU Velocity, and Yaw of EP: When the vehicle starts driving, we perform dynamic initialization to obtain the visual scale and initial yaw component of the EP. We do not estimate the acceleration bias in this article since there is usually not enough motivation during the initialization. First of all, we collect several visual frames in a sliding window $\mathcal{W}$ to limit the computation cost and do preintegration with initial IMU bias as [30]. We assume that the IMU bias is constant in the initialization process. Meanwhile, the initial up-to-scale camera poses $(\bar{\mathbf{p}}_{c_k}^v, \mathbf{q}_{c_k}^v, k \in \mathcal{W})$ are recovered by vision-only structure as [9] dose. Then, we perform visual-inertial-odometer alignment to recover the initial visual scale, IMU velocity, and yaw component of the orientation.

First, the visual measurements are aligned with IMU preintegration by initial camera poses and visual scale parameter s. Considering two consecutive frames b_k and b_{k+1} in the sliding window, we have the following expressions of position

preintegration $\boldsymbol{\alpha}_{b_{k+1}}^{b_k}$ and velocity preintegration $\boldsymbol{\beta}_{b_{k+1}}^{b_k}$

$$\begin{aligned}\boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \mathbf{q}_v^{b_k}\left(s\left(\bar{\mathbf{p}}_{b_{k+1}}^v - \bar{\mathbf{p}}_{b_k}^v\right) + \mathbf{q}_{b_k}^v v_{b_k}^{b_k}\Delta t_k - \frac{1}{2}\mathbf{q}_c^{b^{-1}}\mathbf{g}^b\Delta t_k^2\right)\\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \mathbf{q}_v^{b_k}\left(\mathbf{q}_{b_{k+1}}^v v_{b_{k+1}}^{b_{k+1}} - \mathbf{q}_{b_k}^v v_{b_k}^{b_k} + \mathbf{q}_c^{b^{-1}}\mathbf{g}^b\Delta t_k\right)\end{aligned} \tag{10}$$

where

$$\begin{aligned}\mathbf{q}_{b_k}^v &= \mathbf{q}_{c_k}^v \otimes \mathbf{q}_c^{b^{-1}}\\ s\bar{\mathbf{p}}_{b_k}^v &= s\bar{\mathbf{p}}_{c_k}^v - \mathbf{q}_{b_k}^v\mathbf{p}_c^b\\ s\bar{\mathbf{p}}_{b_{k+1}}^v &= s\bar{\mathbf{p}}_{c_{k+1}}^v - \mathbf{q}_{b_{k+1}}^v\mathbf{p}_c^b.\end{aligned} \tag{11}$$

In (11), $\mathbf{q}_{c_k}^v$ and $\bar{\mathbf{p}}_{c_k}^v$ refer to the camera rotation and translation at the kth frame while $\bar{\mathbf{p}}_{c_{k+1}}^v$ refers to the camera translation at the $k+1$th frame. These parameters are all obtained beforehand in a vision-only structure. $\mathbf{q}_v^{b_k}$ is the inverse rotation of $\mathbf{q}_{b_k}^v$ that refers to the IMU rotation at kth frame. Δt_k is the time interval between two consecutive frames.

Then, we align odometer measurements with IMU measurements. Equation (2) indicates the relationship between odometer measurements and IMU measurements. Considering (8), (2) can be written as

$$\mathbf{v}^b = \mathbf{R}_b^{wT}\mathbf{R}_o^w\mathbf{v}^o - \boldsymbol{\omega}^b \times \mathbf{p}_o^b. \tag{12}$$

It is worth noting that we do not estimate the translation $\mathbf{p}_o^b$ here since the initialization process usually takes a very short time and there is no large motivation to obtain a good estimation of translation. Therefore, we fix the translation $\mathbf{p}_o^b = [0, 0, 0]$ during the initialization process.

The estimated variables in this kind of initialization process are defined as

$$\mathcal{X} = \left[s, \mathbf{R}_o^w\right]. \tag{13}$$

Substitute (12) with kth frame and $k+1$th frame into (10), we can obtain the visual-inertial-odometer alignment residual

$$\mathbf{r}_{\mathcal{W}}\left(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X}\right) = \left[\delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k}, \delta\boldsymbol{\beta}_{b_{k+1}}^{b_k}\right] \tag{14}$$

$$\begin{aligned}\delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \mathbf{q}_v^{b_k}\Bigg(s\left(\bar{\mathbf{p}}_{b_{k+1}}^v - \bar{\mathbf{p}}_{b_k}^v\right)\\ &\quad + \mathbf{q}_{b_k}^v\left(\mathbf{R}_b^{wT}\mathbf{R}_o^w\mathbf{v}^{o_k} - \boldsymbol{\omega}^{b_k} \times \mathbf{p}_o^b\right)\Delta t_k\\ &\quad - \frac{1}{2}\mathbf{q}_c^{b^{-1}}\mathbf{g}^b\Delta t_k^2\Bigg) - \boldsymbol{\alpha}_{b_{k+1}}^{b_k}\\ \delta\boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \mathbf{q}_v^{b_k}\left(\mathbf{q}_{b_{k+1}}^v\left(\mathbf{R}_b^{wT}\mathbf{R}_o^w\mathbf{v}^{o_{k+1}} - \boldsymbol{\omega}^{b_{k+1}} \times \mathbf{p}_o^b\right)\right.\\ &\quad - \mathbf{q}_{b_k}^v\left(\mathbf{R}_b^{wT}\mathbf{R}_o^w\mathbf{v}^{o_k} - \boldsymbol{\omega}^{b_k} \times \mathbf{p}_o^b\right)\\ &\quad \left. + \mathbf{q}_c^{b^{-1}}\mathbf{g}^b\Delta t_k\right) - \boldsymbol{\beta}_{b_{k+1}}^{b_k}.\end{aligned} \tag{15}$$

As we need to estimate the rotation matrix $\mathbf{R}_o^w$, we formulate a nonlinear least square problem according to the above residuals

$$\min_{\mathcal{X}} = \sum_{k\in\mathcal{W}}\left\|\mathbf{r}_{\mathcal{W}}\left(\hat{z}_{b_{k+1}}^{b_k}, \mathcal{X}\right)\right\|^2. \tag{16}$$

The corresponding factor graph is shown in Fig. 4. Ceres solver [31] C++ library is used to implement and solve the above optimization problem.

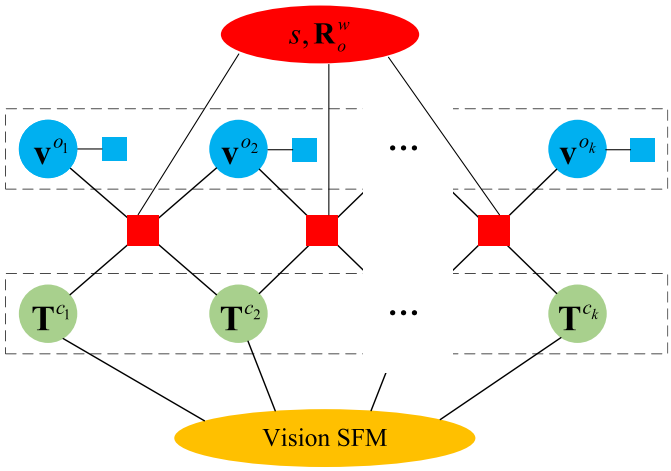


Fig. 4. Visual-inertial-odometer alignment factor graph. Green circles represent the camera poses from the vision-only structure. Blue circles represent the vehicle velocity from the odometer and the boxes with the same color represent the $\mathbf{b}_\omega$, $\mathbf{g}$, and *pitch* of EP. Red boxes represent the visual-inertial-odometer alignment residual with the estimated variables s and $\mathbf{R}_o^w$. Dashed lines indicate the fixed variables, of which the value we have obtained before.

It can be seen that the rotation matrix $\mathbf{R}_o^w$ only represents the yaw component of the orientation. Therefore, we need to define a variable update rule to ensure that only the yaw component can be optimized in the manifold. Similar to [10], we have

$$\hat{\mathbf{R}}_o^w = \mathbf{R}_o^w\mathbf{Exp}(0, 0, \delta\phi) \tag{17}$$

where **Exp** represents the exponential map of the Lie Group and $\delta\phi$ indicates the minor change of the yaw component during the optimization.

In addition, to ensure that the visual scale s is always positive during the optimization, we utilize the attribute of the exponential function, thus the update rule is defined as

$$\hat{s} = s\exp(\delta s) \tag{18}$$

where δs represents the minor change of visual scale during the optimization.

After the estimator converges, the IMU-odometer extrinsic rotation matrix $\mathbf{R}_o^b$ can be calculated by (8) with $\mathbf{R}_b^w$ and the optimal $\mathbf{R}_o^w$. Then, it will become the initial value in the nonlinear optimization process. In addition, the IMU velocities $\mathbf{v}_{b_k}^{b_k}$ can also be recovered by (12).

Overall, since the wheel speed is accurate in a short time, the proposed method can not only enhance the initialization performance but also improve the robustness by reducing the estimated variables compared with VINS-Mono. The public dataset experiments are performed to evaluate our method in Section IV-B.

B. Nonlinear Optimization

Since the IMU is fixed on the vehicle, the variation of the orientation can be considered consistent at both of the two frames. It is necessary to indicate that the roll component of $\mathbf{R}_o^b$ is unobservable cause one wheel encoder odometer can

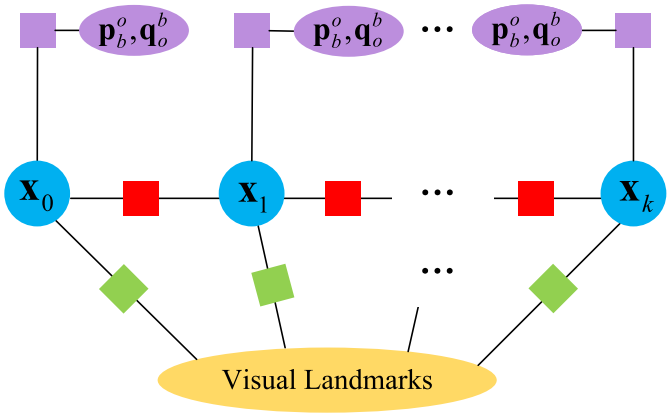


Fig. 5. Visual-inertial-odometer optimization factor graph. The blue circles represent the IMU state. The green boxes represent the visual reprojection residuals, the red ones represent the IMU preintegration residuals, and the purple ones represent the vehicle motion constraint residuals with the estimated IMU-odometer EPs $\mathbf{p}_o^b$ and $\mathbf{q}_o^b$.

only provide the forward velocity of the vehicle, as illustrated by

$$\begin{bmatrix} \mathbf{v}_x^o \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \mathbf{v}_x^o \\ 0 \\ 0 \end{bmatrix} \tag{19}$$

where ψ represents the roll angle of $\mathbf{R}_o^b$. It can be seen that the rotation matrix of the roll angle dose not affect the odometer velocity vector in any way. Therefore, only the pitch and yaw components of $\mathbf{R}_o^b$ can be estimated in the nonlinear optimization. In addition, it is obvious that the z-axis of the translation from the vehicle frame to the IMU frame, $\mathbf{p}_b^o$, is also unobservable since there is no vertical motion for the ground vehicles in general. In order to eliminate the impact of unobservability, we estimate $\mathbf{p}_b^o$ instead of $\mathbf{p}_o^b$. And its relationship regarding to $\mathbf{p}_o^b$ can be written as

$$\mathbf{p}_o^b = -\mathbf{R}_o^b \mathbf{p}_b^o. \tag{20}$$

Similar to [6], we formulate a MAP estimation problem to realize the visual, inertial, and odometer fusion as Fig. 5. Meanwhile, the full-state vector is defined as

$$\begin{aligned} \mathcal{X} &= \left[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_o^b, \lambda_0, \lambda_1, \ldots, \lambda_m\right] \\ \mathbf{x}_k &= \left[\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_\omega\right], \quad k \in [0, n] \\ \mathbf{x}_o^b &= \left[\mathbf{p}_b^o, \mathbf{q}_o^b\right] \end{aligned} \tag{21}$$

where $\mathbf{x}_k$ is the IMU state corresponding to the kth key-frame, including positions, velocities, orientations of IMU in the world frame, accelerometer and gyroscope bias. n is the total number of the key-frames in the sliding window. m is the total number of the features. λ_l is the inverse depth of the lth feature in the first observation. $\mathbf{x}_o^b$ is the IMU-odometer EP including translation $\mathbf{p}_o^b$ and the rotation $\mathbf{q}_o^b$.

Then, we minimize the measurement residuals weighted by Mahalanobis distance to solve the MAP estimation problem

$$\min_{\mathcal{X}} \left\{ \left\| \mathbf{r}_p - \mathbf{H}_p \mathcal{X} \right\|^2 + \sum_{k \in \mathcal{B}} \left\| \mathbf{r}_{\mathcal{B}}\left(\hat{z}_{b_{k+1}}^{b_k}, \mathcal{X}\right) \right\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2 + \sum_{(l,j) \in \mathcal{C}} \rho\left(\left\| \mathbf{r}_{\mathcal{C}}\left(\hat{z}_l^{c_j}, \mathcal{X}\right) \right\|_{\mathbf{P}_l^{c_j}}^2 \right) \right\} \tag{22}$$

where

$$\begin{aligned} &\mathbf{r}_{\mathcal{B}}\left(\hat{z}_{b_{k+1}}^{b_k}, \mathcal{X}\right) \\ &= \begin{bmatrix} \delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \delta\boldsymbol{\gamma}_{b_{k+1}}^{b_k} \\ \delta\mathbf{b}_a \\ \delta\mathbf{b}_\omega \\ \delta\mathbf{v}_{\text{odo}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_w^{b_k}\left(\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 - \mathbf{v}_{b_k}^w \Delta t_k\right) - \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} \\ \mathbf{R}_w^{b_k}\left(\mathbf{v}_{b_{k+1}}^w + \mathbf{g}^w \Delta t_k - \mathbf{v}_{b_k}^w\right) - \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \\ 2\left[\mathbf{q}_{b_k}^{w\,-1} \otimes \mathbf{q}_{b_{k+1}}^w \otimes \left(\hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k}\right)^{-1}\right]_{xyz} \\ \mathbf{b}_{a_{b_{k+1}}} - \mathbf{b}_{a_{b_k}} \\ \mathbf{b}_{\omega_{b_{k+1}}} - \mathbf{b}_{\omega_{b_k}} \\ \mathbf{R}_o^b \hat{\mathbf{v}}_{b_{k+1}}^o + \left(-\mathbf{R}_o^b \mathbf{p}_b^o\right) \times \left(\hat{\boldsymbol{\omega}}^{b_{k+1}} - \mathbf{b}_{\omega_{b_{k+1}}}\right) - \mathbf{R}_w^{b_{k+1}} \mathbf{v}_{b_{k+1}}^w \end{bmatrix} \end{aligned} \tag{23}$$

is the vehicle motion constraint residual, which consists of the IMU preintegration residual $\delta\boldsymbol{\alpha}_{b_{k+1}}^{b_k}$, $\delta\boldsymbol{\beta}_{b_{k+1}}^{b_k}$, $\delta\boldsymbol{\gamma}_{b_{k+1}}^{b_k}$, $\delta\mathbf{b}_a$, $\delta\mathbf{b}_\omega$, and odometer lever-arm compensation residual $\delta\mathbf{v}_{\text{odo}}$. The IMU preintegration residual is derived in detail from [6], whereas the odometer residual can be derived from (2) [32]. Since the odometer lever-arm compensation residual is correlated to the IMU measurement, we construct their residuals together. $\mathbf{g}^w$ represents the gravity vector in the world frame. Δt_k represents the time interval between kth and $k+1$th frame. $\hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}$, $\hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}$, and $\hat{\boldsymbol{\gamma}}_{b_{k+1}}^{b_k}$ are the IMU preintegration with respect to position, velocity, and rotation, respectively. $\mathbf{R}_o^b$ is the rotation matrix of $\mathbf{q}_o^b$. $\hat{\mathbf{v}}_{b_{k+1}}^o$ and $\hat{\boldsymbol{\omega}}^{b_{k+1}}$ represent the real value of the vehicle velocity and the raw output of the gyroscope at the kth key-frame, respectively. $\mathbf{r}_p$ and $\mathbf{r}_{\mathcal{C}}(\hat{z}_l^{c_j}, \mathcal{X})$ are the marginalization prior residual and visual reprojection residual, respectively, which are illustrated in [6]. For the reason of the length, we omit their introduction here.

We also derive the error propagation covariance of the proposed IMU-odometer residual. Since the odometer measurement residual does not adopt the form of preintegration, its noise covariance matrix is built separately, thus the noise covariance matrix of the IMU-odometer residual is defined as

$$\mathbf{P}_{b_{k+1}}^{b_k} = \begin{bmatrix} \mathbf{P}_{\text{imu}} & 0 \\ 0 & \mathbf{P}_{\text{odo}} \end{bmatrix} \tag{24}$$

where $\mathbf{P}_{\text{imu}}$ is the noise covariance matrix of IMU preintegration residual which is the same as illustrated in [6], whereas

the noise covariance matrix of odometer residual $\mathbf{P}_{\text{odo}}$ can be written as

$$\mathbf{P}_{\text{odo}} = \mathbf{R}_o^b \mathbf{Q}_v \mathbf{R}_o^{bT} + \left(\mathbf{p}_o^b\right)^\wedge (\mathbf{Q}_\omega) \left(\left(\mathbf{p}_o^b\right)^\wedge\right)^T \tag{25}$$

where $(\mathbf{p}_o^b)^\wedge$ is the skew-symmetric matrix of $\mathbf{p}_o^b$, $\mathbf{p}_o^b = -\mathbf{R}_o^b \mathbf{p}_b^o$. $\mathbf{Q}_\omega = \text{diag}(\sigma_\omega^2, \sigma_\omega^2, \sigma_\omega^2)$ is the noise covariance matrix $\mathbf{n}_\omega$ while $\mathbf{Q}_v = \text{diag}(\sigma_v^2, \sigma_y^2, \sigma_z^2)$ is the noise covariance of $\mathbf{n}_v$. Considering that the NHC can not be completely met when turning at a large angle, means the y-axis NHC noise covariance σ_y^2 will tend to increase. Therefore, we choose to adjust σ_y^2 adaptively according to the motion state of the vehicle. Reference [33] has pointed out that the magnitude of y-axis NHC noise is affected by both the forward velocity and turning motion. Inspired by this, we change σ_y dynamically when turning but set it the same as the x-axis NHC noise $\sigma_x = 0.1$ otherwise, as follows:

$$\sigma_y = \begin{cases} k_{\text{nhc}} \cdot \hat{\mathbf{v}}^o \cdot \left\| \hat{\boldsymbol{\omega}}^b \right\|, & \text{turning} \\ \sigma_x, & \text{not turning} \end{cases} \tag{26}$$

where k_{nhc} is the weight coefficient, which is used to control the scale of the parameter for the different kinds of vehicles. $\hat{\mathbf{v}}^o$ and $\hat{\boldsymbol{\omega}}^b$ are the output of odometer and gyroscope, respectively. In addition, we set a threshold $\sigma_{\text{th}} = \sigma_x$ to constrain the minimum value of σ_y to avoid the *zero* value when parking

$$\sigma_y = \sigma_x, \quad \text{if } \sigma_y <= \sigma_x. \tag{27}$$

Based on our proposed IMU-odometer residual, the EPs can be online calibrated easily with the Jacobeans of $\delta \mathbf{v}_{\text{odo}}$ with respect to $\mathbf{R}_o^b$ and $\mathbf{p}_b^o$

$$\begin{aligned} \mathbf{J}_{\mathbf{R}_o^b} &= -\mathbf{R}_o^b \left(\hat{\mathbf{v}}^{o_{k+1}}\right)^\wedge - \left(\hat{\boldsymbol{\omega}}^{b_{k+1}} - \mathbf{b}_{\omega_{b_{k+1}}}\right)^\wedge \mathbf{R}_o^b \left(\mathbf{p}_b^o\right)^\wedge \\ \mathbf{J}_{\mathbf{p}_b^o} &= \left(\hat{\boldsymbol{\omega}}^{b_{k+1}} - \mathbf{b}_{\omega_{b_{k+1}}}\right)^\wedge \mathbf{R}_o^b. \end{aligned} \tag{28}$$

As mentioned above, the z-axis component of $\mathbf{p}_b^o$ is unobservable as well as the roll component of $\mathbf{p}_o^b$. Thus, we can fix them in the nonlinear optimization process for stability. To ensure that only the pitch, yaw, x, and y component of the EPs are estimated, we define the variable update rules for $\mathbf{R}_o^b$ and $\mathbf{p}_b^o$, respectively, as follows:

$$\begin{aligned} \hat{\mathbf{R}}_o^b &= \mathbf{R}_o^b \mathbf{Exp}(0, \delta\theta, \delta\phi) \\ \hat{\mathbf{p}}_b^o &= \mathbf{p}_b^o + \{\delta x, \delta y, 0\} \end{aligned} \tag{29}$$

where $\delta\theta, \delta\phi, \delta x$, and δy represent the minor change of the above variables during the optimization, respectively. Then, we use the Gauss–Newton or Levenberg–Marquardt method by Ceres solver to optimize the variables.

C. Observability-Aware Online Extrinsic Calibration

As is known to us, extrinsic calibration is not only related to the stability of the system but also the observability of the parameters. In this section, we study the observability of the extrinsic calibration by considering the two kinds of vehicle motion states, parking and turning. Then, we calibrate the EPs according to the observability judgment and above-mentioned motion detection so as to ensure both stability and accuracy.

1) Observability Judgment: Based on the motion characteristics from turning and parking for the ground vehicle, two theorems are proposed for observability judgment.

Theorem 1: Only the pitch and yaw component of $\mathbf{R}_o^b$ (2 DoF) are observable when the vehicle drives on the straight road. Considering the odometer lever-arm compensation residual $\delta \mathbf{v}_{\text{odo}}$ in (23), ignoring the noise impact and let $\delta \mathbf{v}_{\text{odo}} = 0$, we have

$$\mathbf{R}_o^b \hat{\mathbf{v}}_{b_{k+1}}^o + \mathbf{p}_o^b \times \left(\hat{\boldsymbol{\omega}}^{b_{k+1}} - \mathbf{b}_{\omega_{b_{k+1}}}\right) - \mathbf{R}_w^{b_{k+1}} \mathbf{v}_{b_{k+1}}^w = \mathbf{0}. \tag{30}$$

When the vehicle drives on a straight road, the gyroscope output can be regarded as its bias, thus,

$$\hat{\boldsymbol{\omega}}^{b_{k+1}} - \mathbf{b}_{\omega_{b_{k+1}}} = \mathbf{0}. \tag{31}$$

Substitute (31) into (30), we have

$$\mathbf{R}_o^b \hat{\mathbf{v}}_{b_{k+1}}^o - \mathbf{R}_w^{b_{k+1}} \mathbf{v}_{b_{k+1}}^w = \mathbf{0}. \tag{32}$$

It can be seen that (32) only contains the rotation $\mathbf{R}_o^b$ other than the translation $\mathbf{p}_o^b$, which means that only rotation is observable on the straight road. Furthermore, (19) indicates that the roll component of the rotation matrix is also unobservable. Therefore, we enable the calibration of pitch and yaw but disable the calibration of translation before the vehicle turns into the first curve.

Theorem 2: The x and y component of $\mathbf{p}_b^o$ are observable as well as the pitch and yaw component of $\mathbf{R}_o^b$ when the vehicle turns. Since (31) cannot be satisfied when turning and both the error of the rotation and translation can be motivated at the same time, as illustrated in (23). Benefiting from the marginalization of the optimization framework, the translation is still observable after the turning. Therefore, we enable both the extrinsic rotation and translation calibration when the vehicle turns into the first curve and keep them enabled throughout the whole process. It is worth noting that the turning check is significant for not only the stability of the estimation but the convergence of the parameters, which will be introduced next.

Theorem 3: All the EPs are unobservable when parking. When the vehicle is parking or in the static state, the outputs from both the IMU and the odometer are near zero, which means that both the orientation and translation of the EP are unobservable. Therefore, we fix them for the sake of the stability of the estimator as well as the accurate estimation of another parameter such as IMU bias.

2) Motion State Detection: According to the above analysis, it can be known that the vehicle motion state determines the observability of the estimated parameters dramatically. Therefore, correct vehicle motion state detection is crucial to extrinsic calibration. In this article, we mainly study the detection method with respect to the turning state and parking state. For clarity, we denote the current vehicle motion state as $\mathbf{S}$, the turning state as $\mathbf{S}_{\text{turn}}$, the go-straight state as $\mathbf{S}_{\text{line}}$, the parking state as $\mathbf{S}_{\text{park}}$, and the driving state as $\mathbf{S}_{\text{drive}}$, which includes both the turning state and go-straight state.

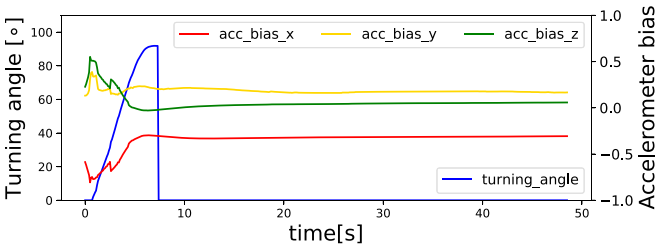
a) Turning state detection: Since the extrinsic calibration introduces extra variables into the estimator as (23), the stability of the system becomes challenging. Some earlier

relevant works such as [19] have pointed out that only when the accelerometer bias converges does the visual-inertial-odometer system become well-constrained, which requires that the vehicle drives through the first curve. To ensure the system stability when calibrating the EPs, we propose an effective and robust turning state detection method in this article, which utilizes both the angular velocity and relative rotation at the same time, as shown in Algorithm 1. First, we set a sliding window $\mathbf{W}_{\text{turn}}$ with a size of $\mathbf{N}_{\text{turn}}$ to collect angular velocities ω. Then, we calculate the average value $\bar{\omega}$ in the sliding window. Given the threshold ω_{th}, the vehicle can be regarded as turning when $\bar{\omega} > \omega_{\text{th}}$. In addition, our experiments show that only when the vehicle turns at a large angle does the accelerometer bias start to converge. Consequently, the relative rotation angle is used to assist the detection. We accumulate the variation of the yaw angle of the relative rotation as $\Delta\psi$ once the turning is detected by the condition of the angular velocity. Given the threshold ψ_{th}, it can be considered that the vehicle has turned at a large angle when $\Delta\psi > \psi_{\text{th}}$. It is noted that only $\bar{\omega}$ meets the condition is the relative rotation angle accumulated, which can ensure that the vehicle is in the turning state all the time. After $\Delta\psi$ meets the condition, the turning state detection can be considered a success.

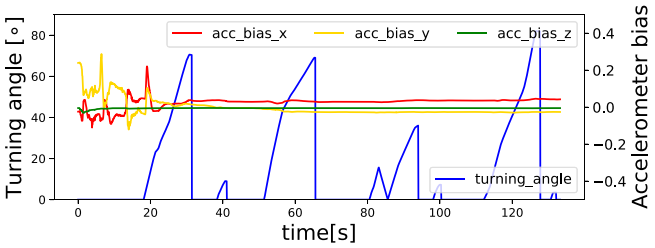
Algorithm 1 Turning State Detection

Input: $\mathbf{W}_{turn}, \omega, \omega_{th}, \Delta\psi, \psi_{th}$
Output: The current motion state $\mathbf{S}$
  while $\mathbf{W}_{turn}$ is full **do**
    Calculate $\bar{\omega}$ from $\mathbf{W}_{turn}$
    if $\bar{\omega} > \omega_{th}$ **then**
      Accumulate the variation of the turning angle as $\Delta\psi$
      if $\Delta\psi > \psi_{th}$ **then**
        $\mathbf{S} \leftarrow \mathbf{S}_{turn}$
      else
        $\mathbf{S} \leftarrow \mathbf{S}_{line}$
      end if
    else
      $\mathbf{S} \leftarrow \mathbf{S}_{line}$
    end if
    Slide the window $\mathbf{W}_{turn}$
  end while

For the threshold parameters ω_{th} and ψ_{th} mentioned above, we determine their value by both the experience and experiments. In our proposed algorithm, the use of angular velocity is aimed at determining the time when the vehicle starts to turn, whereas the variation of the turning angle indicates the magnitude of the turning, which is dramatically related to the convergence of the accelerometer bias. Therefore, to improve the sensitivity of the detection, we set ω_{th} as a smaller value of 0.05 rad/s in our experiment. To determine the value of ψ_{th}, we count the data of $\Delta\psi$ and accelerometer bias in three axes from Kaist dataset sequence urban26 after the angular velocity condition is met, as shown in Fig. 6(a). It can be seen that the bias starts to converge when the turning angle reaches about 35°. Therefore, we set ψ_{th} as 35°, and in this way, the convergence of accelerometer bias can be ensured when detecting the turning. To evaluate the universality of the



(a)



(b)

Fig. 6. Variation of acceleration bias with turning angle in (a) Kaist dataset sequence urban 26 and (b) real world ground robot.

selected threshold, we also collect the experimental data from real-world ground robots as shown in Fig. 6(b). The Kaist dataset data represents the high-speed scenario, whereas the ground robot data illustrates the effectiveness of the parameters in the low-speed scenario. It is easy to see that the selected threshold is still reasonable and effective.

b) Parking state detection: It is worth noting that since the wheel speed varies much in different vehicle platforms, we only utilize the IMU for the parking state detection. Besides, the proposed initialization method requires the vehicle to start with an obvious acceleration change for good estimation, using IMU for parking state detection is more suitable since it can directly obtain the acceleration data. As shown in Algorithm 2, we use accelerometer data and sliding window to detect the parking state. To detect the parking state correctly for static initialization, we set a sliding window with a time of $2T$ and divide it into two subwindows $w1$, $w2$, $w1 \in (T, 2T)$, $w2 \in (0, T)$. Then, we can detect the parking state or starting motion by comparing the standard deviation of the accelerometer measurements a from two subwindows a_{w1} and a_{w2} with the threshold std_{th}, to be specific, if a_{w1} is bigger than std_{th} while a_{w2} is less than std_{th}, we consider that the vehicle starts to move at $w1$ but keeps stationary at $w2$, then we can initialize the system using the data of $w2$. Otherwise, if a_{w1} is less than std_{th}, it means that the vehicle is still stationary and we slide the window until the start state is detected. Conversely, only both a_{w1} and a_{w2} are less than the threshold std_{th} can we consider that the vehicle has stopped.

Similarly, we determine the value of std_{th} by collecting the data of acceleration and its standard deviation from two subwindows, as shown in Fig. 7(a). It can be seen that the acceleration varies obviously from the 2nd second, which means that the vehicle starts from parking. a_{w1} also varies a lot at the same time, while a_{w2} is still near to zero. According to our algorithm, it can be known that the starting motion is detected successfully. More specifically, the variation of a_{w1}

Algorithm 2 Parking State Detection

Input: $w1, w2, a, std_{th}, T$
Output: The current motion state $\mathbf{S}$
  while $w1$ and $w2$ are both full **do**
    Calculate the a_{w1} and a_{w2} in $w1$ and $w2$
    if $a_{w2} < std_{th}$ **then**
      if $a_{w1} > std_{th}$ **then**
        $\mathbf{S} \leftarrow \mathbf{S}_{park}, t \in w1$, $\mathbf{S} \leftarrow \mathbf{S}_{drive}, t \in w2$
      else
        $\mathbf{S} \leftarrow \mathbf{S}_{park}$
      end if
    else
      $\mathbf{S} \leftarrow \mathbf{S}_{drive}$
    end if
    Slide the window $w1$ and $w2$
  end while

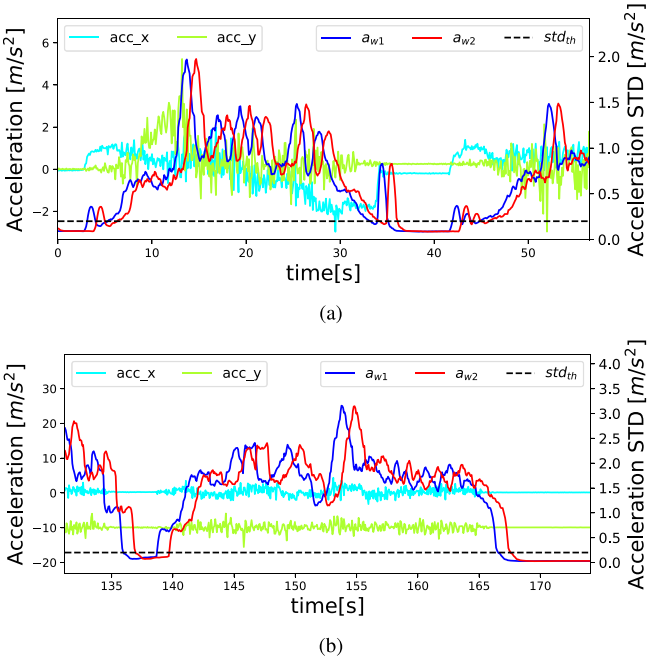


Fig. 7. Variation of acceleration data and its standard deviation of the sliding window in (a) Kaist dataset sequence urban 26 and (b) real-world ground robot.

has the maximum value of 0.35 during that time. Based on the above, we set std_{th} as 0.2 to ensure the sensitivity of the detection. In addition, the vehicle parks at about the 36th second according to the variation of the acceleration, while both the a_{w1} and a_{w2} are lower than the threshold std_{th}. Also, we evaluate the selected threshold in ground robot data to illustrate the universality as Fig. 7(b). It indicates that our method is effective in detecting the parking state in both low-speed and high-speed scenarios.

IV. Experiments and Discussions

In this section, we perform the experiments in different scenarios to evaluate our method. First, we perform the simulation experiments to evaluate the performance of the proposed online extrinsic calibration method, since the required motion environment and EPs can be provided by our own which is benefit to our experiment. Then, we evaluate the performance of the initialization and positioning with the autonomous driving public dataset. As described above, we only fuse the forward velocity of the wheel encoder in the experiments. Our implementation is based on VINS-Fusion [34], [35] and all the experiments are carried out on a laptop computer with an Intel i9-12900H CPU and 16G RAM.

TABLE I
EPs Sets for Simulation

Sets	Pitch[°]	Yaw[°]	x[m]	y[m]
EP.I	-1.0	0.0	0.15	-0.05
EP.II	0.0	0.0	0.3	-0.2
EP.III	-1.0	1.0	0.2	-0.1

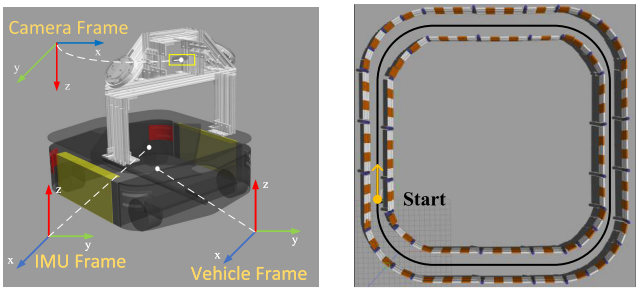


Fig. 8. Simulation vehicle and experiment route.

A. Simulation Experiments

We construct the simulation environment and simulated vehicle model by Gazebo simulation referring to the work of BetaGo [36]. The simulated vehicle and experiment route are shown in Fig. 8. The vehicle is equipped with a D435i camera model (global shutter and 30 Hz), a consumer-level IMU (100 Hz), and a differential chassis that can feedback the wheel speed at 100 Hz. The center of the chassis is taken as the origin of baselink. The experiment route has four straight lines and four 90° corners. The vehicle moves forward with a linear velocity of 1.5 m/s and an angular velocity of 0.5 rad/s. To obtain a good convergence result, a lot of laps are made in the experiment route. We set three different sets of EPs for simulation experiments as Table I. Since the roll and z-axis translation are not observable, we only set the pitch and yaw in orientation, x and y in translation. In the simulation experiment, we set the NHC noise weight coefficient $k_{\text{nhc}} = 0.1$.

We perform the experiments to evaluate the performance of extrinsic calibration in both initialization and back-end optimization, in which the necessity of initial estimation of EPs and the calibration accuracy are discussed, respectively.

1) Extrinsic Calibration in Initialization: As mentioned above, the initial estimation of EPs influences not only the stability of the fusion method but also the performance of the extrinsic calibration, especially for the yaw component of the orientation. In this section, we perform experiments to evaluate the

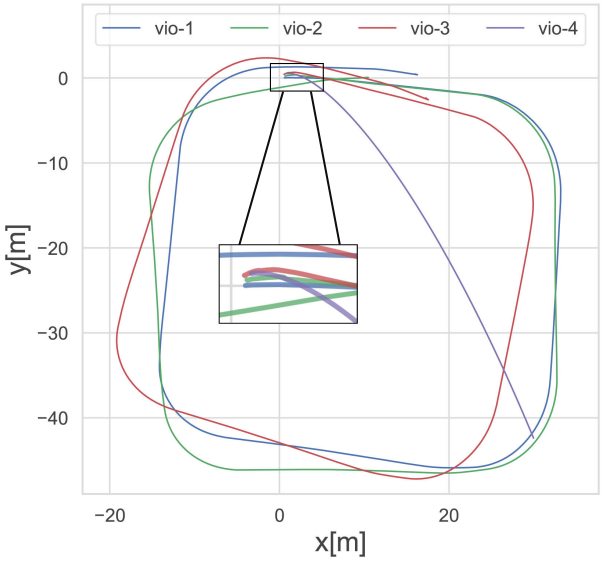


Fig. 9. Trajectories of four schemes in a simulation experiment. The legend vio-1–vio-4 corresponds to schemes 1)–4).

influence on the trajectory with different initial values of the yaw. The EPs set EP.II is selected to complete the experiment with both the pitch and yaw are 0°. First of all, we set four schemes for comparison: 1) initialize the EPs; 2) fix the initial yaw as 30°; 3) fix the initial yaw as 60°; and 4) fix the initial yaw as 90°. Since schemes 2) to 4) have set a different value of yaw compared with the given groundtruth, we use the method from [9] to complete the initialization. After the system initialization, we enable online extrinsic calibration in the back-end optimization to correct the error of the initial value of yaw. Then, we compare the trajectories produced by four schemes which are shown in Fig. 9. It can be seen that there is a big difference at the start point of the trajectory. With the initial value of yaw increasing, the trajectory deviation near the start point enlarges from vio-2 to vio-3, even if it causes divergence in trajectory vio-4. However, the trajectory produced by our proposed initialization method (vio-1) is smoother than others at the start part due to the effective estimation of yaw in initialization. In conclusion, a good initial value for the orientation of EPs is necessary and the error value can badly influence the trajectory and even cause divergence.

We also collect the numerical initialization results of the above four schemes in 10 runs to illustrate our method quantitatively, as shown in Fig. 10. It can be seen that the yaw estimation in scheme 1) can be around 0° with an error range of about 2°, while the results in scheme 4) show a similar phenomenon. In addition, the yaw estimation in schemes 2) and 3) shows a little bit more error range of about 4°, but can still be around the groundtruth. In fact, our experiments show that the yaw initialization error range within 15° is acceptable. Overall, the above results demonstrate that our initialization method is effective and robust for extrinsic calibration with different kinds of parameters.

2) Extrinsic Calibration in Nonlinear Optimization: We estimate the EPs online in nonlinear optimization with the proposed observability-aware method. The results are shown in

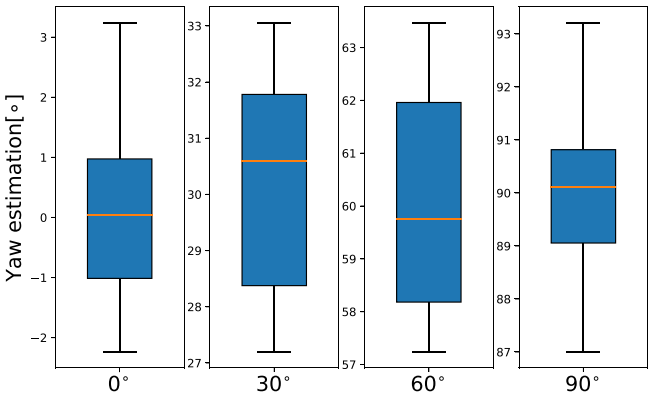


Fig. 10. Initial yaw estimation of four schemes in 10 runs.

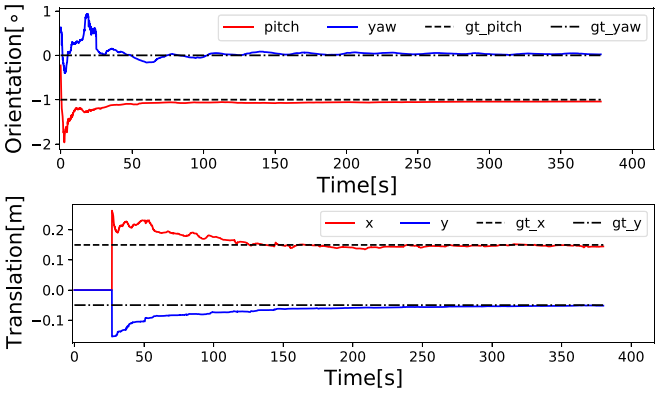


Fig. 11. Calibration results for EP.I.

Figs. 11–13. In all three figures, the orientation parameters are estimated from the start of the vehicle, whereas the translation parameters are fixed as zero at the beginning and start to estimate when the turning condition is satisfied. In the mean time, it can be seen that all the estimated parameters can converge to the groundtruth. Furthermore, all three EP sets have an almost translation calibration error at about 0.02 m. For the rotation calibration, both EP.I and EP.III have a small calibration error at about 0.05° whereas a larger error of 0.1° is shown at EP.II. This is because of both the pitch and yaw of EP.II are zero and the motivation for the extrinsic rotation calibration is inadequate. Overall, the experiment results show that our calibration methods with observability judgment can achieve accurate and consistent estimation in different kinds of EPs. It also proves that the proposed method is effective and robust.

B. Public Dataset Experiments

The Kaist dataset is utilized to complete the experiments in this section. The dataset captures features in metropolis areas, complex buildings, residential areas, and other urban environments. The sensor configuration includes a stereo camera (global shutter and 10 Hz), a consumer-level IMU (100 Hz), wheel encoders (4096 pulses per round and 100 Hz), and other various position sensors. The groundtruth is generated based on the SLAM algorithm which fuses the wheel encoder,

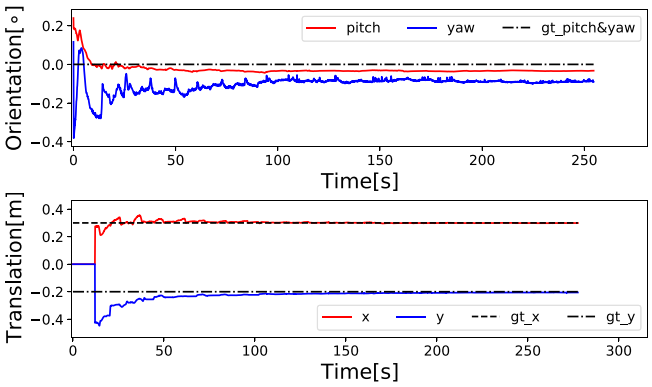


Fig. 12. Calibration results for EP.II.

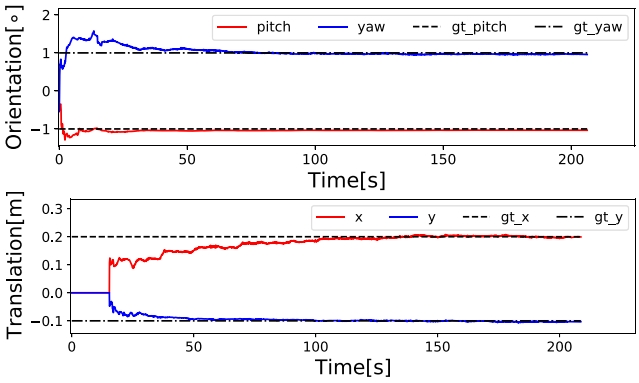


Fig. 13. Calibration results for EP.III.

FOG, VRS-GPS, and lidar measurements. Furthermore, it is noted that a lot of parking and starting places are included in the dataset which can evaluate the proposed method strongly. In the public dataset experiment, we set the NHC noise weight coefficient $k_{\text{nhc}} = 1.0$.

1) Initial Visual Scale Recovery: To evaluate the performance of initialization, we compare our method with the initialization method from VINS and an open-source visual-inertial-wheel encoder fusion method VIW-Fusion [20]. Considering the requirement of the static initialization of our method, we select six places with the parking state in sequences urban26, urban28, and urban30 of the Kaist dataset as Fig. 14, to complete the experiment and perform the other two methods at the same start point. We set the size of the sliding window as 15 and kept other common parameters from VINS identical. Since the visual scales estimated at the different start points are not at the same exact, to make a reasonable evaluation, we compare the recovered trajectory with groundtruth. We define that the translation from visual poses with recovered scale is t_{rec}, and the corresponding groundtruth is t_{gt}, then the scale recovery error can be defined as

$$e_{\text{scale}} = \frac{\left|t_{\text{rec}} - t_{\text{gt}}\right|}{t_{\text{gt}}} \times 100\%. \tag{33}$$

The comparison results are shown in Table II. It can be seen that both of the three methods work well in urban26-1 and urban28-3 with an error within 3% since there is a good visual condition in both above two places for estimating visual poses

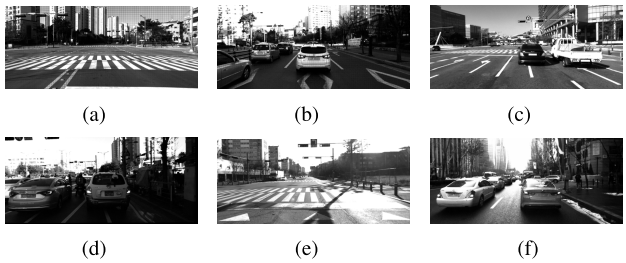


Fig. 14. Places with parking state in Kaist dataset. (a) Urban26-1. (b) Urban26-2. (c) Urban28-1. (d) Urban28-2. (e) Urban28-3. (f) Urban30.

TABLE II
INITIAL VISUAL SCALE RECOVERY COMPARISON

start point	Proposed	VINS	VIW-Fusion
urban26-1	2.15%	2.21%	**0.35%**
urban26-2	**1.66%**	78.17%	11.24%
urban28-1	**2.73%**	×	24.11%
urban28-2	**0.44%**	×	1.84%
urban28-3	2.69%	3.08%	**2.42%**
urban30	**2.53%**	35.79%	2.81%

as is shown in Fig. 14. However, our method performs better in the other four places, where more vehicles appear in front of the camera which leads to worse visual measurements when starting. Especially in urban28-1, our method only has an error of 2.73% while VIW-Fusion has a larger one of 24.11%, even the initialization failed for VINS. Overall, the initialization error remains within 3% in all of the experiment points for our method, which indicates that the proposed initialization method is more robust and effective.

2) Positioning Accuracy Comparison: Finally, we evaluate the trajectory accuracy of our fusion method quantitatively in autonomous driving scenarios with part of sequence urban26 of the Kaist dataset, which is about 3.6 km and covers most scenarios in urban areas. To evaluate the trajectory accuracy under the online extrinsic calibration, we set three modes of our method which correspond to fixing EPs, calibrating EPs directly, and calibrating EPs with observability analysis, respectively. In addition, our method is compared with the visual-inertial fusion methods VINS-Fusion and VIW-Fusion, both of which provide the function of online calibrating EPs. Considering the assumption that the vehicle usually drives on a flat pavement, we only evaluate the horizontal position error and use absolute pose error (APE) to evaluate the global consistency of trajectories.

The trajectories are shown in Fig. 15 and the root mean square error of each algorithm are VINS-Fusion (mono) 79.764 m, VIW-Fusion with calibration 45.684 m, VIW-Fusion without calibration 37.239 m, proposed with calibration 1) 16.225 m, proposed with calibration 2) 12.637 m and proposed without calibration 10.637 m. It can be seen obviously that our method either with online extrinsic calibration or

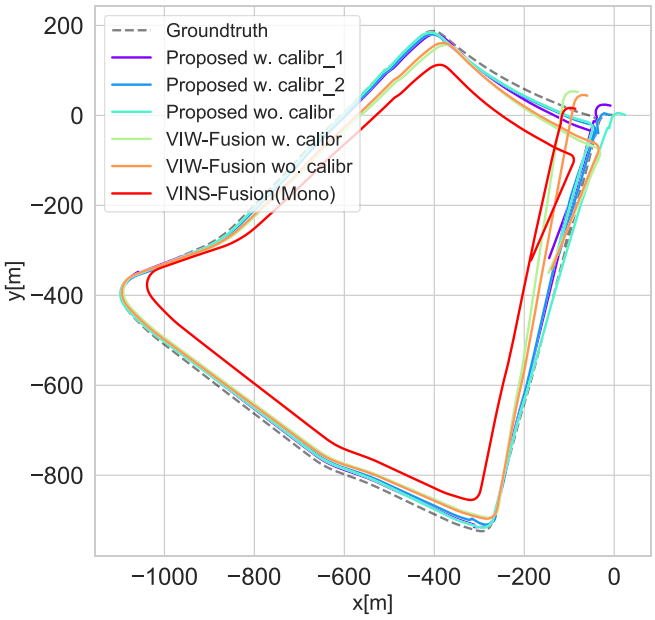


Fig. 15. Trajectories of part of the sequence urban26: VINS-Fusion (mono) represents the mono mode of VINS-Fusion, VIW-Fusion w. calibr and VIW-Fusion wo. calibr represent the VIW-Fusion with/without online calibration, respectively. Proposed w.calibr 1 represents calibrating EPs directly, proposed w.calibr 2 represents calibrating EPs with observability judgment, and proposed wo. calibr represents fixing EPs.

fixing the EPs performs better than VINS-Fusion and VIW-Fusion. In addition, fixing EPs is the best of our method since accurate EPs can reduce the most errors of extrinsic calibration. For the online calibration, our method with observability analysis performs better than the other due to the reasonable calibration environment. In conclusion, our method performs well with respect to both the position accuracy in autonomous driving scenarios, especially under the circumstance of no prior knowledge of the IMU-odometer EPs, our method can online calibrate the EPs and reach a considerable accuracy at the same time.

C. Real-World Experiments

We use our ground robot to carry out real-world experiments. Hardware and frames are shown in Fig. 16. The ground robot is equipped with a differential chassis and a D455 stereo camera (30 Hz) with built-in IMU (100 Hz). Wheel speed can be obtained from the chassis. The EPs of the camera-IMU ($\mathbf{R}_c^b$, $\mathbf{p}_c^b$) have been calibrated with *Kalibr* tool beforehand and we calibrate the IMU-odometer EPs online by the proposed method. Here, we set the same NHC noise weight coefficient k_{nhc} as the public dataset experiment.

To demonstrate the consistency of our method, we use a 90-m long calibration experiment route to calibrate the IMU-odometer EPs first, as shown in Fig. 17. Then, we conduct the position accuracy experiments with and without the calibrated EPs in another 408-m long route as shown in Fig. 18. Specifically, we set an identity matrix for extrinsic rotation and set a zero vector for extrinsic translation as the initial parameters. Finally, we compare the position accuracy generated from the above two sets of EPs to evaluate

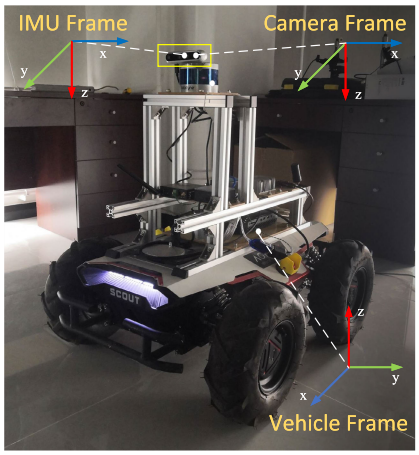


Fig. 16. Real word experiment vehicle.

Fig. 17. Extrinsic calibration experiment route.

Fig. 18. Positioning experiment route (408 m).

our method. In the position accuracy experiment, the trajectory from LIO-SAM [37] with loop closure is used as the groundtruth. Furthermore, we use absolute trajectory error (ATE) to evaluate the position accuracy.

The position results are shown in Fig. 19. It can be seen clearly that the trajectory with online extrinsic calibration is closer to the groundtruth than the trajectory with EPs fixed. The former has a position error of 1.86 m (RMSE) in our real-world experiment resulting in 0.46% error in the positioning, whereas the latter has a position error of 1.19 m (RMSE) resulting in 0.29% error. Furthermore, Figs. 20 and 21 show the convergence results of the extrinsic calibration. Since the IMU and the vehicle have a different coordinate that results in a difference of 90° regarding the orientation, we transform it into the standard coordinate to collect the results for clarity. To demonstrate the convergence performance, we also evaluate the standard deviation of the calibration results in a 10-s long slide window. The results show that the pitch converges at about 0.1° with about a standard deviation of 0.05°, while the yaw converges at about −0.1° with a standard deviation of about 0.005°. The x-axis translation converges at about 0.12 m while that of the y-axis converges at −0.05 m. Both have the same standard deviation of about 0.002 m. It can be observed that the yaw component has a better convergence than pitch.

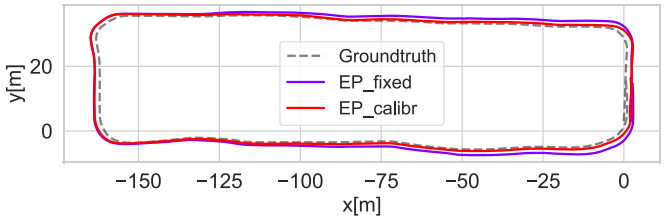


Fig. 19. Trajectory of the real world experiment. EP caliber represents the trajectory with calibrated EPs and EP fixed represents the trajectory with initial EPs.

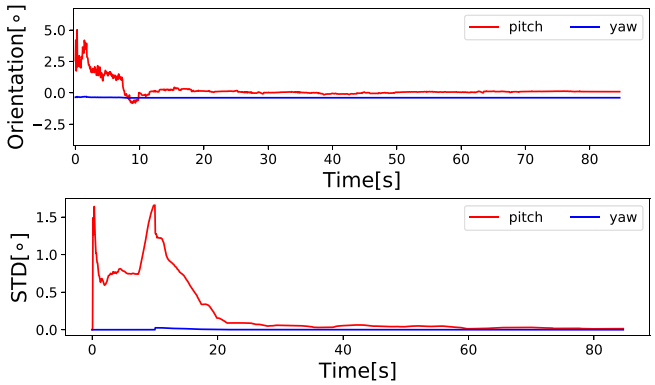


Fig. 20. Extrinsic orientation calibration with STD in a real-world experiment.

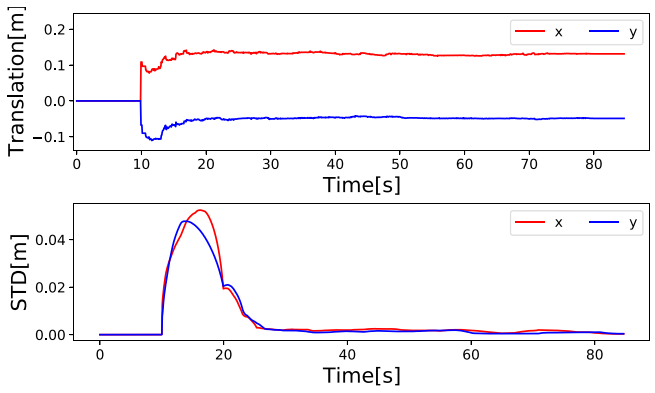


Fig. 21. Extrinsic translation calibration with STD in a real-world experiment.

This is because the motivation in the pitch component is much less, leading to a worse observability than yaw's. Overall, all the parameters manage to converge at a very low error range. In conclusion, the comparison results strongly prove that the proposed method is effective and efficient for online IMU-odometer extrinsic calibration.

V. Conclusion

In this article, we propose a visual-inertial-odometer framework for ground vehicles which is implemented based on the vehicle motion constraint. The proposed framework allows to online calibration of the IMU-odometer EPs explicitly. We also developed an observability-aware method to enhance the system stability and the performance of online extrinsic calibration. Besides, a robust initialization algorithm is developed to obtain good values of the system and EPs. The key of the proposed method is that it fuses the raw odometer velocity measurement *directly* rather than preintegrating them into the displacement. This is beneficial not only to the accuracy but the online extrinsic calibration. A variety of experiments from simulation, public datasets, and the real world show that our method is effective, robust, and can achieve the same level as the other state-of-the-art methods. The limitation of our approach lies in the fact that the heading drift from the gyroscope still cannot be ignored in the long-term positioning. We will research this aspect by combining the heading measurements from both gyroscope and multiple wheel encoders in the future.

References

[1] 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.*, vol. 2, Apr. 2007, pp. 3565–3572.

[2] K. Sun et al., "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

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

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

[5] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[6] 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.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. 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.

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

[9] T. Qin and S. Shen, "Robust initialization of monocular visual–inertial estimation on aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4225–4232.

[10] C. Campos, J. M. M. Montiel, and J. D. Tardós, "Inertial-only optimization for visual-inertial initialization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 51–57.

[11] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4304–4311.

[12] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1280–1286.

[13] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2088–2095.

[14] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 473–480.

[15] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1360–1367.

[16] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, Jan. 2011.

[17] W. Huang and H. Liu, "Online initialization and automatic camera-IMU extrinsic calibration for monocular visual-inertial SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5182–5189.

[18] F. Ma, J. Shi, Y. Yang, J. Li, and K. Dai, "ACK-MSCKF: Tightly-coupled Ackermann multi-state constraint Kalman filter for autonomous vehicle localization," *Sensors*, vol. 19, no. 21, p. 4816, Nov. 2019.

[19] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "VINS on wheels," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5155–5162.
[20] T. Zhuang. (2021). *VIW-Fusion: Visual-Inertial-Wheel Fusion Odometry*. [Online]. Available: https://github.com/TouchDeeper/VIW-Fusion
[21] 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.
[22] J. Liu, W. Gao, and Z. Hu, "Bidirectional trajectory computation for odometer-aided visual-inertial SLAM," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1670–1677, Apr. 2021.
[23] J. Liu, W. Gao, and Z. Hu, "Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5391–5397.
[24] M. Quan, S. Piao, M. Tan, and S.-S. Huang, "Tightly-coupled monocular visual-odometric SLAM using wheels and a MEMS gyroscope," *IEEE Access*, vol. 7, pp. 97374–97389, 2019.
[25] Y. He, Y. Guo, A. Ye, and K. Yuan, "Camera-odometer calibration and fusion using graph based optimization," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 1624–1629.
[26] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 642–657, May 2019.
[27] J. Xinchun, W. Dongyan, Y. Hong, and Z. Deyun, "Vehicle multi-source fusion navigation method based on smartphone platform," *J. Chin. Inertial Technol.*, vol. 28, no. 5, pp. 638–644, 2020.
[28] Q. Zhang, Y. Hu, and X. Niu, "Required lever arm accuracy of nonholonomic constraint for land vehicle navigation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8305–8316, Aug. 2020.
[29] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[30] 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.
[31] S. Agarwal and K. Mierle. (Mar. 2022). *Ceres Solver*. TCS Team. [Online]. Available: https://github.com/ceres-solver/ceres-solver
[32] H. Zhao, X. Ji, D. Wei, and J. Zhang, "Online IMU-odometer extrinsic calibration based on visual-inertial-odometer fusion for ground vehicles," in *Proc. IEEE 12th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2022, pp. 1–8.
[33] W. Liu, Q. Nong, X. Tao, F. Zhu, and J. Hu, "OD/SINS adaptive integrated navigation method with non-holonomic constraints," *Acta Geodaetica Cartographica Sinica*, vol. 51, no. 1, pp. 9–17, 2022.
[34] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019, *arXiv:1901.03638*.
[35] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," 2019, *arXiv:1901.03642*.
[36] T. Zhuang. (2021). *Betago: A Dual Arm Mobile Robot Simulation Platform with Multiple-Sensor Based on ROS*. [Online]. Available: https://github.com/TouchDeeper/BetaGo
[37] 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.

Hang Zhao received the B.S. degree in automation engineering from the University of Science and Technology Beijing, Beijing, China, in 2020, and the M.S. degree in signal and information processing from the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China, in 2023. He is currently pursuing the Ph.D. degree in signal and information processing with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, under the guidance of Prof. Dongyan Wei.

Xinchun Ji received the B.S. and M.S. degrees in guidance navigation and control (GNC) from the Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China, in 2010 and 2013, respectively. He is pursuing the Ph.D. degree in electronic information with Northwestern Polytechnical University (NWPU), Xi'an, China.

He is currently an Engineer at the Aero Information Research (AIR) Institute, Chinese Academy of Sciences (CAS), Beijing. His research interests include multisensor fusion and geomagnetic matching.

Dongyan Wei (Member, IEEE) received the B.S. degree in communication engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2006, and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011.

He is currently a Research Fellow with the Aerospace Information Research (AIR) Institute, Chinese Academy of Sciences (CAS), Beijing. He is the author of one book, more than 30 articles, and more than 20 inventions. His research interests include indoor position, multisensor fusion, and positioning in wireless networks.

Dr. Wei is the Co-Chair of the International Conference on Indoor Positioning and Indoor Navigation (IPIN) in 2023.