2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 1-5, 2023. Detroit, USA

VIW-Fusion: Extrinsic Calibration and Pose Estimation for Visual-IMU-Wheel Encoder System

Chunxiao Qiao, Shuying Zhao, Yunzhou Zhang, Yahui Wang, Dan Zhang

Abstract— **The data fusion of camera, IMU, and wheel encoder measurements has proved its effectiveness in localizing ground robots, and obtaining accurate sensor extrinsic parameters is its premise. We propose an extrinsic parameter calibration algorithm and a multi-sensor-based pose estimation algorithm for the camera-IMU-wheel encoder system. First, we propose a joint calibration algorithm for the extrinsic parameters of the camera-IMU-wheel encoder system, which improves the accuracy and robustness of the camera-wheel encoder calibration. We then extend the visual-inertial odometry (VIO) to incorporate the measurements from the wheel encoder and weight the wheel encoder measurements according to angular velocity in global optimization to improve the performance. We further propose a novel method for VIO initialization by integrating wheel encoder information, which significantly reduces the scale error in initialization. We conduct extrinsic parameter calibration experiments on a real self-driving car and validate the performance of our multi-sensor-based localization system on the KAIST dataset and a dataset collected by our self-driving vehicles by performing an exhaust comparison with the state-of-the-art algorithms. Our implementations are open source[1].**

I. INTRODUCTION

Various universities and technology companies are joining in the research and development of autonomous vehicles. The pose estimation of ground vehicles has become a current research hotspot. At present, the mainstream solutions for localization rely heavily on the global positioning and inertial navigation systems (GPS-INS) together with Lidar [1]–[3]. These localization systems usually cost thousands or even tens of thousands of dollars. Thus the use of low-cost equipment such as cameras, wheel encoders, and IMUs in ground-vehicle localization has received more and more attention. Such as, Qin et al. [4] proposed to localize the vehicle with wheel encoders, RTK and pre-built maps. J Liu et al. [5] proposed a SLAM system using the camera, IMU and wheel encoders. Among the solutions, visual odometry (VO) and visual-inertial-odometry have thrived in the unmanned aerial vehicle (UAV) and handheld devices, such as ORB-SLAM [6], ORB-SLAM3 [7], and VINS-Fusion [8]. With the help of IMU, VIO is able to render scale available. [9] explains that moving along straight lines or circular arcs with constant acceleration or speed can raise extra unobservable directions to the vision-aided inertial navigation system (VINS) model. While ground vehicles such as cars often undergo approximately constant acceleration or speed motion, resulting in the absence of scale observation. Since wheel encoder is common in ground vehicles, and including wheel encoder information could compensate for the observability degeneration problem, we aim to design a low-cost, efficient, and robust vision-IMU-wheel encoder localization algorithm in the work of this paper.

As we all know, the premise of multi-sensor fusion is to have accurate sensor extrinsic parameters. While the public datasets [10], [11] often provide accurate extrinsic parameters of the sensors, deploying localization algorithms in real vehicles needs to calibrate the sensors to acquire extrinsic parameters. Traditional methods for wheel encoder and camera extrinsic parameter calibration could perform offline [12]–[14] or online [15], [16], often with the help of specific devices such as calibration boards or AR-markers. Most of the calibration programs have been based on the assumption of the planar movement of vehicles, which could barely be satisfied in reality. As vehicles undergo planar movement, the height of the camera cannot be calibrated. Since we have included an IMU in our work, we aim to design a camera-wheel encoder extrinsic parameter calibration scheme assisted by an IMU. With the schema, we rely no longer on the planar assumption and additional tools such as calibration boards. Moreover, the camera height could also be computed. The main contributions of this work include:

- We propose a joint calibration algorithm for the extrinsic parameters of the camera-IMU-wheel encoder system and validate the calibration method's accuracy and robustness in self-driving cars.
- We propose a two-stage initialization method for VIO systems fused with wheel encoders. The problem of unobservable scale caused by constant acceleration or speed of the ground mobile robot during the initialization of the VIO algorithm is solved.
- We propose a tightly-coupled data fusion framework of VINS and wheel encoder for localization, which can adjust the optimized weights of the wheel encoder by angular velocity. And conduct extensive experiments on a self-driving car and datasets to verify the accuracy of the proposed algorithm.

The system has been successfully deployed on our self-

*This work was supported by National Natural Science Foundation of China (No. 61973066) and Major Science and Technology Projects of Liaoning Province(No. 2021JH1/10400049)

Chunxiao Qiao is with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China, and also with Da Jiang Innovations, Shenzhen 518041, China 1900862@stu.neu.edu.cn

Shuying Zhao and Yunzhou Zhang are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China zhaoshuying@ise.neu.edu.cn, hangyunzhou@mail.neu.edu.cn

Yahui Wang and Dan Zhang are with UISEE Technology, Beijing 102402, China dan.zhang@uisee.com

[1]https://github.com/chunxiaoqiao/VIW-Fusion.git

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-9190-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IROS55552.2023.10341453

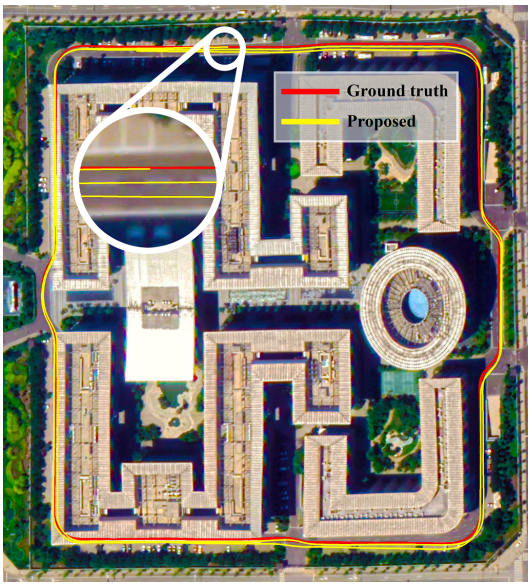


Fig. 1. Ground truth and estimated trajectory of the proposed method plotted on the Google map. The red line is the ground truth obtained by RTK, and the yellow line is the trajectory obtained by our approaches.

driving cars in road scenarios, as shown in Fig. 1.

II. Related Work

A. Camera-Wheel Encoder Extrinsic Parameter Calibration

The extrinsic parameter calibration of the camera-wheel encoder focuses on calibrating the rotation and translation between the wheel encoder and the camera coordinate system. [17] proposed solving the extrinsic parameter calibration problem using a planar marker. But the additional requirement has constrained the method to the factory and laboratory environment. [12], [13] obtained the incremental motion of the sensors by a visual odometry and wheel odometer. They then calibrated the extrinsic parameters of the camera-wheel encoder by using the hand-eye calibration method. Tang et al. [18] proposed a non-iterative initialization algorithm to calibrate extrinsic parameters of the camera-odometry system. The system is based on road sign measurements and roughly calibrated odometry, which makes the algorithm less adaptable because identifiable road signs need to be placed and measured in advance. To compensate for possible extrinsic parameter changes and improve estimation, Lee et al. [16] proposed that visual-inertial-wheel odometry(VIWO) performs online sensor calibration of the spatiotemporal extrinsic of odometer-IMU/camera as well as the wheel encoder's intrinsics. But these algorithms have assumed the planar motion of the vehicles. The translation is thus two-dimensional, and the rotation is all about the z-axis, and the estimation of camera height becomes impossible. Liu et al. [15] designed a computationally efficient online extrinsic parameters calibration method that is triggered as soon as the bias of the accelerometer converges, where they estimate the orientation of the vehicle by integrating IMU measurements. And [19] is based on Vins-mono [20] to tightly couple the wheel encoder and GPS and open source the code. In our work, we extend [13] to estimate camera-wheel encoder extrinsic parameters with a coupled IMU to significantly improve the accuracy and additionally enable the estimate of camera height.

B. SLAM Using Camera

In recent years, vision-based localization methods for mobile robots have developed rapidly. One of the most representative VO and VIO frameworks is ORB-SLAM3 [7]. The framework supports multiple kinds of cameras, such as monocular, stereo, RGB-D and fisheye cameras. They also proposed the VIO system with a fast and accurate IMU initialization method and a comprehensive multi-map management system. VINS-Mono [20], proposed by Qin et al., was a very accurate and robust VIO system. In VINS-Fusion [8], they extended their system to support binocular and GPS data fusion. For SLAM of ground vehicles, Wu et al. [9] proved that the VINS system has additional observability degradation in constant acceleration or speed movement, such as the scale being unobservable. They also demonstrated that scale degradation could be eliminated by using wheel encoders. Lee et al. [16] developed an efficient and consistent MSCKF-based VIWO system that incorporated information on wheel encoders, IMUs, and cameras. Additionally, the system supported online calibration of the extrinsic parameters. Liu et al. [15] proposed a new strategy for fusing camera, IMU and wheel encoder data for car localization. In the pre-integration stage, IMU measurements are combined with wheel encoder readings. With the help of encoder readings, a more robust, computationally efficient online extrinsic calibration method is designed, which is used immediately when the accelerometer bias reaches convergence. Liu et al. [5] proposed a wheel encoder-based VI-SLAM bidirectional trajectory computation method. They used a thread to optimize the trajectory before the first turn and solved the problem of the unobservability of accelerometer bias and extrinsic parameters before the first turn. Zhang et al. [21] proposed a motion-manifold-based method for pose estimation of ground robots, which performs motion manifold-based 6-D integration with wheel encoders and fusion with a monocular camera.

III. METHODOLOGY

A. Camera-IMU-Wheel Encoder Calibration

To fuse measurements from multi-sensor, we need extrinsic parameters among the sensors. Although public datasets such as KITTI [11] and KAIST [10] provide calibrated extrinsic parameters of the sensor, to deploy the algorithm on a real car, we need to calibrate by ourselves. Most of the existing algorithms [12], [13] for camera-wheel encoder calibration usually assumed vehicles move in the two-dimensional plane. This assumption is often not strictly established in the actual calibration process. Besides, when the planar movement is assumed, the extrinsic translation parameters of the z-axis cannot be calibrated. To set aside the plane assumption, we use the measurements from the IMU to provide the wheel encoder with orientation and perform a

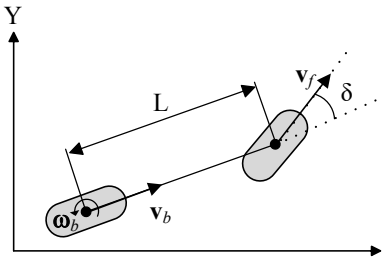


Fig. 2. In the bicycle model of the car, where $\boldsymbol{v}_f$ is the wheel speed of the front wheel, $\boldsymbol{v}_b$ and $\boldsymbol{\omega}_b$ are the wheel speed and angular velocity of the rear wheel, δ is the steering angle of the front wheel, and L is the wheelbase of the front and rear axles of the vehicle.

6-DoF pose integration. We first acquire vehicle orientation by integrating IMU measurements and then use spherical interpolation to interpolate the orientation parameterized by quaternion to align with the wheel encoder timestamps. The calibration also needs to estimate the poses of the monocular camera, and we use the VO of ORB-SLAM3 [7] to recover camera poses and the loop-closure algorithm to guarantee a consistent global scale.

1) Vehicle Kinematics Model: We use a bicycle model [22] to simplify the motion of the vehicle as shown in Fig. 2. For both calibration and dead reckoning, the centre of the rear axle is set as the origin of the vehicle coordinate frame. The x-axis and y-axis of the vehicle coordinate frame point to the right and the front of the vehicle, respectively. The angular velocity and linear velocity of the rear wheels can be derived by the following equation:

$$\begin{aligned} \boldsymbol{v}_b &= \boldsymbol{v}_f \cos(\delta) \\ \boldsymbol{\omega}_b &= \boldsymbol{v}_f \frac{\tan(\delta)}{L} \end{aligned} \tag{1}$$

2) IMU-wheel encoder extrinsic parameter calibration: Before integrating IMU information, we need to calibrate the rotation transformation between the gyroscope and wheel encoder. We first follow CamOdoCal [13] to calibrate the initial value of the 5-DoF extrinsic parameters, namely the rotation matrix $\boldsymbol{R}_c^e$, the translation in x and y directions $\boldsymbol{t}_c^e$ of the extrinsic parameters of the camera and wheel encoder, with the trajectories estimated by pure dead-reckoning and VO of ORB-SLAM3 as input. But at the current stage, the z-axis element of $\boldsymbol{t}_c^e$ cannot be computed. Then we further refine the extrinsic parameters. Since the camera and IMU sensors are mounted in different positions in our self-driving cars, it is not suitable for use with existing calibration toolboxes like Kalibr [23], which is designed for handheld devices like MYNTEYED camera or ZED camera with integrated IMUs. We propose a combination of hand-eye calibration and joint optimization [24] to calibrate the extrinsic parameters $\boldsymbol{R}_c^I$ and $\boldsymbol{t}_c^I$ of the camera and IMU sensors in our cars. Finally, through the two rotating extrinsic parameters, $\boldsymbol{R}_c^e$ and $\boldsymbol{R}_c^I$, the IMU-wheel encoder rotating extrinsic parameter $\boldsymbol{R}_I^e$ can be deduced:

$$\boldsymbol{R}_I^e = \boldsymbol{R}_c^e (\boldsymbol{R}_c^I)^{-1} \tag{2}$$

After obtaining the rotation matrix of the IMU-wheel encoder extrinsic parameter $\boldsymbol{R}_I^e$, it is possible to integrate the wheel encoder measurement coupled with IMU gyroscope information in the 3D space and obtain the 6-DoF poses of the vehicle.

3) Wheel Encoder Odometer Coupled with Gyroscope: In order to calibrate the extrinsic parameters of the camera and wheel encoder by the hand-eye calibration method, the poses estimated by dead-reckoning with the wheel encoder and VO are required [25]. We integrate the linear velocity from the wheel encoder and angular velocity from the gyroscope for dead-reckoning, as shown in equation (3) and equation (4):

$$\boldsymbol{u} = \int_t^{t+1} \boldsymbol{v}_b(t) dt \tag{3}$$

$$\boldsymbol{R}_{e_i}^{e_{i+1}} = \boldsymbol{R}_I^e \boldsymbol{R}_{I_t}^{I_{t+1}} (\boldsymbol{R}_I^e)^{-1} \tag{4}$$

where $\boldsymbol{R}_{e_i}^{e_{i+1}}$ is the rotation attitude of the vehicle, $\boldsymbol{R}_I^e$ is the rotation matrix of the IMU-wheel encoder extrinsic parameters, $\boldsymbol{u}$ is the displacement increment of the vehicle coordinate frame, $\boldsymbol{R}_{I_t}^{I_{t+1}}$ is the rotation increment of the IMU.

4) Camera-Wheel Encoder Calibration: First, we start with the well-known hand-eye calibration problem [12]:

$$\boldsymbol{q}_{e_i}^{e_{i+1}} \otimes \boldsymbol{q}_c^e = \boldsymbol{q}_c^e \otimes \boldsymbol{q}_{c_i}^{c_{i+1}} \tag{5}$$

$$\left(\boldsymbol{R}\left(\boldsymbol{q}_{e_i}^{e_{i+1}}\right) - \boldsymbol{I}\right) \boldsymbol{t}_c^e = s\boldsymbol{R}\left(\boldsymbol{q}_c^e\right) \boldsymbol{t}_{c_i}^{c_{i+1}} - \boldsymbol{t}_{e_i}^{e_{i+1}} \tag{6}$$

where $\boldsymbol{q}_{e_i}^{e_{i+1}}$is the unit quaternion that represents the orientation of the wheel encoder frame at the time $i+1$ in the frame at the time i, $\boldsymbol{q}_{c_i}^{c_{i+1}}$ is the unit quaternion of the camera frame at the time $i+1$ in the frame at the time i , $\boldsymbol{q}_c^e$ is the rotational part of the camera-wheel encoder extrinsic parameters, $\boldsymbol{t}_c^e$ is the translational part of the camera-wheel encoder extrinsic parameters, $\boldsymbol{t}_{e_i}^{e_{i+1}}$ is the translation of the wheel encoder frame at the time $i+1$ in the frame at the time i, $\boldsymbol{t}_{c_i}^{c_{i+1}}$ is the translation of the camera frame at the time $i+1$ in the frame at the time i, s is the scale, and $\otimes$ denotes quaternion multiplication. Thanks to the loop-closure function in ORB-SLAM3, we consider the scale unchanged in the calibration process. And this requires the calibration process to have a loop closure.

We use $\hat{\boldsymbol{q}}_{e_i}^{e_{i+1}}$ and $\hat{\boldsymbol{t}}_{e_i}^{e_{i+1}}$ to represent the rotational and translational increment of the wheel encoder from the time i to time $i+1$ by integrating the measurements of the wheel encoder and gyroscope during the period. $\hat{\boldsymbol{q}}_{c_i}^{c_{i+1}}$ and $\hat{\boldsymbol{t}}_{c_i}^{c_{i+1}}$ represent the rotational and translational increment of the camera frame from the time i to time $i+1$ estimated by ORB-SLAM3. We first calibrate the rotational part of the extrinsic parameters. We transform equation (5) into the following form:

$$\begin{aligned} \boldsymbol{\xi}_i &= \hat{\boldsymbol{q}}_{e_i}^{e_{i+1}} \otimes \boldsymbol{q}_c^e - \boldsymbol{q}_c^e \otimes \hat{\boldsymbol{q}}_{c_i}^{c_{i+1}} \\ &= (\mathcal{L}(\hat{\boldsymbol{q}}_{e_i}^{e_{i+1}}) - \mathcal{R}(\hat{\boldsymbol{q}}_{c_i}^{c_{i+1}})) \boldsymbol{q}_c^e \end{aligned} \tag{7}$$

where $\boldsymbol{\xi}_i$ is the error of $\boldsymbol{q}_c^e$ constructed from dead-reckoning and VO estimations from the time i to time $i+1$. And $\mathcal{L}$ and $\mathcal{R}$ are the left and right quaternion-product matrices.

Summing up the errors in all the n periods, we could build a least-squares model:

$$ {}^{*}\boldsymbol{q}_c^e = \arg\min_{\boldsymbol{q}_c^e} \sum_{i=0}^{n} \|\boldsymbol{\xi}_i\|_2^2 \tag{8} $$

Because the dead-reckoning is performed in 3D space without the planar motion assumption, equation (7) satisfies the observability requirement of hand-eye calibration to estimate the 6-DoF extrinsic parameters [26]. We thus estimate $\boldsymbol{q}_c^e$ by singular value decompositions (SVD). After calibrating the rotational part of the extrinsic parameters $\boldsymbol{q}_c^e$, we substitute it into equation (6) to obtain equation (9), which is the error of $\boldsymbol{t}_c^e$:

$$ \boldsymbol{\varepsilon}_i = (\boldsymbol{R}(\hat{\boldsymbol{q}}_{e_i}^{e_{i+1}}) - \boldsymbol{I})\boldsymbol{t}_c^e - s\boldsymbol{R}(\boldsymbol{q}_c^e)\hat{\boldsymbol{t}}_{c_i}^{c_{i+1}} + \hat{\boldsymbol{t}}_{e_i}^{e_{i+1}} \tag{9} $$

Similarly, by summing up the errors in all the n periods, we build a least-squares model:

$$ {}^{*}\boldsymbol{t}_c^e, {}^{*}s = \arg\min_{\boldsymbol{p}_c^e, s} \sum_{i=0}^{m} \|\boldsymbol{\varepsilon}_i\|_2^2 \tag{10} $$

In order to solve the least-squares model, we reformulate equation (10) in matrix form as in equation (9):

$$ \begin{bmatrix} \boldsymbol{D} & \boldsymbol{F} \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \\ -s \end{bmatrix} = -\hat{\boldsymbol{t}}_{e_i}^{e_{i+1}} \tag{11} $$

Among them

$$ \begin{aligned} \boldsymbol{D} =& \boldsymbol{R}(\hat{\boldsymbol{q}}_{e_i}^{e_{i+1}}) - \boldsymbol{I} \\ \boldsymbol{F} =& \boldsymbol{R}(\boldsymbol{q}_c^e)\hat{\boldsymbol{t}}_{c_i}^{c_{i+1}} \\ \boldsymbol{t}_c^e =& [t_x, t_y, t_z]^T \end{aligned} \tag{12} $$

For all the n periods, we construct a matrix:

$$ \boldsymbol{G} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{F}_1 \\ \cdots & \cdots \\ \boldsymbol{D}_n & \boldsymbol{F}_n \end{bmatrix} \tag{13} $$

Equation(10) expands to:

$$ \boldsymbol{G} \begin{bmatrix} t_x \\ t_y \\ t_z \\ -s \end{bmatrix} = \begin{bmatrix} -\hat{\boldsymbol{t}}_{e_i}^{e_{i+1}} \\ \cdots \\ -\hat{\boldsymbol{t}}_{e_0}^{e_1} \end{bmatrix} \tag{14} $$

Because we estimate VO with loop closure, a global scale consistency is guaranteed in Sim(3). We thus no longer estimate multiple scales like CamOdoCal [13]. First, the least square solutions $\boldsymbol{t}_c^e$ and s are obtained by SVD. We then construct a nonlinear least-squares optimization residual model with the SVD solutions as initial values:

$$ \boldsymbol{C} = \sum_{i=0}^{n} (\boldsymbol{R}(\hat{\boldsymbol{q}}_{e_i}^{e_{i+1}}) - \boldsymbol{I})\boldsymbol{t}_c^e - s\boldsymbol{R}(\boldsymbol{q}_c^e)\hat{\boldsymbol{t}}_{c_i}^{c_{i+1}} + \hat{\boldsymbol{t}}_{e_i}^{e_{i+1}} \tag{15} $$

The $\boldsymbol{t}_c^e$, $\boldsymbol{q}_c^e$and s can be further refined by minimizing the above residual function using nonlinear optimization. Finally, we derive the extrinsic parameters $\boldsymbol{R}_I^e$ of the IMU and the wheel encoder by equation (2). For optimal performance, we iterate the whole calibration process described in III-A.3 and III-A.4 several times by making the results of the last iteration as the initial values of the current iteration.

B. IMU-Wheel Encoder Pre-Integration

For dead-reckoning, we pre-integrate the measurements of the wheel encoder and gyroscope by integrating the linear speed, and angular velocity [15]. IMU timestamps and wheel encoder timestamps are not strictly equal. We assume a constant acceleration motion model and interpolate the IMU measurements linearly in timestamps of the wheel encoder measurements. The pre-integration equation is shown in equation (16):

$$ \begin{aligned} \boldsymbol{\alpha}_{b_{k+1}}^{b_k} &= \int\int_{t\in[k,k+1]} \boldsymbol{R}_t^{b_k}(\hat{\boldsymbol{a}}_t - \boldsymbol{b}_{a_t})dt^2 \\ \boldsymbol{\beta}_{b_{k+1}}^{b_k} &= \int_{t\in[k,k+1]} \boldsymbol{R}_t^{b_k}(\hat{\boldsymbol{a}}_t - \boldsymbol{b}_{a_t})dt \\ \boldsymbol{\gamma}_{b_{k+1}}^{b_k} &= \int_{t\in[k,k+1]} \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}}_t - \boldsymbol{b}_{\omega_t})\boldsymbol{\gamma}_t^{b_k}dt \\ \boldsymbol{\eta}_{e_{k+1}}^{b_k} &= \int_{t\in[k,k+1]} \boldsymbol{R}_t^{b_k}\boldsymbol{R}_e^b\hat{\boldsymbol{v}}_{e_t}dt \end{aligned} \tag{16} $$

where we follow the definitions of $\boldsymbol{\alpha}_{b_{k+1}}^{b_k}$, $\boldsymbol{\beta}_{b_{k+1}}^{b_k}$ and $\boldsymbol{\gamma}_{b_{k+1}}^{b_k}$ in [20] for simplicity, and we further add $\boldsymbol{\eta}_{e_{k+1}}^{b_k}$ as the counterpart of wheel encoder dead-reckoning. And $\boldsymbol{R}_e^b = \boldsymbol{R}_e^I$ is the rotational component of the IMU-wheel encoder extrinsic parameter.

C. Initialization

To initialize the system, VIO such as Vins-mono [20] loosely coupled the camera with IMU and initialized system states such as scale, speed, and gravity direction sequentially given the motion of the vehicle is sufficient in 3D space. However, ground vehicles often have minor acceleration changes, which hinders the performance of the classic initialization algorithms. An inevitable error in scale recovery would occur in the initialization, and the scale error will indirectly affect the calculation of the gravity direction, resulting in an error in the roll-pitch angle of the initialization attitude. Therefore, we incorporate the information of the wheel encoder in initialization to solve the problem. First we use the IMU pre-integration value $\boldsymbol{\gamma}_{b_{k+1}}^{b_k}$ and the results of the visual SFM to initialize the bias of the gyroscope [20], and then use the scale information of the wheel encoder to initialize other states:

$$ \begin{aligned} \boldsymbol{t}_{b_{k+1}}^{b_k} &= \boldsymbol{\eta}_{e_{k+1}}^{b_k} - \boldsymbol{R}_{b_{k+1}}^{b_k}\boldsymbol{t}_e^b + \boldsymbol{t}_e^b \\ s\boldsymbol{R}_c^b\boldsymbol{t}_{c_{k+1}}^{c_k} &= \boldsymbol{R}_{b_{k+1}}^{b_k}\boldsymbol{t}_c^b + \boldsymbol{t}_{b_{k+1}}^{b_k} - \boldsymbol{t}_c^b \end{aligned} \tag{17} $$

where $\boldsymbol{t}_{b_{k+1}}^{b_k}$ is the translation of the IMU, $\boldsymbol{R}_{b_{k+1}}^{b_k}$ is the rotation of the IMU, $\boldsymbol{t}_e^b$ is the translational component, $\boldsymbol{t}_{c_{k+1}}^{c_k}$ is the translation of the camera, $\boldsymbol{t}_c^b$ is the translational part of the IMU-camera extrinsic parameters, $\boldsymbol{R}_c^b$ is the rotational part of the IMU-camera extrinsic parameters, and s is the scale. By calculating the norm of the vector of equation (17), we can get:

$$ s\left\|\boldsymbol{R}_c^b\boldsymbol{t}_{c_{k+1}}^{c_k}\right\|_2 = \left\|\boldsymbol{R}_{b_{k+1}}^{b_k}\boldsymbol{t}_c^b + \boldsymbol{t}_{b_{k+1}}^{b_k} - \boldsymbol{t}_c^b\right\|_2 \tag{18} $$

We reformulate the equation for all the m periods:

$$ s\boldsymbol{A} = \boldsymbol{b} \tag{19} $$

where $\boldsymbol{A} = \begin{bmatrix} \left\|\boldsymbol{R}_c^b \boldsymbol{t}_{c_1}^{c_0}\right\|_2 \cdots \left\|\boldsymbol{R}_c^b \boldsymbol{t}_{c_{m+1}}^{c_m}\right\|_2 \end{bmatrix}^T$

$\boldsymbol{b} = \begin{bmatrix} \left\|\boldsymbol{R}_{b_1}^{b_0} \boldsymbol{t}_c^b + \boldsymbol{t}_{b_1}^{b_0} - \boldsymbol{t}_c^b\right\|_2 \cdots \left\|\boldsymbol{R}_{b_{m+1}}^{b_m} \boldsymbol{t}_c^b + \boldsymbol{t}_{b_{m+1}}^{b_m} - \boldsymbol{t}_c^b\right\|_2 \end{bmatrix}^T$

The scale information s can be calculated by solving (19). We then fix s to optimize the speed and gravity direction as in the following:

$$\boldsymbol{\chi}_I = \left[\boldsymbol{v}_{b_0}^{b_0}, \boldsymbol{v}_{b_1}^{b_1}, \cdots, \boldsymbol{v}_{b_m}^{b_m}, \boldsymbol{g}^{c_0}\right] \tag{20}$$

where $\boldsymbol{v}_{b_k}^{b_k}$ is the IMU velocity when the $m-th$ image is taken, and $\boldsymbol{g}^{c_0}$ is the gravity vector in the c_0 frame. For two consecutive frames b_k and b_{k+1}, the pre-integrated value and the visual observation can be formulated as the following equation:

$$\begin{aligned} \hat{\boldsymbol{z}}_{b_{k+1}}^{b_k} &= \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} - \boldsymbol{p}_c^b + \boldsymbol{R}_{c_0}^{b_k} \boldsymbol{R}_{b_{k+1}}^{c_0} \boldsymbol{p}_c^b - \boldsymbol{p}_{b_{k+1}}^{b_k} \\ \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \end{bmatrix} \\ &= \boldsymbol{H}_{b_{k+1}}^{b_k} \boldsymbol{\chi}_I + \boldsymbol{n}_{b_{k+1}}^{b_k} \end{aligned} \tag{21}$$

where

$$\boldsymbol{p}_{b_{k+1}}^{b_k} = s\boldsymbol{R}_{c_0}^{b_k}\left(\boldsymbol{p}_{c_{k+1}}^{c_0} - \boldsymbol{p}_{c_k}^{c_0}\right) \tag{22}$$

$$\boldsymbol{H}_{b_{k+1}}^{b_k} = \begin{bmatrix} -\boldsymbol{I}\Delta t_k & \boldsymbol{0} & \frac{1}{2}\boldsymbol{R}_{c_0}^{b_k}\Delta t_k^2 \boldsymbol{R}_{c_0}^{b_k}\Delta t_k \\ -\boldsymbol{I} & \boldsymbol{R}_{c_0}^{b_k}\boldsymbol{R}_{c_0}^{c_0} & \end{bmatrix} \tag{23}$$

$\hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k}$ and $\hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k}$ are the pre-integrated measurements disturbed by noises as in [20]. By solving the following linear least-squares problem:

$$\min_{\chi_I} \sum_{k\in j} \left\|\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k} - \boldsymbol{H}_{b_{k+1}}^{b_k}\boldsymbol{\chi}_I\right\|^2 \tag{24}$$

we can obtain the velocity, gravity direction and scale information in the image frame for each frame in the sliding window.

D. Nonlinear Optimization

Similar to Vins-mono [20] and VINS-Fusion [8], we also exploit the sliding window optimization with marginalization. We extend the pre-integration with the aforementioned wheel encoder item and weight this item by the average angular velocity of the pre-integration. The full state vector in the sliding window is defined as:

$$\begin{aligned} \boldsymbol{\chi} &= \left[\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots \boldsymbol{x}_n, \boldsymbol{x}_c^b, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1, \cdots \boldsymbol{\lambda}_m\right] \\ \boldsymbol{x}_k &= \left[\boldsymbol{p}_{b_k}^w, \boldsymbol{v}_{b_k}^w, \boldsymbol{q}_{b_k}^w, \boldsymbol{b}_a, \boldsymbol{b}_g\right], k = 0 \ldots k-1 \\ \boldsymbol{x}_c^b &= \left[\boldsymbol{t}_c^b, \boldsymbol{q}_c^b\right] \end{aligned} \tag{25}$$

where λ_m is the inverse distance of the $m-th$ feature from its first observation, $\boldsymbol{x}_k$ is the state of the IMU when the kth image is taken. $\boldsymbol{p}_{b_k}^w$, $\boldsymbol{v}_{b_k}^w$, $\boldsymbol{q}_{b_k}^w$ are position, velocity, and orientation of the IMU frame in the world frame, $\boldsymbol{b}_a$, $\boldsymbol{b}_g$ are acceleration and gyroscope biases in the IMU frame. n is the total number of keyframes, and m is the total number of features in the sliding window, $\left[\boldsymbol{t}_c^b, \boldsymbol{q}_c^b\right]$ is camera-IMU extrinsic parameters [20].

(a) (b)

Fig. 3. Vehicle platform used in our experiments. (a) Our self-driving car. (b) KAIST dataset car.

Our bundle adjustment fusing the measurements from the camera, IMU and wheel encoder can be formulated as the following equation:

$$\begin{aligned} \boldsymbol{\chi}^* = \arg\min \Bigg\{ &\|\boldsymbol{r}_p - \boldsymbol{H}_p\boldsymbol{\chi}\|^2 + \sum_{k\in B} \left\|\boldsymbol{r}_B\left(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k}, \boldsymbol{\chi}\right)\right\|_{\boldsymbol{P}_{b_{k+1}}^{b_k}}^2 \\ &+ \sum_{(l,j)\in C} \rho\left(\left\|\boldsymbol{r}_C\left(\hat{\boldsymbol{z}}_l^{c_j}, \chi\right)\right\|_{\boldsymbol{P}_l^{c_j}}^2\right) \\ &+ \sum_{(l,j)\in E} \mu\left(\left\|\boldsymbol{r}_E\left(\hat{\boldsymbol{z}}_{e_{k+1}}^{e_k}, \boldsymbol{\chi}\right)\right\|_{\boldsymbol{P}_{b_{k+1}}^{b_k}}^2\right)\Bigg\} \end{aligned} \tag{26}$$

where $\{\boldsymbol{r}_p, \boldsymbol{H}_p\}$ is the marginalized prior information, $\boldsymbol{r}_C\left(\hat{\boldsymbol{z}}_l^{C_j}, \boldsymbol{\chi}\right)$ is the reprojection residual of feature points, ρ is the robust sum function, $\boldsymbol{r}_B\left(\hat{\boldsymbol{z}}_{b_{k+1}}^{b_k}, \boldsymbol{\chi}\right)$ is the residual of IMU pre-integration, and $\boldsymbol{r}_E\left(\hat{\boldsymbol{z}}_{e_{k+1}}^{e_k}, \boldsymbol{\chi}\right)$ is the residual of the IMU-wheel encoder pre-integration. The residual is defined as:

$$\begin{aligned} \boldsymbol{r}_E\left(\hat{\boldsymbol{z}}_{e_{k+1}}^{e_k}, \boldsymbol{\chi}\right) = \Big[&\boldsymbol{R}_w^{b_k}\left(\boldsymbol{p}_{b_{k+1}}^w - \boldsymbol{p}_{b_k}^w\right) - \boldsymbol{t}_e^b \\ &+ \boldsymbol{R}_w^{b_k}\boldsymbol{P}_{b_{k+1}}^w \boldsymbol{t}_e^b - \tilde{\boldsymbol{\eta}}_{b_{k+1}}^{b_k}\Big] \end{aligned} \tag{27}$$

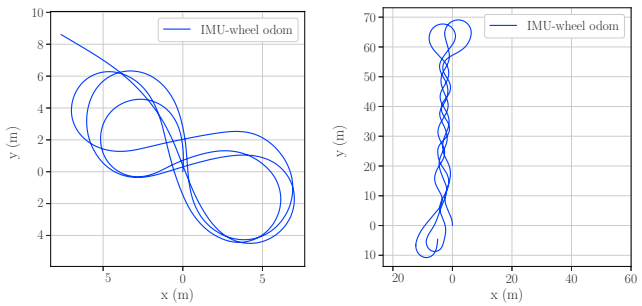
where $\tilde{\boldsymbol{\eta}}_{b_{k+1}}^{b_k}$ is the pre-integrated value of the IMU-wheel encoder corrected by the IMU bias [15]. μ is the wheel encoder weight factor, defined as :

$$\mu(\bar{\omega}) = \begin{cases} 1 & \bar{\omega} \le \omega_{\max} \\ 0 & \bar{\omega} > \omega_{\max} \end{cases} \tag{28}$$

where $(\bar{\omega})$ is the average angular velocity during two consecutive images computed from pre-integration, and $\omega_{\max}$ is a pre-defined angular velocity threshold. To solve the problem of wheel slippage, the wheel encoder is not used when the angular velocity is greater than this threshold.

IV. Experiment and Analysis

We evaluated the extrinsic parameter calibration algorithm and the novel localization algorithm on the self-driving car of UISEE. We further validated the proposed localization algorithm on the KAIST dataset. The examples of our self-driving car and the KAIST dataset cars are shown in Fig. 3.



(a) (b)

Fig. 4. Trajectory examples of extrinsic parameter calibration datasets, (a) '8'-like trajectory. (b) 'S'-like trajectory.

A. Extrinsic Parameter Calibration

1) Accuracy of extrinsic parameter calibration: We compared the results of the proposed calibration method with the factory-calibrated results. The factory calibration uses a chessboard and assistant support to force the chessboard perpendicular to the axle of the car. The length between the chessboard and axle could be exactly measured. And the extrinsic parameters could then be derived with the length and chessboard images as input. We drove the car in different patterns to collect multiple datasets for calibration. The trajectories for calibration are shown in Fig. 4. For each dataset, we compared the proposed calibration method with different settings and the factory calibration method:

1) Calibration using IMU.
2) Calibration without IMU.
3) Factory calibration

Where the method of calibration without IMU refers to [13], The extrinsic parameters are calibrated by the three methods, as shown in Table I. The calibration results of the proposed methods are very close to the factory-calibrated result, except for the translation along the z-axis. This is because the calibration dataset lacks variation in the z-axis. Although there is a large error in the calibration result along the z-axis, we still declare our calibration method as the first one that has successfully recovered the full 6-DoF extrinsic parameters for ground vehicles. And hopefully, the error will be eliminated given sufficient variation in the height direction. We also notice the rotation angles (Pitch-Yaw-Roll) of the extrinsic parameters don't show much difference whether IMU is incorporated in dead-reckoning or not. This also proves the versatility of the proposed framework.

2) Stability of extrinsic parameter calibration: To evaluate the stability of the calibration, we calibrated the extrinsic parameters of the same vehicle five times in different environments and calculated the standard deviation and maximum difference of the calibration results, as shown in Table II. Since dead-reckoning without IMU is based on the assumption of a planar environment, the calibration results without IMU have shown obvious inconsistency, which is because the environments have different conditions, and the assumption does not hold. Both the standard deviation and the maximum difference are large. On the contrary, the calibration results with IMU are less affected by the environment, and the calibration results are more stable.

TABLE I

CALIBRATION RESULTS OF THE THREE METHODS, DIFF. IS THE DIFFERENCE FROM FACTORY CALIBRATION, AND '\' IN THE TABLE REPRESENTS THE ITEM CANNOT BE CALIBRATED.

	Factory	Without IMU		Proposed	
	Results	Results	Diff.	Results	Diff.
x(m)	-0.0085	-0.0167	-0.0082	-0.0091	**-0.0007**
y(m)	-0.2002	-0.2028	**-0.0026**	-0.2245	-0.0243
z(m)	1.4474	\	\	1.6123	**0.1649**
Pitch (°)	89.1753	89.2063	**0.0310**	89.3579	0.1826
Yaw (°)	-0.0475	0.7853	0.8328	0.1478	**0.1953**
Roll (°)	179.0870	179.0837	**-0.0033**	179.1986	0.1116

TABLE II

STANDARD DEVIATION AND MAXIMUM DIFFERENCE OF THE CALIBRATION RESULTS, '\' IN THE TABLE REPRESENTS THE ITEM CANNOT BE CALIBRATED.

	Without IMU		Using IMU	
	STD	range	STD	range
x(m)	**0.0110**	**0.0326**	0.0220	0.0603
y(m)	0.0227	0.0674	**0.0076**	**0.0176**
z(m)	\	\	**0.3202**	**0.8594**
Pitch (°)	0.1443	**0.3976**	**0.1393**	0.4293
Yaw (°)	0.2681	0.8386	**0.1245**	**0.3383**
Roll (°)	0.4453	1.2193	**0.0413**	**0.1168**

B. Initialization Algorithm Test

To verify the two-stage initialization algorithm proposed in this paper, we tested our proposed method on sequences 26, 28, 38, and 39 of the KAIST dataset. We randomly selected a frame to start initialization. When the initialization is completed, we recorded the time distance between the current frame and the start frame and the keyframe poses during the initialization process. We compared our algorithm with the VIO initialization method of VINS-Fusion. The time used for initialization, the results of scale error and pose estimation error are shown in Table III. The scale error is the scale ratio of the ground truth value to the estimated scales. The pose estimation error is the root mean square error(RMSE), which is the absolute trajectory error (ATE) after scale alignment. As can be seen from Table III, in terms of initialization time, the time consumed by our algorithm is less than that of VINS-Fusion. Because most of the cars in the KAIST dataset have small motion excitations, resulting in small changes in acceleration. The less variation of acceleration has impeded VINS-Fusion from

TABLE III

INITIALIZATION TIME, SCALE ERROR AND POSE ESTIMATION ERROR.

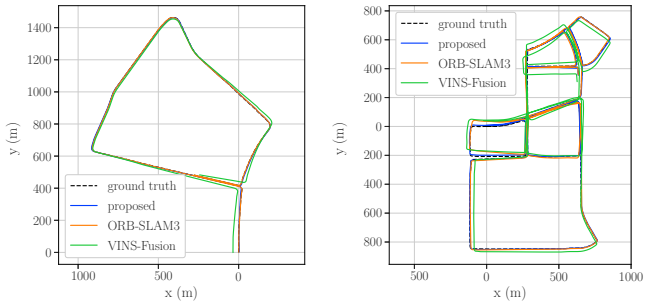
	VINS-Fusion			Proposed		
KAIST	Time(s)	Scale	RMSE	Time(s)	Scale	RMSE
26	3.7004	254.39	0.3374	**1.0003**	**0.9975**	**0.1557**
28	0.9997	65.38	0.1810	0.9997	**1.0036**	**0.0386**
38	1.6000	13.45	0.0688	**1.0000**	**1.0061**	**0.0294**
39	1.5999	226.77	0.3572	**1.0000**	**1.0412**	**0.0686**

TABLE IV

COMPARISON OF TRAJECTORY ACCURACY ON KAIST (RMSE ATE IN M)

KAIST-urban	[5]	[15]	[21]	[27]	[8]	[28]	[7]	Proposed without μ	proposed
26(4.0km)	12.0	11.9	14.8	16.1	20.7	32.8	7.3	4.0486	**3.9943**
28(11.5km)	15.4	27.8	25.0	33.1	23.4	34.7	12.3	9.9755	**8.2709**
38(11.4km)	11.8	16.0	33.5	43.0	50.9	55.5	32.9	10.9307	**8.3603**
39(11.0km)	7.5	8.0	21.3	24.0	32.8	33.4	15.6	8.1448	**6.8761**

Note: Results for [5], [15], [21], [27] and [28] are referred to [5].



(a) (b)

Fig. 5. Ground truth and estimated trajectories of various methods on KAIST dataset. (a)urban26,(b)urban39

quick and accurate initialization, which further degrades the scale estimation. The two-stage initialization method proposed in our work effectively solves the scale problem during initialization. And the pose accuracy initialized by the proposed method is higher than that of VINS-Fusion.

C. Pose Estimation Performance

We conducted localization accuracy experiments on the KAIST dataset and autonomous vehicles in real-world scenarios. The root mean square error (RMSE), which is the absolute trajectory error (ATE) after 6-DoF trajectory alignment, is adopted for comparison in our experiments. The ground truth provided in the KAIST dataset is used for evaluation. And data from high-precision RTK is used as ground truth in the real-car evaluation. All the algorithms are evaluated without performing loop closure.

1) Pose Estimation Evaluation on Public Datasets: We tested our proposed method, VINS-Fusion using the stereo camera and IMU [8], and ORB-SLAM3 [7] using the stereo camera on the KAIST [10] dataset. For other approaches we use the evaluation results from [5], including the approaches using a camera, IMU, and wheel encoder [5], [15], [21], [27], and the method using a camera and IMU [28]. This experiment is performed on the urban26, urban28, urban38 and urban39 sequences in the KAIST dataset, which is also selected in [5]. The evaluation results are shown in Table IV. The trajectory comparison of urban26 and urban28 sequences is shown in Fig. 5.

From Table IV, we can see that the trajectory generated by the algorithm in this paper is better than other algorithms in the four sequences, and the accuracy is improved by adding the wheel encoder weight factor in pre-integration.

TABLE V

COMPARISON OF POSE ESTIMATION ACCURACY OF SELF-DRIVING CAR DRIVING IN INDUSTRIAL PARK ENVIRONMENT (RMSE ATE IN M).

length	VINS-Fusion -VIO	ORB-SLAM3 -VO	ORB-SLAM3 -VIO	proposed
2.32km	18.6115	14.655	36.6981	**1.9027**

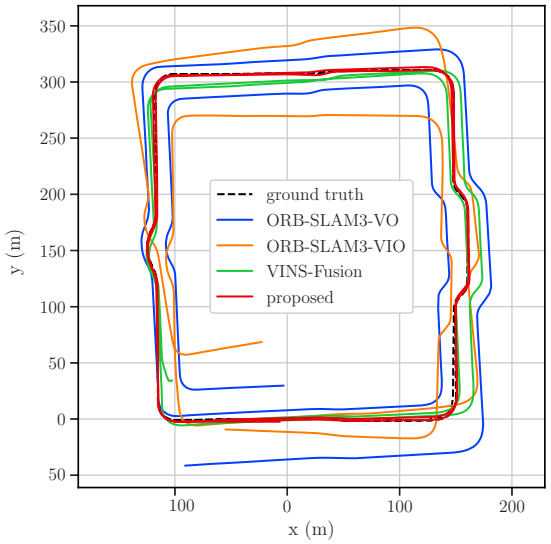


Fig. 6. Trajectories estimated by the proposed method and other methods.

2) Pose Estimation Evaluation in a Real-world Experiment: We deployed our proposed method to the self-driving car, which is shown in Fig. 3 (a). We drove the vehicle around the inner road of the industrial park and recorded the data of the camera, IMU, and wheel encoder for evaluation. The trajectory is visualized on Google Maps in Fig. 1. We compared our proposed method with the VIO of VINS-Fusion, VO and VIO of ORB-SLAM3. The evaluation results are shown in Table V, and the trajectories generated by the four algorithms are shown in Fig. 6 (a). The estimated scales of VINS-Fusion-VIO and ORB-SLAM-VIO drift due to the constant acceleration motions the vehicle undergoes. And it can be seen that the performance of the proposed method in the self-collected dataset is significantly better than that of VINS-Fusion and ORB-SLAM3.

V. CONCLUSION

In this paper, we considered the data fusion problem of cameras, IMUs, and wheel encoders for SLAM on ground vehicles and proposed a method that can calibrate

the camera-wheel encoder extrinsic parameters. First, we calibrated the camera-wheel encoder extrinsic parameters by incorporating the measurement of the gyroscope, which enables the full 6-DoF calibration and improves the accuracy and robustness. Second, we proposed a novel initialization algorithm for VIO by fusing a wheel encoder, which solves the scale degradation problem during VIO initialization. Finally, we proposed the camera-IMU-wheel encoder SLAM system and introduced a novel wheel encoder weight factor for optimal performance. We provided a comprehensive evaluation of a self-collected self-driving dataset and KAIST Urban dataset [10] and proved the practical feasibility of our method. And experimental result demonstrates that our proposed method can provide more accurate pose estimation in comparison with the state-of-the-art algorithms. In this paper, the extrinsic calibration is performed offline, and we will try to integrate the calibration processing in the SLAM algorithm to perform online calibration to further simplify the whole system.

REFERENCES

[1] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Robotics: science and systems*, vol. 4, no. Citeseer. Citeseer, 2007, p. 1.

[2] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4670–4677.

[3] B. Klingner, D. Martin, and J. Roseborough, "Street view motion-from-structure-from-motion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 953–960.

[4] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, "A light-weight semantic map for visual localization towards autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 248–11 254.

[5] J. Liu, W. Gao, and Z. Hu, "Bidirectional trajectory computation for odometer-aided visual-inertial slam," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1670–1677, 2021.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

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

[8] Q. Tong, C. Shaozu, P. Jie, L. Peiliang, and S. Shen, "Vins-fusion: An optimization-based multi-sensor state estimator," 2019.

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

[10] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[12] C. X. Guo, F. M. Mirzaei, and S. I. Roumeliotis, "An analytical least-squares solution to the odometer-camera extrinsic calibration problem," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3962–3968.

[13] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1793–1800.

[14] Y. He, Y. Guo, A. Ye, and K. Yuan, "Camera-odometer calibration and fusion using graph based optimization," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 1624–1629.

[15] J. Liu, W. Gao, and Z. Hu, "Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5391–5397.

[16] W. Lee, K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Visual-inertial-wheel odometry with online calibration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4559–4566.

[17] G. Antonelli, F. Caccavale, F. Grossi, and A. Marino, "Simultaneous calibration of odometry and camera for a differential drive mobile robot," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5417–5422.

[18] H. Tang and Y. Liu, "A fully automatic calibration algorithm for a camera odometry system," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 4208–4216, 2017.

[19] L. Wang, "Vins-gps-wheel: Visual-inertial odometry coupled with wheel encoder and gnss." https://github.com/Wallong/VINS-GPS-Wheel, 2021.

[20] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[21] M. Zhang, X. Zuo, Y. Chen, Y. Liu, and M. Li, "Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1081–1099, 2021.

[22] S. F. Campbell, "Steering control of an autonomous ground vehicle with application to the darpa urban challenge," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[23] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4304–4311.

[24] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3662–3669.

[25] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image and Vision Computing*, vol. 68, pp. 14–27, 2017.

[26] K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *proceedings of 13th International Conference on Pattern Recognition*, vol. 1. IEEE, 1996, pp. 318–322.

[27] M. Zhang, Y. Chen, and M. Li, "Vision-aided localization for ground robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2455–2461.

[28] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in *Robotics: Science and Systems*. Berlin Germany, 2013, pp. 241–248.