



Accurate and robust odometry by fusing monocular visual, inertial, and wheel encoder

Yuqian Niu¹ · Jia Liu¹ · Xia Wang¹ · Wei Hao¹ · Wenjie Li¹ · Lijun Chen¹

Received: 15 June 2020 / Accepted: 19 August 2020 / Published online: 10 September 2020
© China Computer Federation (CCF) 2020

Abstract

Tracking the pose of a robot has been gaining importance in the field of Robotics, e.g., paving the way for robot navigation. In recent years, monocular visual–inertial odometry (VIO) is widely used to do the pose estimation due to its good performance and low cost. However, VIO cannot estimate the scale or orientation accurately when robots move along straight lines or circular arcs on the ground. To address the problem, in this paper we take the wheel encoder into account, which can provide us with stable translation information as well as small accumulated errors and momentary slippage errors. By jointly considering the kinematic constraints and the planar moving features, an odometry algorithm tightly coupled with monocular camera, IMU, and wheel encoder is proposed to get robust and accurate pose sensing for mobile robots, which mainly contains three steps. First, we present the wheel encoder preintegration theory and noise propagation formula based on the kinematic mobile robot model, which is the basis of accurate estimation in backend optimization. Second, we adopt a robust initialization method to obtain good initial values of gyroscope bias and visual scale in reality, by making full use of the camera, IMU and wheel encoder measurements. Third, we bound the high computation complexity with a marginalization strategy that conditionally eliminates unnecessary measurements in the sliding window. We implement a prototype and several extensive experiments showing that our system can achieve robust and accurate pose estimation, in terms of the scale, orientation and location, compared with the state-of-the-art.

Keywords Multi-sensor fusion · Visual–inertial–wheel encoder odometry · State estimation · Localization · Robots

1 Introduction

Robot localization has been gaining importance in the field of Robotics, spanning from robot navigation, three-dimensional reconstruction to simultaneous localization and

mapping (SLAM). Visual inertial odometry (VIO) is a common way for robot localization. By fusing the measurements captured by camera and IMU, VIO can make the metric scale together with the pitch and roll angles observable, which (especially for the scale) underlies the tasks like SLAM and navigation. In addition, the VIO sensor with small size is easy-to-deploy on mobile robots, unmanned aerial vehicles, and handheld devices. In spite of these advantages, VIO requires generic three-dimensional motion along different directions, which is hard to satisfy in practice as the robot usually moves horizontally. Even worse, Wu et al. (2017) have proved that the pitch and roll angles and scale will be unobservable when robot moves along straight lines or circular arcs on the ground. This motivates us to seek for another vehicle to address this problem.

In this paper, we propose an accurate and robust odometry by jointly using VIO and the wheel encoder (VIWO), and expect to obtain the benefits from both: VIO has accurate relative translation and rotation information and meanwhile the wheel encoder provides absolute scale based on

✉ Jia Liu
jjaliu@nju.edu.cn

✉ Lijun Chen
chenlj@nju.edu.cn

Yuqian Niu
niuYuqian@smail.nju.edu.cn

Xia Wang
wangxia@smail.nju.edu.cn

Wei Hao
hw@smail.nju.edu.cn

Wenjie Li
wenjielee@smail.nju.edu.cn

¹ State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

long-time, high-frequency, and stable translation and rotation information. However, fusing VIO and wheel encoder is not easy. First, the wheel slips sometimes, which leads to some measurement errors in scale. Second, the system requires good initial values as the input, which is vital to accurate localization. Third, the VIO system suffers from a high degree of nonlinearity itself. When the wheel encoder is taken into account, this degree will be higher, which makes the pose estimation harder and the computation overhead heavier.

VIWO overcomes these challenges with the following three steps. First, we propose a kinematic motion scheme that deals with the accumulated slippage errors by using a preintegration model and a noise propagation model. Second, we obtain good initial values by loosely aligning IMU and odometer preintegration with the vision-only structure and simplify the pose estimation by considering the constraints from odometer and planar motion in the back-end optimization. Third, we improve a sliding window update strategy and reduce the computation overhead by removing unnecessary measurements. In summary, our contributions are threefold. (1) We propose an accurate and robust odometry system called VIWO that combines the sensor information of visual, IMU, and wheel encoder. (2) We address the key issues of slippage errors and a high degree of nonlinearity by jointly using a kinematic motion scheme together with an improved sliding window update strategy. (3) We implement a prototype of VIWO and several experiments showing that VIWO can achieve higher accuracy of pose estimation, compared with the baseline.

2 Related work

In recent years, there are many excellent works on SLAM, including monocular visual SLAM and visual–inertial odometry (VIO). Although monocular visual SLAM, such as ORB-SLAM (Mur-Artal et al. 2015) and DSO (Engel et al. 2018), can generate compact and trackable map, it is unable to acquire accurate pitch, roll and absolute scale. Instead, VIO is capable of making the metric scale together with the pitch, roll angles observable. VIO can usually be divided into filtering-based method (Mourikis and Roumeliotis 2007) and nonlinear optimization-based method (Leutenegger et al. 2014; Mur-Artal and Tardós 2016; Qin et al. 2018). As a model of the former, MSCKF (Mourikis and Roumeliotis 2007) is an Extended Kalman Filter (EKF)-based method which constraints the IMU and camera pose at the same time, as well as the multiple camera poses that have the same feature observation. The latter introduces nonlinear optimization methods based on the sliding window. OKVIS (Leutenegger et al. 2014) applies monocular and stereo camera, and integrates the inertial measurements in

advance, then achieves feature detection by using BRISK (Leutenegger et al. 2011) algorithm. However, OKVIS (Leutenegger et al. 2014) preintegrates the inertial measurements repeatedly when the linearization point changes and it has no implementation of loop closing. ORB-VISLAM (Mur-Artal and Tardós 2016) introduces inertial measurements based on ORB-SLAM (Mur-Artal et al. 2015), optimizing the inertial error term between two frames and achieving zero-drift localization in mapped areas. VINS-Mono (Qin et al. 2018) is another VIO framework that fulfills a low computation relocation module and its improvement (Qin et al. 2019) can be conveniently extended with other sensors such as GPS.

Some researches work on visual–inertial–wheel encoder odometry. As proven in Wu et al. (2017), VIO has unobservabilities when robot moves without generic 3D motion, such as along straight lines or circular arcs. To solve this problem, Wu et al. (2017) makes the scale, pitch and roll observable by incorporating odometer measurements and planar motion constraints. Li et al. (2017) presents a gyro-aided camera-odometer online calibration and localization method, which is based on the stereo vision without the scale estimation and the initial calibration. Furthermore, Liu et al. (2019) considers both gyroscope and accelerometer measurements in the preintegration and optimization, but fails to pay an attention to the significance of angular velocity of wheel encoder and the planar motion constraint. DRE-SLAM (Yang et al. 2019) fuses the information of RGB-D camera and wheel encoder, then constructs OctoMap in both dynamic and static environments.

The initialization methods are widely adopted in SLAM and the initial values have great influences on the accuracy of the system. In earlier studies (Yang and Shen 2017; Shen et al. 2016; Martinelli 2014), the initialization methods only utilize relative rotation from the IMU, without considering gyroscope bias and image noises. Kaiser et al. (2017) introduces the gyroscope bias calibration, but requires double integration of IMU measurements. ORB-VISLAM (Mur-Artal and Tardós 2016) proposes an IMU initialization method, which is able to compute the scale, gravity direction, velocity, and gyroscope and accelerometer biases in a few seconds with high accuracy. In order to improve efficiency, VINS-Mono (Qin et al. 2018) loosely aligns IMU preintegration with the vision-only structure without considering accelerometer bias.

However, visual–inertial odometry suffers from the additional unobservabilities when the robot moves along straight lines or circular arcs, the metric scale and other values can not be able to be initialized accurately. We propose in this paper an accurate and robust odometry by jointly using VIO and the wheel encoder, called VIWO, expecting to obtain the accurate relative translation and the rotation information, as well as the absolute scale.

3 System model

3.1 Notations

We begin with defining the notations used throughout this paper. \mathbf{R}_M^N denotes the rotation matrix from frame M to frame N , and \mathbf{q}_M^N is its quaternion form. \mathbf{p}_M^N denotes the translation vector from frame M to frame N . w is world frame. b_k, c_k and o_k are IMU frame, camera frame and odometer frame respectively when we obtain the k th image.

Besides, the extrinsic parameters between the IMU and odometer are presented as $\mathbf{R}_B^O, \mathbf{q}_B^O, \mathbf{p}_B^O$ and \mathbf{T}_B^O , which indicates the rotation matrix, quaternion form, translation vector and transformation matrix from IMU frame to odometer frame respectively. Similarly, the extrinsic parameters between the IMU and camera are presented as $\mathbf{R}_B^C, \mathbf{q}_B^C, \mathbf{p}_B^C$ and \mathbf{T}_B^C , which indicates the rotation matrix, quaternion form, translation vector and transformation matrix from IMU frame to camera frame respectively. The extrinsic parameters between the IMU and odometer are calibrated manually and the extrinsic parameters between the IMU and camera are estimated by tightly coupled nonlinear optimization in Sect. 6. In addition, $(n)_x$ is vector $[n, 0, 0]^T$, $(n)_z$ is vector $[0, 0, n]^T$, and (\cdot) refers to its noisy measurement.

3.2 System definition

The input for our system is a stream of monocular camera frames, IMU measurements and odometer measurements. The raw measurements from IMU are composed of angular velocity vector $\hat{\omega}_t$ and linear accelerated velocity vector $\hat{\mathbf{a}}_t$. The measurements from odometer are composed of $\hat{\omega}_t$ and \hat{v}_t , which means angular velocity and linear velocity

respectively. And monocular camera captures a series of grayscale images.

The output for our system is a state vector in the sliding window including robot poses and velocities, 3D feature locations, acceleration bias and gyroscope bias and extrinsic parameters.

The structure of the proposed VIWO system is shown in Fig. 1 and the details are presented in the following sections.

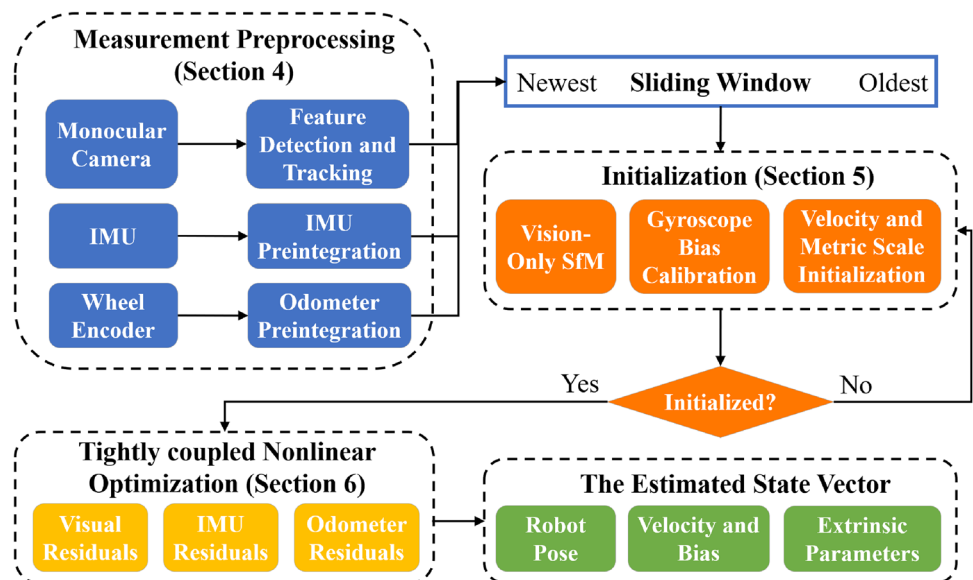
4 Measurement preprocessing

This section presents preprocessing methods for monocular visual, IMU and odometer measurements. Monocular visual measurement preprocessing is responsible for extracting features and tracking relative transformation between two consecutive frames. IMU and odometer measurement preprocessing are responsible for calculating preintegration between two selected keyframes. For monocular visual and IMU measurements, we adopt the existing methods to perform measurement preprocessing. For the odometer measurements, we propose a novel method to perform measurement preprocessing, reducing the accumulated slippage errors.

4.1 Visual measurements

The preprocessing of visual measurements refers to visual processing front-end in VINS-Mono (Qin et al. 2018). We detect the features for each image using GFTT corner detection algorithm (Jianbo and Tomasi 1994), which is an improved corner detection algorithm based on Harris (Mikolajczyk and Schmid 2004), and adopt KLT sparse optical flow algorithm (Lucas and Kanade 1997) for pose tracking. After obtaining multiple sets of matched features,

Fig. 1 The structure of the proposed VIWO system



we also use RANSAC algorithm (Fischler and Bolles 1981) to adjust feature locations and eliminate outliers. Finally, we get relative visual rotation and translation of camera, as well as the detected feature locations.

4.2 Preintegration of IMU measurements

We employ VINS-Mono algorithm (Qin et al. 2018) to calculate IMU preintegration. The preintegration of translation $\alpha_{b_k}^{b_{k+1}}$, velocity $\beta_{b_k}^{b_{k+1}}$ and rotation $\mathbf{R}_{b_k}^{b_{k+1}}$ between two consecutive frames b_k and b_{k+1} can be presented as

$$\begin{aligned} \alpha_{b_k}^{b_{k+1}} &= \int \int_{t \in [k, k+1]} [\mathbf{R}_{b_t}^{b_k} ((\hat{\mathbf{a}}_t - \mathbf{b}_t^a - \mathbf{n}^a) - \mathbf{R}_{b_t}^{b_k} \mathbf{g}^w)] \delta t^2, \\ \beta_{b_k}^{b_{k+1}} &= \int_{t \in [k, k+1]} [\mathbf{R}_{b_t}^{b_k} ((\hat{\mathbf{a}}_t - \mathbf{b}_t^a - \mathbf{n}^a) - \mathbf{R}_{b_t}^{b_k} \mathbf{g}^w)] \delta t, \\ \mathbf{R}_{b_k}^{b_{k+1}} &= \int_{t \in [k, k+1]} \mathbf{R}_{b_t}^{b_k} \exp((\hat{\boldsymbol{\omega}}_t - \mathbf{b}_t^g - \mathbf{n}^g)^\wedge) \delta t, \end{aligned} \tag{1}$$

where $\hat{\mathbf{a}}_t$, \mathbf{b}_t^a and \mathbf{n}^a are measurement, bias and Gaussian white noise of accelerator respectively and $\hat{\boldsymbol{\omega}}_t$, \mathbf{b}_t^g and \mathbf{n}^g are corresponding terms of gyroscope.

4.3 Preintegration of odometer measurements

The wheel slippage is one of the main reasons of measurement errors in scale. To solve this problem, we propose a novel kinematic motion scheme to deal with the accumulated slippage errors by using a preintegration model and a noise propagation model between two keyframes. The kinematic motion scheme consists of three steps: *constructing motion model*, *constructing preintegration model* and *constructing noise propagation model*.

4.3.1 Constructing motion model

According to the motion characteristics of the kinematic mobile robot, we construct the motion model. The raw linear velocity $\hat{\mathbf{v}}$ in the forward direction of robot and yaw angular velocity \hat{w} measured at the time t can be given:

$$\begin{aligned} \hat{v}_t &= v_t + \frac{n_r + n_l}{2}, \\ \hat{w}_t &= w_t + \frac{n_r - n_l}{l}. \end{aligned} \tag{2}$$

We assume that additive Gaussian white noises exist in both left and right wheels, where $n_l \sim \mathcal{N}(0, \sigma_l^2)$ and $n_r \sim \mathcal{N}(0, \sigma_r^2)$. Here, l is the distance between left and right wheel center. To simplify the description, we use $n_w = \frac{n_r - n_l}{l}$ and $n_v = \frac{n_r + n_l}{2}$ to present rotation and translation noises in the following sections.

4.3.2 Constructing preintegration model

Since the sampling frequency of odometer is much higher than camera, we integrate the odometer measurements between two consecutive frame b_k and b_{k+1} , which can be given as follows:

$$\begin{aligned} \mathbf{R}_{b_k}^{b_{k+1}} &= \int_{t \in [k, k+1]} \mathbf{R}_{b_t}^{b_k} \exp[(\mathbf{R}_O^B(\hat{w}_t - n_w)_z)^\wedge] \delta t, \\ \mathbf{p}_{b_k}^{b_{k+1}} &= \int_{t \in [k, k+1]} \mathbf{R}_{b_t}^{b_k} [\mathbf{R}_O^B(\hat{v}_t - n_v)_x \delta t + \mathbf{p}_O^B - \mathbf{R}_{b_{t-1}}^{b_t} \mathbf{p}_O^B], \end{aligned} \tag{3}$$

where $(\cdot)^\wedge$ means transformation to skew symmetric matrix form, and $\exp(\xi^\wedge)$ is exponential mapping from Lie algebra $\mathfrak{so}(3)$ to Lie group $\mathbf{SO}(3)$ that means transformation from rotation vector to rotation matrix physically. Furthermore, we can discover that the preintegration model contains the Gaussian white noises at each moment between two consecutive frames.

4.3.3 Constructing noise propagation model

We observe that the noises from odometer are accumulated and propagated with the preintegration processing. For decreasing the effects of accumulated odometer noises, we construct a noise propagation model to separate the noises from preintegration model.

Odometer preintegration between two consecutive frame b_k and b_{k+1} contains the rotation term $\mathbf{R}_{b_k}^{b_{k+1}}$ and the translation term $\mathbf{p}_{b_k}^{b_{k+1}}$. According to first-order approximation of Taylor expansion, we can split the preintegration into measurements and noises as follows:

$$\begin{aligned} \mathbf{R}_{b_k}^{b_{k+1}} &= \int_{t \in [k, k+1]} \mathbf{R}_{b_t}^{b_k} \exp[(\mathbf{R}_O^B(\hat{w}_t - n_w)_z)^\wedge] \delta t \\ &= \hat{\mathbf{R}}_{b_k}^{b_{k+1}} \int_{t \in [k, k+1]} \exp(-\mathbf{J}_{rt} \mathbf{R}_O^B(n_w)_z)^\wedge \delta t \\ &\triangleq \hat{\mathbf{R}}_{b_k}^{b_{k+1}} \exp((-\delta \varphi_{o_k}^{o_{k+1}})^\wedge), \end{aligned} \tag{4}$$

and

$$\begin{aligned} \mathbf{p}_{b_k}^{b_{k+1}} &= \int_{t \in [k, k+1]} \hat{\mathbf{R}}_{b_k}^{b_{k+1}} \exp((-\delta \varphi_{o_k}^{o_{k+1}})^\wedge) \\ &\quad \times [\mathbf{V}_t - \mathbf{R}_O^B(n_v)_x \delta t + \mathbf{p}_O^B - \mathbf{R}_{b_{t-1}}^{b_t} \mathbf{p}_O^B] \\ &= \hat{\mathbf{p}}_{b_k}^{b_{k+1}} + \int_{t \in [k, k+1]} \hat{\mathbf{R}}_{b_k}^{b_{k+1}} \delta \varphi_{o_k}^{o_{k+1}} \\ &\quad \times [\mathbf{V}_t - \mathbf{R}_O^B(n_v)_x \delta t + \mathbf{p}_O^B - \mathbf{R}_{b_{t-1}}^{b_t} \mathbf{p}_O^B]^\wedge \\ &\triangleq \hat{\mathbf{p}}_{b_k}^{b_{k+1}} + \delta p_{o_k}^{o_{k+1}}, \end{aligned} \tag{5}$$

where $\mathbf{V}_t = \mathbf{R}_O^B(\hat{v}_t)_x \Delta t$ and $\mathbf{J}_{rt} = \mathbf{J}_r(\mathbf{R}_O^B(\hat{w}_t - n_w)_z \Delta t)$ is the right jacobian on Lie group $\mathbf{SO}(3)$.

As a result, we can get the odometer rotation error term $\delta\varphi_{o_k}^{o_{k+1}}$ and the translation error term $\delta p_{o_k}^{o_{k+1}}$ from Eqs. (4) and (5), which can be given:

$$\begin{aligned} \delta\varphi_{o_k}^{o_{k+1}} &= \int_{t \in [k, k+1]} \exp(-\mathbf{J}_{rt} \mathbf{R}_O^B(n_w)_z)^\wedge \delta t, \\ \delta p_{o_k}^{o_{k+1}} &= \int_{t \in [k, k+1]} \hat{\mathbf{R}}_{b_k}^{b_{k+1}} \delta\varphi_{o_k}^{o_{k+1}} \\ &\quad \times [\mathbf{V}_t - \mathbf{R}_O^B(n_v)_x \delta t + \mathbf{p}_O^B - \mathbf{R}_{b_{t-1}}^{b_t} \mathbf{p}_O^B]^\wedge. \end{aligned} \tag{6}$$

Furthermore, we can get the accumulated error term in $0 \sim k$ th frame, $\delta\varphi_{o_0}^{o_k}$, $\delta p_{o_0}^{o_k}$ and the accumulated error term in $0 \sim (k + 1)$ th frame, $\delta\varphi_{o_0}^{o_{k+1}}$, $\delta p_{o_0}^{o_{k+1}}$. And the relationship between them can be written as matrix form.

$$\begin{aligned} \begin{bmatrix} \delta\varphi_{o_0}^{o_{k+1}} \\ \delta p_{o_0}^{o_{k+1}} \end{bmatrix} &= \begin{bmatrix} \hat{\mathbf{R}}_{b_{k+1}}^{b_k} & \mathbf{0}_{3 \times 3} \\ \hat{\mathbf{R}}_{b_0}^{b_k} (\mathbf{V}_k - \mathbf{R}_O^B(\frac{n_r+n_l}{2})_x \Delta t + \mathbf{p}_O^B - \mathbf{R}_{b_k}^{b_{k+1}} \mathbf{p}_O^B)^\wedge & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \delta\varphi_{o_0}^{o_k} \\ \delta p_{o_0}^{o_k} \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\mathbf{J}_{rk} \mathbf{R}_O^B \Delta t}{2} & -\frac{\mathbf{J}_{rk} \mathbf{R}_O^B \Delta t}{2} \\ \frac{\mathbf{R}_{b_0}^{b_{k-1}} \mathbf{R}_O^B \Delta t}{2} & \frac{\mathbf{R}_{b_0}^{b_{k-1}} \mathbf{R}_O^B \Delta t}{2} \end{bmatrix} \begin{bmatrix} (\mathbf{n}_r)_{3 \times 1} \\ (\mathbf{n}_l)_{3 \times 1} \end{bmatrix} \\ &\triangleq \mathbf{A} \begin{bmatrix} \delta\varphi_{o_0}^{o_k} \\ \delta p_{o_0}^{o_k} \end{bmatrix} + [\mathbf{B} \ \mathbf{C}] \begin{bmatrix} (\mathbf{n}_r)_{3 \times 1} \\ (\mathbf{n}_l)_{3 \times 1} \end{bmatrix}. \end{aligned} \tag{7}$$

According to forward propagation of covariance, the error term $\begin{bmatrix} \delta\varphi_{o_i}^{o_j} \\ \delta p_{o_i}^{o_j} \end{bmatrix}$ satisfies Gaussian distribution $\mathcal{N}(\mathbf{0}_{6 \times 1}, \Sigma_{o_i}^{o_j})$. Therefore, the covariance matrix can be written as:

$$\Sigma_{o_0}^{o_{k+1}} = \mathbf{A} \Sigma_{o_0}^{o_k} \mathbf{A}^T + \mathbf{B} \Sigma_{\mathbf{n}_r} \mathbf{B}^T + \mathbf{C} \Sigma_{\mathbf{n}_l} \mathbf{C}^T. \tag{8}$$

5 Robust initialization

Robust initialization plays a significant role in the processing of nonlinear optimization, which requires well-performed initial guess at the beginning. However, limited to the visual and IMU information, the initialization in VINS-Mono (Qin et al. 2018) suffers from poor scale and orientation results when moving along straight lines or circular arcs on the ground. Therefore, we propose a new method that takes the odometer preintegration into consideration, providing excellent initial values. We first adopt the sliding window vision-only SfM strategy to achieve the feature observations and relative rotations among different frames. Then, we take advantage of the IMU, the odometer preintegration and the rotation results to provide well-performed initial values including velocity, gyroscope bias, scale, gravity vector, robot poses and feature locations.

5.1 Sliding window vision-only SfM

First, we calculate the relative rotation and translation of frames using SfM algorithm (Wu 2013). Specifically, the five-point algorithm (Nister 2004) devotes to the essential matrix calculation, and the perspective-n-point (PnP) method (Lepetit et al. 2009) devotes to the poses of all frames estimation. If there are stable feature tracking and sufficient parallax compared with other frames in the sliding window, we can obtain the feature observations by minimizing the reprojection errors using the global bundle adjustment (Triggs et al. 2000).

5.2 Gyroscope bias calibration

Upon obtaining the monocular camera rotation $\mathbf{q}_C^{c_{k+1}}$ between the k th and $(k + 1)$ th frames in the sliding window, gyroscope bias can be calculated as:

$$\min_{\delta b_g} \sum_{k \in \mathcal{F}} \|\mathbf{q}_C^B \otimes \mathbf{q}_C^{c_{k+1}} \otimes \mathbf{q}_B^C \otimes \mathbf{q}_{b_{k+1}}^{b_k}\|^2, \tag{9}$$

where $\mathbf{q}_{b_{k+1}}^{b_k} \approx \hat{\mathbf{q}}_{b_{k+1}}^{b_k} \otimes \left[\frac{1}{2} \mathbf{J}_{b_g}^g \delta b_g \right]$. Similarly, the odometer rotation $\mathbf{q}_O^{o_{k+1}}$ can also be used to calculate the gyroscope bias:

$$\min_{\delta b_g} \sum_{k \in \mathcal{F}} \|\mathbf{q}_O^B \otimes \mathbf{q}_O^{o_{k+1}} \otimes \mathbf{q}_B^O \otimes \mathbf{q}_{b_{k+1}}^{b_k}\|^2. \tag{10}$$

The gyroscope bias from Eqs. (9) and (10) are marked as b_{g_1} and b_{g_2} respectively. Then the final result is $\frac{1}{2}(b_{g_1} + b_{g_2})$. Finally, we do repropagation of all IMU preintegrations under the new b_g .

5.3 Velocity and metric scale initialization

The scale is an important feature in initialization procedure, so we optimize the metric scale, the velocity and gravity simultaneously. Firstly, the state vector to be estimated is $\mathcal{X}_I = [\mathbf{v}_0^b, \mathbf{v}_2^b, \dots, \mathbf{v}_n^b, s]$, where \mathbf{v}_k^b is the linear velocity in IMU frame while taking the k th image and s is the metric scale. Here, we assume that the direction and magnitude of gravity are known, $\mathbf{g}^{c_0} = [0, g, 0]^T$, as the robot moves on the planar ground. So, we can get $\hat{\mathbf{z}}_{b_{k+1}}^{b_k}$ and $\mathbf{H}_{b_{k+1}}^{b_k}$ between two consecutive frames as:

$$\hat{\mathbf{z}}_{b_{k+1}}^{b_k} = \begin{bmatrix} \hat{\mathbf{p}}_{b_{k+1}}^{b_k} - \mathbf{p}_C^B + \hat{\mathbf{R}}_{b_{k+1}}^{b_k} \mathbf{p}_C^B - \frac{1}{2} \mathbf{R}_{b_0}^{b_k} \mathbf{R}_C^B \mathbf{g}^{c_0} \Delta t_k^2 \\ \hat{\mathbf{v}}_{b_{k+1}}^{b_k} - \mathbf{R}_{b_0}^{b_k} \mathbf{R}_C^B \mathbf{g}^{c_0} \Delta t_k \\ \mathbf{R}_{b_0}^{b_k} \mathbf{R}_O^B \hat{\mathbf{p}}_{o_{k+1}}^{o_k} + (\mathbf{p}_O^B - \mathbf{p}_C^B) - \hat{\mathbf{R}}_{b_{k+1}}^{b_k} (\mathbf{p}_O^B - \mathbf{p}_C^B) \end{bmatrix}, \tag{11}$$

$$\mathbf{H}_{b_{k+1}}^{b_k} \mathcal{X}_I = \begin{bmatrix} -\mathbf{I}_{3 \times 3} \Delta t_k & \mathbf{0}_{3 \times 3} & \mathbf{R}_{b_0}^{b_k} \mathbf{R}_C^B \bar{\mathbf{p}}_{c_{k+1}}^{c_k} \\ -\mathbf{I}_{3 \times 3} & \mathbf{R}_C^B \bar{\mathbf{R}}_{c_{k+1}}^{c_k} \mathbf{R}_B^C & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{R}_{b_0}^{b_k} \mathbf{R}_C^B \bar{\mathbf{p}}_{c_{k+1}}^{c_k} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b_k}^b \\ \mathbf{v}_{b_{k+1}}^b \\ s \end{bmatrix}, \quad (12)$$

where $\mathbf{R}_{b_0}^{b_k}$ is the preintegration term from the 0th to k th image in IMU frame, and $\bar{\mathbf{R}}_{c_{k+1}}^{c_k}$, $\bar{\mathbf{p}}_{c_{k+1}}^{c_k}$ are visual rotation and translation after SfM algorithm.

According to the constraint that the translation from camera, IMU and odometer under the b_k frame should be the same, we can obtain the following least square equation:

$$\min_{\mathcal{X}_I} \sum_{k \in \mathcal{F}} \|\hat{\mathbf{z}}_{b_{k+1}}^{b_k} - \mathbf{H}_{b_{k+1}}^{b_k} \mathcal{X}_I\|^2, \quad (13)$$

which can use SVD (Golub and Reinsch 1970) to get vector \mathcal{X}_I . If the metric scale in the vector \mathcal{X}_I is positive, the velocities in the sliding window and metric scale are initialized successfully. Furthermore, the gravity refinement is implemented to correct the direction and magnitude again by the method in VINS-Mono (Qin et al. 2018). Finally, all the variables are transformed into the world frame, and the robot poses and feature locations are in the absolute scale.

6 Tightly coupled nonlinear optimization

This section tightly couples all known measurements and state vectors to be estimated in the sliding window based on bundle adjustment (Triggs et al. 2000), which is presented in the factor map form. We also consider the constraints from odometer and planar motion to simplify the pose estimation.

The full state vector to be estimated in the back-end optimization sliding window is defined as:

$$\begin{aligned} \mathcal{X} &= [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n, \lambda_0, \lambda_1, \dots, \lambda_m, \mathbf{p}_C^B, \mathbf{q}_C^B], \\ \mathbf{x}_k &= [\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n], \end{aligned} \quad (14)$$

where n is the number of keyframes, and m is the number of features in the sliding window. \mathbf{x}_k means the IMU state when the system get the k th keyframe, which contains position $\mathbf{p}_{b_k}^w$, velocity $\mathbf{v}_{b_k}^w$, orientation $\mathbf{q}_{b_k}^w$, accelerometer bias \mathbf{b}_a and gyroscope bias \mathbf{b}_g . λ_l is the inverse distance of the l th feature from its first observation. \mathbf{p}_C^B and \mathbf{q}_C^B are extrinsic parameters to be estimated.

6.1 Bundle adjustment

The state estimation problem refers to estimate the inner state from the noisy data. We obtain the visual, IMU and odometer measurements from Sect. 4, and the estimated state vector shown in Eq. (14). However, the state vector calculated only by preprocessing and initialization is

not optimal, which still exists the accumulative errors. To address this problem, we optimize the robot poses and other variables mentioned in the state vector by adopting bundle adjustment, which is a tightly coupled nonlinear optimization approach.

Furthermore, in order to implement online optimization, we use a sliding window to save a certain amount of states as input for bundle adjustment model. With the participation of new preprocessed measurements, the states are constantly updated according to the marginalization strategy.

In visual-inertial-wheel encoder state estimation system, the bundle adjustment is achieved by minimizing the sum of prior and the Mahalanobis norm of all measurement residuals, which can be given as follows:

$$\begin{aligned} \min_{\mathcal{X}} \left\{ \|\mathbf{r}_p - \mathbf{H}_p \mathcal{X}\|^2 + \sum_{k \in \mathcal{F}} \|\mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X})\|_{\Sigma_{b_{k+1}}^{b_k}}^2 \right. \\ \left. + \sum_{k \in \mathcal{F}} \|\mathbf{r}_O(\hat{\mathbf{z}}_{o_{k+1}}^{o_k}, \mathcal{X})\|_{\Sigma_{o_{k+1}}^{o_k}}^2 \right. \\ \left. + \sum_{l \in \mathcal{L}} \sum_{j \in \mathcal{F}_l} \rho(\|\mathbf{r}_C(\hat{\mathbf{z}}_{c_{j,l}}, \mathcal{X})\|_{\Sigma_{c_{j,l}}}) + \sum_{k \in \mathcal{F}} \|\mathbf{r}_{pl}\|_{\Sigma_{pl}}^2 \right\}, \end{aligned} \quad (15)$$

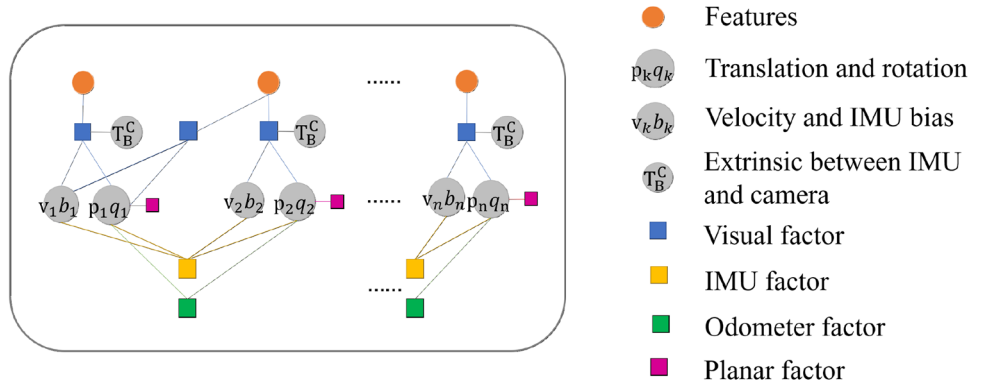
where $(\mathbf{r}_p - \mathbf{H}_p \mathcal{X})$ is the prior information after marginalization, \mathcal{F} means all the frames in the sliding window, $\mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathcal{X})$ is the IMU measurement residual, $\mathbf{r}_O(\hat{\mathbf{z}}_{o_{k+1}}^{o_k}, \mathcal{X})$ is the odometer measurement residual, \mathcal{L} and \mathcal{F}_l means all the features and the frames when the feature l appears, $\mathbf{r}_C(\hat{\mathbf{z}}_{c_{j,l}}, \mathcal{X})$ is the visual measurement residual, ρ is the Huber cost function (Huber 1964), \mathbf{r}_{pl} is the planar constraint residual, and $\Sigma_{b_{k+1}}^{b_k}$, $\Sigma_{o_{k+1}}^{o_k}$ and $\Sigma_{c_{j,l}}$, Σ_{pl} are the covariances corresponding to the residuals.

6.2 Measurement residuals

In the sliding window, we have the state vector composed of rotation $\mathbf{R}_{b_k}^w$, translation $\mathbf{p}_{b_k}^w$ and velocity $\mathbf{v}_{b_k}^w$ in each frame. At the same time, we have rotation and translation measurements of each sensor, $\hat{\mathbf{p}}_{o_{k+1}}^{o_k}$, $\hat{\mathbf{R}}_{o_{k+1}}^{o_k}$, $\hat{\mathbf{p}}_{b_{k+1}}^{b_k}$, $\hat{\mathbf{R}}_{b_{k+1}}^{b_k}$, $\hat{\mathbf{p}}_{c_{k+1}}^{c_k}$, $\hat{\mathbf{R}}_{c_{k+1}}^{c_k}$. In this way, there are residuals between the state vectors and the measurements, which can be used to eliminate the accumulative errors caused by measurement preprocessing.

Each sensor has the residual term of its own, which is associated with the variables to be estimated and the measurements obtained in the preprocessing model. The visual and IMU residuals are the same as VINS-Mono (Qin et al. 2018), and we analyze the odometer residuals and planar residuals in detail. According to the constraints, we can construct the factor map as Fig. 2. We can find that the visual factor constraints multiple robot poses, feature locations, velocity, IMU bias and extrinsic parameters between IMU and camera. The IMU factor constraints two consecutive

Fig. 2 The factor map that describes the optimization. The circles represent states to be estimated, and the squares represent the constrained edges derived by measurements. Each square constrains any number of states



robot poses, velocity and IMU bias. The odometer factor also constrains two consecutive robot poses. The planar factor constrains its own pose individually.

6.2.1 Visual residuals

For each detected feature in the sliding window, we calculate the reprojection errors by projecting the feature pixel coordinates to a unit sphere. First, we can find the first frame i and other frames in which the l th feature appears. Then we reproject the l th feature pixel coordinates of both the first frame i and the frame j onto the unit sphere of frame j to calculate the residual term of the l th feature in the frame j .

$$\mathbf{r}_C(\hat{\mathbf{z}}_{c_j,l}, \mathcal{X}) = \begin{bmatrix} \hat{x}_{c_j} - x_{c_j} \\ \hat{y}_{c_j} - y_{c_j} \\ \hat{z}_{c_j} - z_{c_j} \end{bmatrix}, \tag{16}$$

$$\begin{bmatrix} \hat{x}_{c_j} \\ \hat{y}_{c_j} \\ \hat{z}_{c_j} \end{bmatrix} = \frac{1}{\lambda_l} \boldsymbol{\pi}^{-1} \begin{bmatrix} \hat{u}_{l_j}^c \\ \hat{v}_{l_j}^c \\ 1 \end{bmatrix}, \tag{17}$$

$$= \mathbf{T}_B^C \mathbf{T}_w^{b_j} \mathbf{T}_{b_i}^w \mathbf{T}_C^B \left(\frac{1}{\lambda_l} \boldsymbol{\pi}^{-1} \begin{bmatrix} \hat{u}_{l_i}^c \\ \hat{v}_{l_i}^c \\ 1 \end{bmatrix} \right)_H,$$

where $[\hat{x}_{c_j}, \hat{y}_{c_j}, \hat{z}_{c_j}]^T$ is the reprojection result of the l th feature from the frame j , $[x_{c_j}, y_{c_j}, z_{c_j}, 1]^T$ is the reprojection result of the l th feature from the frame i correspondingly. $\boldsymbol{\pi}^{-1}$ means that project pixel coordinates into unit sphere using intrinsic parameters, λ_l is the inverse depth and $\frac{1}{\lambda_l}$ is the real depth of the l th feature, which helps transform the three-dimensional coordinates into the real world. In order to simplify the computation, we also use transformation matrix \mathbf{T} to present the rotation and translation, and use $(\cdot)_H$ to expand three-dimensional coordinates to homogeneous coordinates formulation.

6.2.2 IMU residuals

The residual of IMU measurements in the sliding window can be defined as:

$$\mathbf{r}_B(\hat{\mathbf{z}}_{b_{k+1}}, \mathcal{X}) = \begin{bmatrix} \delta \boldsymbol{\alpha}_{b_{k+1}}^{b_k} \\ \delta \boldsymbol{\beta}_{b_{k+1}}^{b_k} \\ \delta \mathbf{R}_{b_{k+1}}^{b_k} \\ \delta \mathbf{b}_a \\ \delta \mathbf{b}_g \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^{b_k} (\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w + \frac{1}{2} \mathbf{g}^w \Delta t_k^2 - \mathbf{v}_{b_k}^w \Delta t_k) - \hat{\boldsymbol{\alpha}}_{b_{k+1}}^{b_k} \\ \mathbf{R}_w^{b_k} (\mathbf{v}_{b_{k+1}}^w + \mathbf{g}^w \Delta t_k - \mathbf{v}_{b_k}^w) - \hat{\boldsymbol{\beta}}_{b_{k+1}}^{b_k} \\ \log(\mathbf{R}_w^{b_k} \mathbf{R}_{b_{k+1}}^w \hat{\mathbf{R}}_{b_{k+1}}^{b_k})^\vee \\ \mathbf{b}_{ab_{k+1}} - \mathbf{b}_{ab_k} \\ \mathbf{b}_{gb_{k+1}} - \mathbf{b}_{gb_k} \end{bmatrix}, \tag{18}$$

where $\delta \boldsymbol{\alpha}_{b_{k+1}}^{b_k}$, $\delta \boldsymbol{\beta}_{b_{k+1}}^{b_k}$, $\delta \mathbf{R}_{b_{k+1}}^{b_k}$ are the translation, velocity and rotation error term between measurement and state vector in the two consecutive frames b_k and b_{k+1} . The accelerometer and gyroscope bias in the adjacent two frames should be the same. $\log(\cdot)^\vee$ is the logarithmic mapping from Lie group $\mathbf{SO}(3)$ to Lie algebra $\mathfrak{so}(3)$, which means the transformation from the rotation matrix to the rotation vector.

6.2.3 Odometer residuals

The residual of odometer is associated with the preintegration after preprocessing and the rotation and translation term in the state vector.

$$\mathbf{r}_O(\hat{\mathbf{z}}_{o_{k+1}}, \mathcal{X}) = \begin{bmatrix} \delta \mathbf{p}_{o_{k+1}}^{o_k} \\ \delta \mathbf{R}_{o_{k+1}}^{o_k} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_w^{b_k} (\mathbf{p}_{b_{k+1}}^w - \mathbf{p}_{b_k}^w) - \mathbf{p}_O^B + \mathbf{R}_w^{b_k} \mathbf{R}_{b_{k+1}}^w \mathbf{p}_O^B - \hat{\mathbf{p}}_{o_{k+1}}^{o_k} \\ \log(\mathbf{R}_B^O \mathbf{R}_w^{b_k} \mathbf{R}_{b_{k+1}}^w \mathbf{R}_O^B \hat{\mathbf{R}}_{o_{k+1}}^{o_k})^\vee \end{bmatrix}, \tag{19}$$

where $\hat{\mathbf{p}}_{o_{k+1}}^{o_k}$ and $\hat{\mathbf{R}}_{o_{k+1}}^{o_k}$ are the translation and rotation term of the odometer preintegration between two adjacent frames b_k and b_{k+1} in the sliding window.

6.2.4 Planar residuals

As the robot is moving on a planar ground, there is almost no pitch and roll angle. The translation on the vertical dimension is almost zero. Therefore, the planar residuals can be presented as:

$$\mathbf{r}_{pl} = \begin{bmatrix} (\mathbf{R}_w^{b_r} \mathbf{R}_{b_k}^w)_{roll} \\ (\mathbf{R}_w^{b_r} \mathbf{R}_{b_k}^w)_{pitch} \\ (-\mathbf{R}_w^{b_r} \mathbf{R}_{b_k}^w \mathbf{p}_w^{b_k} + \mathbf{p}_w^{b_r})_z \end{bmatrix}, \quad (20)$$

where b_r means choosing a frame in the sliding window randomly, $(\cdot)_{roll}$ and $(\cdot)_{pitch}$ means transforming rotation matrix into Euler angles firstly, then only select the roll or pitch angle respectively, and $(\cdot)_z$ means only choosing translation in vertical direction.

6.3 Marginalization

For the sake of meeting limited computational resources and the real-time requirement, the frames in the sliding window need to be updated continuously. Firstly, we always keep the latest frame in the sliding window, whether it is a keyframe or not. Keyframe selection depends on the algorithm mentioned in VINS-Mono (Qin et al. 2018). Then we update other frames information by determining if the second latest frame is a keyframe. If it is, we keep this frame in the sliding window, and remove the oldest keyframe, including the visual, IMU and odometer measurements, as well as planar constraints. Otherwise, we remove the visual measurements and planar constraints of the second latest frame, and keep

the IMU and odometer measurements. The process is shown in Fig. 3. Our marginalization strategy keeps the IMU and odometer measurements in the sliding window to provide consecutive motion information. This module is realized by Schur complement (Sibley et al. 2010). Firstly, the prior is constructed based on the marginalized measurements, and then we combine the prior and new measurement information to construct a new information matrix.

7 Experiments

In this section, we evaluate our system VIWO from two aspects, the trajectory accuracy and the initialization robustness. Our experiments are based on the mobile robot platform TurtleBot2 as shown in Fig. 4. The Turtlebot2 is equipped with a 2D scanning LiDAR (SICK LMS100), wheel encoder and a forward-looking global shutter camera integrated with an IMU (MYNTEYE D1000-IR-120/Color). The onboard computation resource is provided by an Intel i5-7300H CPU with 2.40 GHz. The specific camera provides

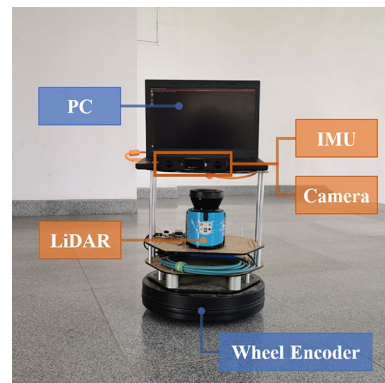
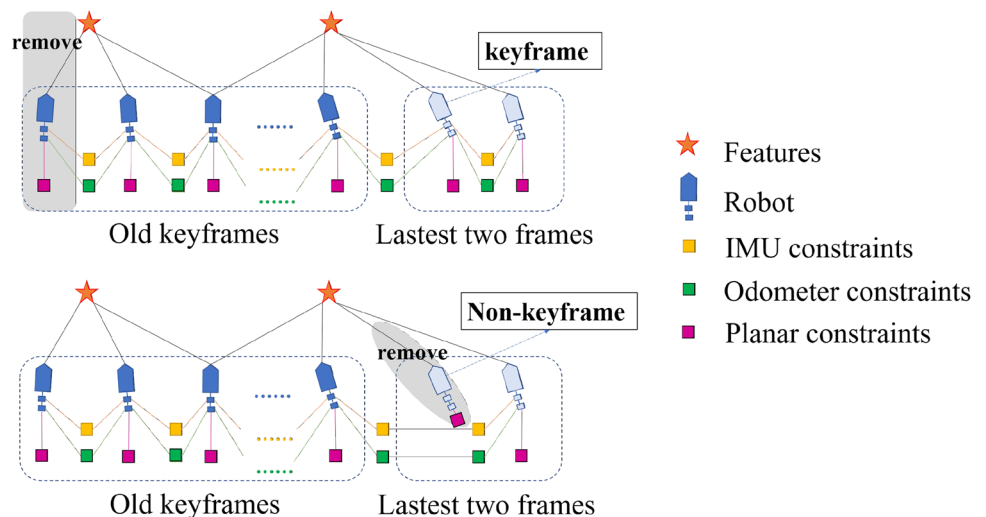


Fig. 4 The mobile robot platform

Fig. 3 The Marginalization strategy. The sliding window is composed of several old keyframes and latest two frames. The different scheme is adopted based on whether the second latest frame is a keyframe or not. The gray region is the measurements to be removed. If the oldest frame is removed, the corresponding measurements should be marginalized. Otherwise, if the second latest frame is removed, we will keep the IMU and odometer measurements and marginalize other information



not only 640×480 resolution gray images at 10 Hz, but also the IMU measurements at 200 Hz, with wheel encoder providing the odometer measurements at 20 Hz.

We conduct experiments under several indoor low-texture scenes such as corridor and lobby. And we collect the data of six sequences: sequence01 to sequence04 that the robot moves along straight lines and pure rotation most of the time, sequence05 that the robot moves along circular arcs and sequence06 that the robot moves freely. The linear velocity is 0.15 m/s in all sequences and the yaw angular velocity is 0.15 rad/s in sequence01 to sequence05. Besides, the trajectory ground truth is obtained by GMapping algorithm (Grisetti et al. 2005, 2007) based on the 2D scanning LiDAR and wheel encoder measurements, and the result grid map is shown in Fig. 5.

We design two groups of experiments to evaluate the trajectory accuracy and the effect of the initialization performance respectively. The first experiment focuses on the trajectory accuracy. We compare our system with the

state-of-the-art algorithm and odometer-only measurements. The second experiment focuses on the effect of initialization method. We perform our initialization method in multiple time periods and calculate the average time and average trajectory accuracy of our method and VINS-Mono (Qin et al. 2018).

7.1 Evaluation of trajectory accuracy

We compare our system with the open source method VINS-Mono (Qin et al. 2018) and Wheel Encoder (odometer-only measurements), and calculate their trajectory accuracy. Table 1 shows the root mean square error (RMSE) of ATE (Sturm et al. 2012) from sequence01 to sequence06. In addition, we compare the trajectory accuracy in the x - y , z and x - y - z directions. We find that the x - y plane trajectory accuracy from our approach is improved dramatically than VIO and odometer-only method. Furthermore, the result shows

Fig. 5 The grid map from GMapping algorithm

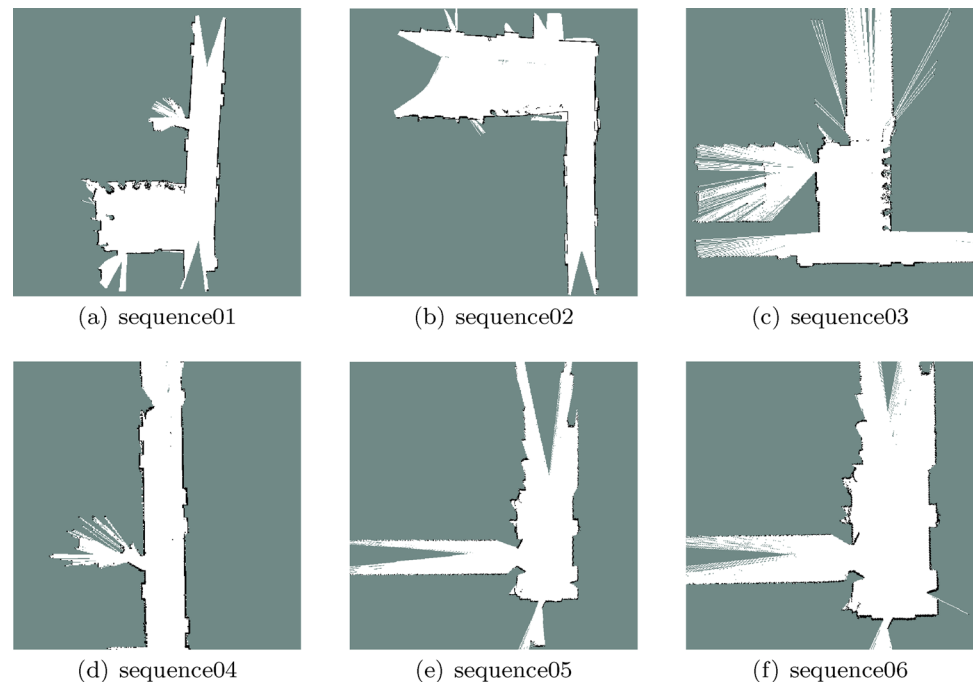


Table 1 RMSE (in meters) for trajectory accuracy of sequences

	VINS-Mono			Wheel encoder			Our system		
	x - y	z	x - y - z	x - y	z	x - y - z	x - y	z	x - y - z
seq01 (37.22 m)	0.70	0.08	0.71	0.30	0.04	0.33	0.29	0.04	0.30
seq02 (55.85 m)	6.13	0.04	6.13	2.10	0.04	2.14	1.10	0.03	1.10
seq03 (16.62 m)	0.74	0.05	0.74	0.30	0.16	0.34	0.15	0.14	0.20
seq04 (28.21 m)	0.78	0.35	0.78	0.89	0.26	0.93	0.27	0.21	0.27
seq05 (31.64 m)	1.71	0.03	1.71	0.37	0.02	0.38	0.03	0.01	0.03
seq06 (19.48 m)	0.43	0.05	0.44	0.27	0.04	0.27	0.10	0.03	0.10

Bold values in each line indicate the minimum RMSE in x - y , z , x - y - z respectively

that planar motion constraints also reduce the errors in the z direction.

As shown in Fig. 6, the whole trajectory results from different approaches are displayed intuitively. We can discover that although the odometer-only method has accumulated errors, absolute metric scale still achieved. The reason is that VINS-Mono (Qin et al. 2018) suffers from scale inaccuracy seriously, but has well-performed trajectory shape in general. The results demonstrate that our system has higher trajectory accuracy as a result of combining the advantages of the existing approaches.

7.2 Initialization

In order to illustrate the experiment results of initialization, we propose two criteria: time required to complete the initialization and the trajectory accuracy during a period of time after initialization. Besides, the back-end optimization and marginalization are kept consistent by tightly coupling the visual, IMU and odometer measurements in this paper. For each sequence, we set the beginning time as multiples of 30 s, and count the time required to finish the initialization. After the initialization, we record the trajectory data in the next 30 s.

We compare the initialization approach proposed in this paper with the method in VINS-Mono (Qin et al. 2018). The average results for each sequence are shown in Fig. 7. We observe that not only the average time required for initialization, but also the trajectory accuracy in the 30 s after initialization is improved.

For each sequence, as shown in Table 2, the average initialization time for our method is less than VINS-Mono (Qin et al. 2018). And at some moment, such as 60 and 120 s for sequence05, the initialization time required in Qin et al. (2018) is longer than 10 s, but all the time durations of our method are less than 10 s, which means that our method is more stable and faster.

Table 3 demonstrates the trajectory accuracy from the x - y - z direction during the 30 s after initialization. There are three empty results because their trajectories fail to converge, because the initialization method in VINS-Mono (Qin et al. 2018) requires generic 3D motion. On the contrary, the trajectories after the initialization method in this paper all succeed to converge and has lower errors, which means more robustness.

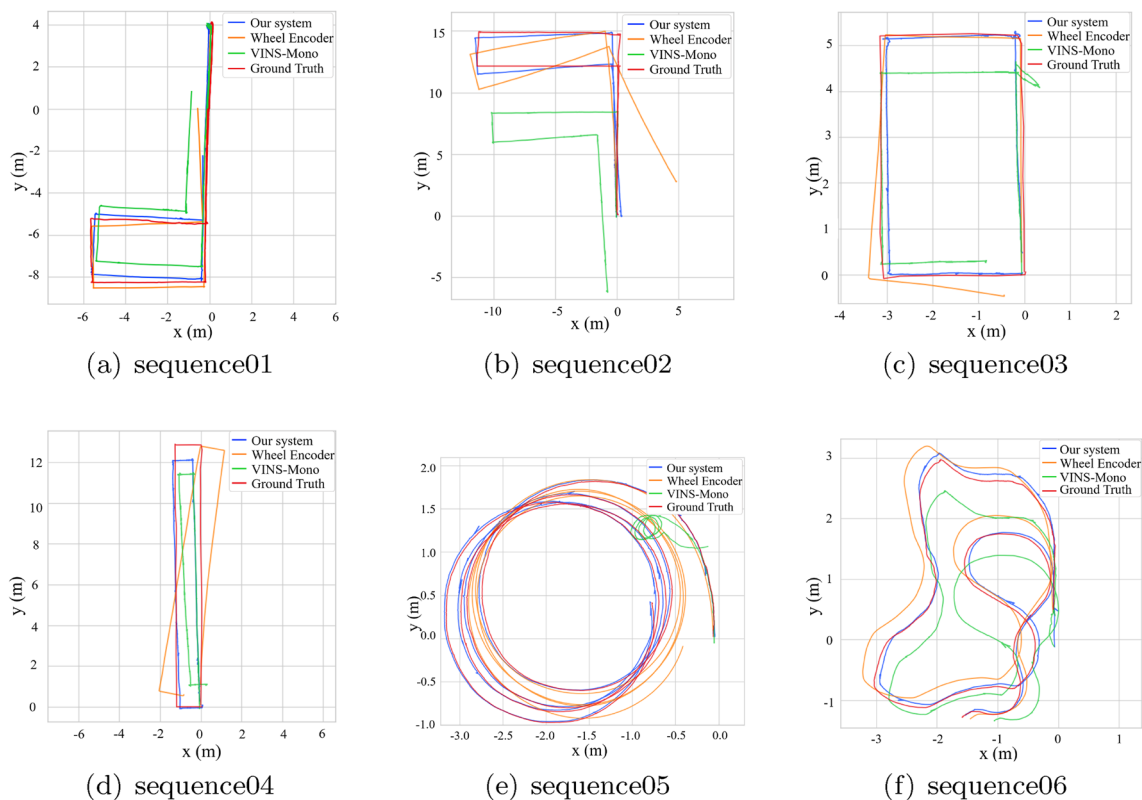


Fig. 6 The trajectory of the proposed method, odometer-only measurements, VINS-Mono and ground truth from GMapping algorithm

Fig. 7 The average time (s) required for initialization and the average RMSE (m) in x–y–z direction in the 30 s after initialization

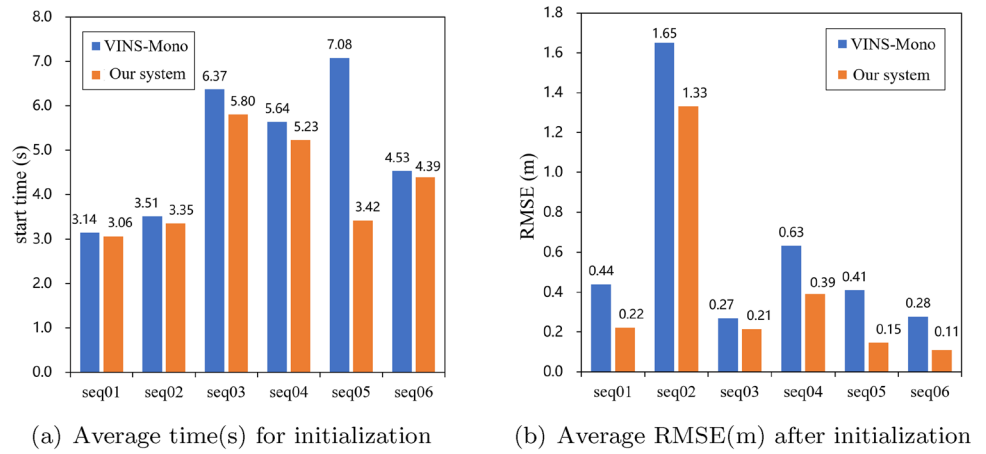


Table 2 Time (in seconds) required to finish initialization

	Start time (s)	30	60	90	120	150	180	210	Average
seq01	VINS-Mono	3.84	2.45	3.16	4.33	2.43	3.31	2.52	3.14
	Our system	3.81	2.44	2.69	4.33	2.39	3.28	2.49	3.06
seq02	VINS-Mono	3.82	1.88	5.87	2.43	3.24	2.55	4.76	3.51
	Our system	3.86	2.21	4.69	2.32	3.27	2.56	4.53	3.35
seq03	VINS-Mono	4.67	9.30	4.04	7.77	6.07			6.37
	Our system	4.38	9.05	4.03	7.72	3.82			5.80
seq04	VINS-Mono	3.68	3.39	3.65	5.36	8.62	9.13		5.64
	Our system	3.34	3.54	3.67	5.89	8.39	6.55		5.23
seq05	VINS-Mono	7.21	12.79	6.55	12.15	1.68	2.10		7.08
	Our system	4.74	3.32	2.78	6.18	1.48	2.00		3.42
seq06	VINS-Mono	2.16	5.46	4.93	4.75	5.36			4.53
	Our system	2.17	5.48	4.86	4.28	5.18			4.39

Bold values indicate the average time required to finish initialization

Table 3 RMSE (in meters) for trajectory accuracy in the 30 s after initialization

	Start time (s)	30	60	90	120	150	180	210	Average
seq01	VINS-Mono	0.23	0.60	0.76	0.36	–	0.39	0.31	0.44
	Our system	0.18	0.16	0.19	0.30	0.15	0.29	0.28	0.22
seq02	VINS-Mono	0.39	2.09	1.60	0.54	–	2.95	2.32	1.65
	Our system	0.36	2.09	1.07	0.48	0.21	3.58	1.52	1.33
seq03	VINS-Mono	0.23	–	0.17	0.37	0.30			0.27
	Our system	0.21	0.17	0.04	0.38	0.28			0.21
seq04	VINS-Mono	0.51	0.89	1.27	0.67	0.16	0.30		0.63
	Our system	0.12	0.88	0.83	0.08	0.20	0.13		0.39
seq05	VINS-Mono	0.24	0.33	0.72	0.56	0.25	0.33		0.41
	Our system	0.12	0.11	0.32	0.16	0.10	0.07		0.15
seq06	VINS-Mono	0.43	0.18	0.13	0.27	0.39			0.28
	Our system	0.03	0.04	0.03	0.10	0.35			0.11

Bold values indicate the average time required to finish initialization

8 Conclusion

In this paper, we propose a VIWO system that tightly couples the measurements of camera, IMU and wheel encoder to provide robust and accurate robot poses. We first deal with the accumulated slippage errors using the kinematic motion scheme with the preintegration model and noise propagation model. Then, we propose a robust and the novel initialization method, improve the sliding window update strategy and reduce the computational overhead. We implement a prototype and several experiments demonstrating that our system can achieve robust and accurate pose estimation, in terms of the scale, orientation and location, compared with the state-of-the-art.

Acknowledgements This work is supported by the National Natural Science Foundation of China (nos. 61702257 and 61771236), Natural Science Foundation of Jiangsu Province (no. BK20170648), Fundamental Research Funds for the Central Universities (14380066), and Collaborative Innovation Center of Novel Software Technology and Industrialization. Jia Liu and Lijun Chen are the corresponding authors.

References

- Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(3), 611–625 (2018)
- Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
- Golub, G., Reinsch, C.: Singular value decomposition and least squares solutions. *Numerische Mathematik* **14**, 403–420 (1970)
- Grisetti, G., Stachniss, C., Burgard, W.: Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In: *Proceedings of IEEE ICRA*, pp. 2432–2437 (2005)
- Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. Robot.* **23**(1), 34–46 (2007)
- Huber, P.J.: Robust estimation of a location parameter. *Ann. Math. Stat.* **35**(1), 73–101 (1964)
- Jianbo, S., Tomasi: Good features to track. In: *Proceedings of IEEE CVPR*, pp. 593–600 (1994)
- Kaiser, J., Martinelli, A., Fontana, F., Scaramuzza, D.: Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation. *IEEE Robot. Autom. Lett.* **2**(1), 18–25 (2017)
- Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* **81**(2), 155–166 (2009)
- Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. In: *Proceedings of IEEE ICCV*, pp. 2548–2555 (2011)
- Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **34**(3), 314–334 (2014)
- Li, D., Eckenhoff, K., Wang, Y., Xiong, R., Huang, G.: Gyro-aided camera-odometer online calibration and localization. In: *Proceedings of IEEE ACC*, pp. 3579–3586 (2017)
- Liu, J., Gao, W., Hu, Z.: Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration. In: *Proceedings of IEEE/RSJ IROS*, pp. 5391–5397 (2019)
- Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *Proceedings of IJCAI*, pp. 674–679 (1997)
- Martinelli, A.: Closed-form solution of visual-inertial structure from motion. *Int. J. Comput. Vis.* **106**, 138–152 (2014)
- Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. *Int. J. Comput. Vis.* **60**(1), 63–86 (2004)
- Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: *Proceedings of IEEE ICRA*, pp. 3565–3572 (2007)
- Mur-Artal, R., Tardós, J.D.: Visual-inertial monocular SLAM with map reuse. *IEEE Robot. Autom. Lett.* **2**(2), 796–803 (2016)
- Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **31**(5), 1147–1163 (2015)
- Nister, D.: An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(6), 756–770 (2004)
- Qin, T., Li, P., Shen, S.: VINS-Mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **34**(4), 1004–1020 (2018)
- Qin, T., Pan, J., Cao, S., Shen, S.: A general optimization-based framework for local odometry estimation with multiple sensors (2019). [arXiv:1901.03638](https://arxiv.org/abs/1901.03638)
- Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Initialization-Free Monocular Visual-Inertial State Estimation with Application to Autonomous MAVs, pp. 211–227 (2016)
- Sibley, G., Matthies, L., Sukhatme, G.: Sliding window filter with application to planetary landing. *J. Field Robot.* **27**(5), 587–608 (2010)
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: *Proceedings of IEEE/RSJ IROS*, pp. 573–580 (2012)
- Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: *Proceedings of IWVA*, pp. 298–372 (2000)
- Wu, C.: Towards linear-time incremental structure from motion. In: *Proceedings of IEEE 3DV*, pp. 127–134 (2013)
- Wu, K.J., Chao, G.X., Georgiou, G., Roumeliotis, S.I.: Vins on wheels. In: *Proceedings of IEEE ICRA*, pp. 5155–5162 (2017)
- Yang, Z., Shen, S.: Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration. *IEEE Trans. Autom. Sci. Eng.* **14**(1), 39–51 (2017)
- Yang, D., Bi, S., Wang, W., Yuan, C., Qi, X., Cai, Y.: DRE-SLAM: dynamic RGB-D encoder SLAM for a differential-drive robot. *Remote Sens.* **11**(4), 380 (2019)



Yuqian Niu is currently a master student in Department of Computer Science and Technology at Nanjing University. She received her Bachelor degree in Software Engineering from Southeast University in 2018. Her research interest includes multi-sensor fusion and visual SLAM.



Jia Liu (Member, IEEE) received the B.E. degree in software engineering from Xidian University, Xi'an, China, in 2010, and the Ph.D. degree in computer science and technology from Nanjing University, Nanjing, China, in 2016. He is currently a Research Associate Professor with the Department of Computer Science and Technology, Nanjing University. His research interest includes RFID systems. He is a member of the ACM.



Wenjie Li is currently a Ph.D. student at the Department of Computer Science and Technology, Nanjing University. He got his Master Degree in Control Theory and Control Engineering from Noreastern University in 2018. His current research interests include visual SLAM and deep learning.



Xia Wang is currently a lecturer in Department of Optoelectronic Engineering at Zaozhuang University. She received her Ph.D. degree in computer science and technology from Nanjing University, Nanjing, China, in 2020. Her research interests include RFID technologies and wireless sensor networks.



Lijun Chen received the B.S. degree in electrical engineering from the Xi'an University of Science and Technology, China, in 1982, and the M.S. and Ph.D. degrees in automatic control from the China University of Mining and Technology, China, in 1993 and 1998, respectively. He was a Post-Doctoral Fellow at Nanjing University, China, from 1998 to 2000, and Michigan State University, USA, from 2001 to 2002, and a Visiting Scholar at The Hong Kong Polytechnic University in 2007. His



Wei Hao is a Ph.D. Candidate in the Department of Computer Science and Technology at Nanjing University. Prior to Nanjing University, he was a Master student in Control Science at Xinjiang university. His current interests include autonomous robotic motion planning, robot visual and imitation learning.

current research interests include distributed computing and quantum communications.