Research Article

Mobile Robot Localization Based on Vision and Multisensor

Lina Yao and **Fengzhe Li**

School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China

Correspondence should be addressed to Lina Yao; michelle_linxq@126.com

Received 19 September 2019; Revised 7 February 2020; Accepted 15 February 2020; Published 8 June 2020

Academic Editor: Keigo Watanabe

Copyright © 2020 Lina Yao and Fengzhe Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To deal with the low accuracy of positioning for mobile robots when only using visual sensors and an IMU, a method based on tight coupling and nonlinear optimization is proposed to obtain a high-precision visual positioning scheme by combining measured value of the preintegrated inertial measurement unit (IMU) and values of the odometer and characteristic observations. First, the preprocessing part of the observation data includes tracking of the image data and the odometer data, and preintegration of IMU data. Second, the initialization part of the above three sensors includes IMU preintegration, odometer preintegration, and gyroscope bias calculation. It also includes the alignment of speed, gravity, and scale. Finally, a local BA (bundle adjustment) joint optimization and global graph optimization are established, so as to obtain more accurate positioning results.

1. Introduction

Positioning technology is undoubtedly the basic premise for a wide range of applications such as robot navigation, autonomous driving, and virtual reality. The visual positioning system has won the attention of researchers because of its small size, low cost, and simple hardware setup [1–4]. In recent years, the visual SLAM (simultaneous location and mapping) has experienced rapid development, among which the more classic methods include the PATM [5] algorithm proposed by Klein et al. in 2007 and its successor ORB-SLAM [6]. Other notable methods are LSD-SLAM [7] for the monocular vision direct method, SVO [8] based on the sparse method, DSO [9] and DVO-SLAM [10], etc. Since the discussion cannot be fully discussed, this section discusses the method of multisensor fusion localization, skipping the above method of localization using only visual sensors. Due to a large amount of information measured by the camera and the low cost, the application on mobile robots is very extensive. Therefore, the following mainly introduces the positioning method combining the camera and other sensors.

The way the camera is integrated with other sensors can be divided into loose coupling [11, 12] and tight coupling. Among them, the loose-coupling method is used to use the

camera and other sensors as separate modules to calculate the respective pose. The fusion information methods generally include Kalman filter (KF), extended Kalman filter (EKF), and other methods, while other sensor information is used for state propagation. The tight-coupling methods are mainly divided into two kinds: EKF [13, 14] and graph-based optimization [15, 16]. A widely concerned VIO method based on EKF is the MSCKF [13] algorithm proposed by MARS Laboratory of the University of Minnesota. It can marginalize old map points and postures to ensure accuracy and computational efficiency under the condition that the number of map points and postures in the state vector is constant. However, with the development of the optimization technology in SLAM, VINS based on graph optimization has also been studied more and more. The limitation of the filtering algorithm is analyzed in reference [17], and the method based on optimization is superior to the method based on the filter. The ICE-BA algorithm proposed in reference [18] improves the BA optimization part of VI-SLAM, changes the incremental nature of SLAM measurement, resolves the direct conflict between sliding window edge deviation and global closed-loop detection constraints, and improves the computational efficiency. In [8], the IMU preintegration technology was extended to Lie algebra in 2015 and implemented in the open-source factor

graph optimization library GTSAM. Based on this, [19, 20] designed the VINS-MONO algorithm and obtained the experimental results that the dynamic performance of continuous-time preintegration is better than the discrete preintegration. However, there are many differences in the adaptability of sensors on different platforms. In the mobile robot, because the robot is mostly in the plane uniform motion [4], the IMU cannot be fully excited, which causes numerical stability problems during the initialization of the scale. Therefore, the applicability of the IMU is not as good as the mileage. At the same time, external calibration and initialization issues between sensors pose significant challenges for highly nonlinear visual-inertial systems.

These problems will affect the positioning accuracy and robustness of mobile robots. To solve these problems, a tightly coupled and optimization-based approach is used to fuse monocular vision sensor, IMU, and wheeled odometer to achieve accurate pose estimation in this paper. The main work of this paper is shown as follows:

- (1) In this paper, the odometer-inertial navigation system is used to estimate the scale to solve the problem that the absolute scale cannot be obtained by location algorithm using monocular camera and scale stability cannot be guaranteed.
- (2) A BA tight-coupling optimization model is established, including marginalization (removal of pose and feature point constraints from the sliding window), IMU residual between adjacent frames in the sliding window, visual reprojection error, and measurement error of odometer. Finally, accurate positioning results are obtained by minimizing the error cost function.

2. System Composition and Measurement Data Preprocessing

2.1. System Composition and Coordinate System Definition. In Figure 1, the system starts with data preprocessing (Section 2.2), including image feature extraction and tracking, preintegration of IMU measurements between two consecutive frames, and calculation of pose measurement by two consecutive interframe odometers. In the second part (Section 3), the initialization process provides the contents including attitude, velocity, gravity vector, gyroscope bias, and three-dimensional feature location for subsequent nonlinear optimization. The third part is local BA joint optimization (Section 4) and global map optimization, including a sliding window BA optimization model; the fourth part is loop detection, eliminating drift and ensuring map consistency. The main application method is DOW2 [21], so it will not be discussed in this paper.

The relevant coordinate system in this paper is shown in Figure 2, which defines the coordinate system and symbols used in this paper. Among them, $(\text{xyz})^w$ represents the world coordinate system, $(\text{xyz})^c$ represents the camera coordinate system, $(\text{xyz})^i$ represents the IMU coordinate system, $(\text{xyz})^d$ represents the odometer coordinate system, and $(\text{xyz})^b$ represents the robot coordinate system, which is

defined as the same as the IMU coordinate system. The rotation matrix R and quaternion q are used simultaneously to represent the rotation. q_b^w and p_b^w represent the rotation and translation from the robot coordinate system to the world coordinate system. b_j represents the coordinate system of the robot when the j frame image is obtained. c_j represents the camera coordinate system when the j frame image is obtained. $\otimes$ denotes multiplication between quaternions. $(\bullet)$ denotes the measured or estimated value of a specific quantity.

2.2. Data Preprocessing

2.2.1. Feature Point Tracking. The tracking and matching of feature points in monocular vision are used to inherit or generate new feature points from the previous frame to the next frame. The specific process is shown as follows: firstly, the strongest f corner features are found on the first frame image, and the nonmaximum suppression radius is set at the same time. Then, when the next frame image arrives, these feature points are tracked by the optical flow method [22], and the matching points are found in the next frame image. The matching points in the front and back frames are corrected to get the final corrected position. Based on the corrected position, the F-matrix and RANSAC algorithm are used to remove the outliers. Finally, the strongest new feature points are searched in the area beyond the matched feature points, and the number of feature points is complemented to f , which ensures that the matching number of feature points is consistent with that of the next frame. The value of f is generally in the range of 100–300, and the specific content is referenced in [23].

2.2.2. Inertial Navigation Preintegration. The original accelerometer and gyroscope can measure the acceleration $\bar{a}$ and angular velocity $\bar{\omega}$ of the robot. It is assumed that the additional noise in the accelerometer and gyroscope is the Gaussian white noise $n_a \sim N(0, \sigma_a^2)$ and $n_\omega \sim N(0, \sigma_\omega^2)$. The bias of accelerometer and gyroscope is assumed to be a random walk model, whose derivatives obey Gaussian normal distribution $\dot{m}_a \sim N(0, \sigma_{b_a}^2)$ and $\dot{m}_\omega \sim N(0, \sigma_{b_\omega}^2)$. The measurement results are illustrated as follows:

$$\begin{aligned}\bar{a}_t &= a_t + m_a + R_t^t g^w + n_a, \\ \bar{\omega}_t &= \omega_t + m_\omega + n_\omega.\end{aligned}\tag{1}$$

From the above formula, the position, velocity, and direction of the next moment can be calculated, and the time interval between the two moments is given as $\Delta t = [j, j+1]$:

$$\begin{aligned}p_{b_{j+1}}^w &= p_{b_j}^w + v_{b_j}^w \Delta t + \iint_{t \in [j, j+1]} (R_{b_t}^w \bar{a}_t - g^w) dt^2, \\ v_{b_{j+1}}^w &= v_{b_j}^w + \int_{t \in [j, j+1]} (R_{b_t}^w \bar{a}_t - g^w) dt, \\ q_{b_{j+1}}^w &= q_{b_j}^w \otimes \int_{t \in [j, j+1]} \frac{1}{2} \Omega \bar{\omega}_t q_t^{b_j} dt,\end{aligned}\tag{2}$$

FIGURE 1: The overall system block diagram.

FIGURE 2: The overall coordinate system.

where

$$\bar{a}_t = \tilde{a}_t - m_{a_t} - n_a, \bar{\omega}_t = \tilde{\omega}_t - m_{\omega_t} - n_w, \quad (3)$$

$$\Omega(\omega) = \begin{bmatrix} -|\omega|_{\times} & \omega \\ -\omega^T & 0 \end{bmatrix}, |\omega|_{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

The reference coordinate system is transformed from the world coordinate system to the robot coordinate system. Formula (2) is multiplied by the transformation matrix $R_w^{b_j}$ from the world coordinate system to the b_j frame coordinate system. By using b_j as the reference frame of a given bias, the preintegration term can be obtained only by the IMU measurement, and the covariance matrix $Q_{b_{j+1}}^{b_j}$ of α, β , and q propagates accordingly. Then, when the IMU bias changes slightly, the first-order approximation estimation of the bias state is adjusted based on α, β , and q , and the resulting preintegral is expressed as

$$\begin{aligned} \alpha_{b_{j+1}}^{b_j} &\approx \tilde{\alpha}_{b_{j+1}}^{b_j} + J_{b_a}^{\alpha} \delta m_{a_j} + J_{b_w}^{\alpha} \delta m_{w_j}, \\ \beta_{b_{j+1}}^{b_j} &\approx \tilde{\beta}_{b_{j+1}}^{b_j} + J_{b_a}^{\beta} \delta m_{a_j} + J_{b_w}^{\beta} \delta m_{w_j}, \\ q_{b_{j+1}}^{b_j} &\approx \tilde{q}_{b_{j+1}}^{b_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{b_w}^q \delta m_{w_j} \end{bmatrix}. \end{aligned} \quad (4)$$

The preintegration method does not need to repeat IMU measurements, and it saves a lot of computing time and resources for subsequent graph-based optimization algorithms.

2.2.3. Odometer Preintegration. The wheel odometer can measure the linear velocity $\tilde{v}_d$ and the angular velocity $\tilde{\omega}_d$ of the robot. It is assumed that the noise between the measured value and the true value is Gaussian noise, $n = [n_v, n_g]$. The measurement model is given as follows:

$$\begin{aligned}\tilde{v}_d &= v_d + n_v, \\ \tilde{\omega}_d &= \omega_d + n_g.\end{aligned}\quad (5)$$

Considering the problem of slipping off the wheeled robot, the angular velocity measured by the gyroscope can be used as the acceleration of the robot. Referring to the IMU preintegration algorithm, the preintegration is calculated for the odometer measurement and the angular velocity measured by the gyroscope; the preintegration of the velocity is simpler, and there is no offsetting effect in the acceleration. The odometer preintegration results are shown as follows:

$$\begin{aligned}p_{d_{j+1}}^w &= p_{d_j}^w + \int_{t \in [j, j+1]} (R_{d_t}^w (\tilde{v}_{d_t} - n_v)) dt, \\ q_{d_{j+1}}^w &= q_{d_j}^w \otimes \int_{t \in [j, j+1]} \frac{1}{2} \Omega(\tilde{\omega}_{d_t} - m_{\omega_t} - n_{\omega}) dt.\end{aligned}\quad (6)$$

Similar to IMU preintegral, the preintegral measurement is obtained by separating the initial value and the constant term of the integral. When the gyroscope bias changes slightly, the approximation of the previous moment is used to update $\alpha_{d_{j+1}}^j$ and $q_{d_{j+1}}^j$, instead of iteratively calculating the preintegration measurements:

$$q_{d_{j+1}}^j \approx \tilde{q}_{d_{j+1}}^j \otimes \begin{bmatrix} 1 \\ J_{b_{\omega}}^q \delta m_{\omega_j} \end{bmatrix}. \quad (7)$$

3. System Initialization State Estimation

The initial value of the system is obtained by a loosely coupled method. Firstly, the structure from motion (SFM) is used to estimate the camera pose and feature 3D position. Then, the SFM results are aligned with IMU preintegration and odometer preintegration, respectively, to correct the gyroscope bias. At the same time, the corresponding velocity of each frame, gravity vector direction, and scale factor are solved. It should be noted that the scale is the final result obtained by weighting the scale estimated by the two alignment processes according to the dynamic weight method.

3.1. Vision-Only Initialization Based on the Sliding Window. The monocular initialization uses SFM to estimate the camera scale pose and feature point spatial position of each frame within the sliding window. Firstly, the feature matching between the latest frame and any other frame in the window is checked. If a stable feature tracking and sufficient parallax are found, the scale and pose between the two frames are restored by the five-point method [24]. Then, the feature triangulation and the reprojection error are performed to be minimized. Otherwise, the current latest frame is saved, continued to wait for the new frame. The pose $(p_{c_j}^{c_0}, q_{c_j}^{c_0})$ of all frames in the sliding window is obtained by the above procedure. It is assumed that the external parameter between camera and IMU is (p_c^b, q_c^b) , the external

parameter between camera and the odometer is (p_c^d, q_c^d) . The pose is converted to the robot coordinate system and odometer coordinate system, respectively. The attitude is converted to the IMU coordinate system as

$$\begin{aligned}s_1 \bar{p}_{b_j}^{c_0} &= s_1 \bar{p}_{c_j}^{c_0} - R_{b_j}^{c_0} p_c^b, \\ q_{b_j}^{c_0} &= q_{c_j}^{c_0} \otimes (q_c^b)^{-1}.\end{aligned}\quad (8)$$

Similarly, the conversion to the odometer coordinate system is shown as

$$\begin{aligned}s_2 \bar{p}_{d_j}^{c_0} &= s_2 \bar{p}_{c_j}^{c_0} - R_{d_j}^{c_0} p_c^d, \\ q_{d_j}^{c_0} &= q_{c_j}^{c_0} \otimes (q_c^d)^{-1},\end{aligned}\quad (9)$$

where s_1 and s_2 both represent the scale information of the visual measurement and are used for the subsequent dynamic weighting to solve the final scale parameter s .

3.2. Vision and IMU Preintegral Alignment

3.2.1. Gyro Offset Calibration. Two consecutive frames b_j and b_{j+1} in the sliding window are rotated by visual SFM to obtain $q_{b_j}^{c_0}$ and $q_{b_{j+1}}^{c_0}$. The relative constraint $\tilde{q}_{b_{j+1}}^{b_j}$ is obtained by IMU preintegration. Then, the linearization of the IMU preintegral term is obtained for the gyroscope bias and the cost function is minimized as

$$\min_{\delta b_{\omega}} \sum_{k \in B} \left\| q_{b_{j+1}}^{c_0-1} \otimes q_{b_j}^{c_0} \otimes q_{b_{j+1}}^{b_j} \right\|^2, \quad (10)$$

where B represents all keyframes in the sliding window, where

$$q_{b_{j+1}}^{b_j} = \tilde{q}_{b_{j+1}}^{b_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2} J_{b_{\omega}}^{b_j} \delta b_{\omega_j} \end{bmatrix}. \quad (11)$$

After the initial calibration value of the gyroscope bias m_{ω} is obtained, the IMU preintegration is recalculated based on the new gyroscope bias.

3.2.2. Velocity, Gravity Vector, and Scale Initialization. After the gyro bias is initialized, the gyro measurement is regarded as accurate and continues to be used to initialize other variables to be optimized:

$$\chi_I = \left[v_{b_0}^{b_0}, v_{b_1}^{b_1}, \dots, v_{b_n}^{b_n}, g^{c_0}, s_1 \right]^T, \quad (12)$$

where $v_{b_j}^{b_j}$ is the velocity of the j frame image robot coordinate system, g^{c_0} is the gravity vector in the c_0 coordinate system, and s_1 is based on the monocular scale obtained by the IMU preintegration.

The world coordinate system w in the preintegral terms $\alpha_{b_{j+1}}^{b_j}$ and $\beta_{b_{j+1}}^{b_j}$ is replaced with the c_0 coordinate system and converted into the form of $Hx = b$:

$$H_{b_{j+1}}^{b_j} \chi_I = \begin{bmatrix} \tilde{\alpha}_{b_{j+1}}^{b_{kj}} - p_c^b + R_{c_0}^{b_j} R_{b_{j+1}}^{c_0} p_c^b \\ \tilde{\beta}_{b_{j+1}}^{b_j} \end{bmatrix} \approx \tilde{z}_{b_{j+1}}^{b_j}, \quad (13)$$

where

$$H_{b_{j+1}}^{b_j} = \begin{bmatrix} -I\Delta t_j & 0 & \frac{1}{2}R_{c_0}^{b_j}\Delta t_j^2 & R_{c_0}^{b_j} \\ -I & R_{c_0}^{b_j}R_{b_{j+1}}^{c_0} & R_{c_0}^{b_j}\Delta t_j & 0 \end{bmatrix}, \quad (14)$$

$$p_c^b = \bar{p}_{c_{j+1}}^{c_0} - \bar{p}_{c_j}^{c_0}.$$

By solving the linear least-squares problem,

$$\min_{\chi_I} \sum_{j \in B} \left\| \tilde{z}_{b_{j+1}}^{b_j} - H_{b_{j+1}}^{b_j} \chi_I \right\|^2. \quad (15)$$

According to the above formulas, χ_I can be optimized, including all keyframe velocities, gravity vectors, and monocular scales in sliding windows.

3.3. Vision and Odometer Preintegral Alignment. Since the gyro measurement is used for the rotation measurement of the odometer preintegration, the gyro offset after calibration can be used directly. The variables to be optimized are defined as

$$\chi_{II} = \left[v_{d_0}^{d_0}, v_{d_1}^{d_1}, \dots, v_{d_m}^{d_m}, s_2 \right]^T. \quad (16)$$

Similarly, the odometer preconcentration term is transferred to the c_0 coordinate system and converted into the form of $Hx = b$:

$$H_{d_{j+1}}^{d_j} \chi_{II} = \tilde{\alpha}_{d_{j+1}}^{d_{kj}} - p_c^d + R_{c_0}^{d_j} R_{d_{j+1}}^{c_0} p_c^d \approx \tilde{z}_{d_{j+1}}^{d_j}, \quad (17)$$

$$H_{d_{j+1}}^{d_j} = \begin{bmatrix} -I\Delta t_j & 0 & R_{c_0}^{d_j} (\bar{p}_{c_{j+1}}^{c_0} - \bar{p}_{c_j}^{c_0}) \end{bmatrix}.$$

By solving the linear least-squares problem,

$$\min_{\chi_{II}} \sum_{j \in B} \left\| \tilde{z}_{d_{j+1}}^{d_j} - H_{d_{j+1}}^{d_j} \chi_{II} \right\|^2. \quad (18)$$

The coordinate system speed of odometer in each frame in the window and the parameters of the monocular scale can be obtained.

3.4. Dynamic Weighted Scale Parameter. The idea of dynamic weighting method is used to design a sliding window with a size of 5 keyframes based on the original sliding window of the system and reoperate the sliding window of the system with a size of 10 keyframes. Therefore, the keyframe $B_{i \in \{1,2,3,4,5\}}$ is selected from the sliding window, where 1–5 frames within the window are included in B_1 , 2–6 frames within the window are included in B_2 , and so on, and B_5 is derived. The scale parameters $s_{1,i}$ and $s_{2,i}$ at B_i are calculated according to Sections 3.2 and 3.3, respectively, and the minimum mean square errors $e_{1,i}$ and $e_{2,i}$ are obtained. The scale parameter $s = W_1 s_1 + W_2 s_2$ is calculated by dynamically adjusting the weights W_1 and W_2 according to the above parameters, wherein $\sum_{n=1}^2 W_n = 1$. The initial values of W_1 and W_2 are both 0.5 as shown in Figure 3.

4. Back-End Tight Coupling Optimization

4.1. State Variables and Cost Functions. To improve computational efficiency, a sliding window is used to achieve graph optimization in this paper, so the state variables defining the sliding window optimization at time j are shown as

$$\chi = [x_0, x_1, \dots, x_n, \lambda_0, \lambda_1, \dots, \lambda_k], \quad (19)$$

$$x_i = [p_{b_i}^w, v_i^w, q_{b_i}^w, m_a, m_w], \quad i \in [0, n],$$

where x_i represents the position, velocity, the orientation of the robot in the global coordinate system, and the bias of the IMU. λ_k represents the inverse depth of the k th feature point. n and k represent the number of keyframes and the total number of 3D feature points, respectively.

In the sliding window, the residual of all measurement models is minimized to optimize the state variables:

$$\min_{\chi} \left\{ \rho \|r_p - J_p \chi\|^2 + \sum_{i \in B} \rho \|r_B(\tilde{z}_{b_{i+1}}^{b_i}, \chi)\|_{P_{b_{i+1}}^{b_i}}^2 + \sum_{i \in D} \rho \|r_D(\tilde{z}_{d_{i+1}}^{d_i}, \chi)\|_{P_{d_{i+1}}^{d_i}}^2 + \sum_{(i,j) \in C} \rho \|r_C(\tilde{z}_i^j, \chi)\|_{P_i^{c_j}}^2 \right\}, \quad (20)$$

where $r_B(\tilde{z}_{b_{i+1}}^{b_i}, \chi)$ and $r_D(\tilde{z}_{d_{i+1}}^{d_i}, \chi)$ are the residual measurements of the IMU and odometer between state x_i and x_{i+1} , and B and D are the preintegral measurements of the IMU and odometer; $r_C(\tilde{z}_i^j, \chi)$ is the 3D feature point reprojection error; r_p is the prior information in the window; J_p is the Jacobian matrix obtained from the last optimization result; ρ is the Huber kernel function to make the optimization more robust.

4.2. Camera Measurement Residual. In this paper, a pin-hole model is used to calculate the reprojection error. First, the reprojection error $r_C(\tilde{z}_i^j, \chi)$ is defined according to the observation value $z_i^j = [u_i^j, v_i^j, 1]^T$ of the k th feature point in the image frame c_j in the normalized camera coordinate system:

FIGURE 3: Dynamic weighting flowchart.

$$r_C(\tilde{z}_i^c, \chi) = \begin{bmatrix} \frac{x^{c_j}}{z^{c_j}} - u_i^{c_j} \\ \frac{y^{c_j}}{z^{c_j}} - v_i^{c_j} \end{bmatrix}, \quad (21) \quad \text{where}$$

$$\begin{bmatrix} u^{c_j} \\ v^{c_j} \\ z^{c_j} \end{bmatrix} = \begin{pmatrix} R_b^c \left(R_w^{b_j} \left(R_{b_j}^w \left(R_b^c \frac{1}{\lambda_k} \begin{bmatrix} u^{c_i} \\ v^{c_i} \\ 1 \end{bmatrix} + p_c^b \right) + p_{b_i}^w \right) - p_{b_j}^w \right) - p_c^b \end{pmatrix}. \quad (22)$$

The above formula is a three-dimensional coordinate form of the reprojection equation, and the optimization variable is based on the estimated value and the reprojection error in the normalized camera coordinate system, including the state quantities $[p_{b_i}^w, q_{b_i}^w]$ and $[p_{b_j}^w, q_{b_j}^w]$ and the inverse depth λ_i at two moments. To minimize the reprojection error, it is necessary to optimize the above state quantities,

and the corresponding Jacobian matrix is obtained as shown in equations (23)–(25).

4.3. IMU Measurement Residual. According to the IMU measurement in the two consecutive frames b_i and b_{i+1} and the content in formula (4), the residual of the IMU is obtained as follows:

$$r_B(\tilde{z}_{b_{i+1}}^{b_i}, \chi) = \begin{bmatrix} \delta \alpha_{b_{i+1}}^{b_i} \\ \delta \beta_{b_{i+1}}^{b_i} \\ \delta \theta_{b_{i+1}}^{b_i} \\ \delta m_a \\ \delta m_\omega \end{bmatrix} = \begin{bmatrix} R_w^{b_i} \left(p_{b_{i+1}}^w - p_{b_i}^w + \frac{1}{2} g^w \Delta t_i^2 - v_{b_i}^w \Delta t_i \right) - \tilde{\alpha}_{b_{i+1}}^{b_i} \\ R_w^{b_i} (v_{b_{i+1}}^w + g^w \Delta t_i - v_{b_i}^w) - \tilde{\beta}_{b_{i+1}}^{b_i} \\ 2 \left[(q_{b_i}^w)^{-1} \otimes q_{b_{i+1}}^w \otimes (\tilde{q}_{b_{i+1}}^{b_i})^{-1} \right]_{xyz} \\ m_{ab_{i+1}} - m_{ab_i} \\ m_{wb_{i+1}} - m_{wb_i} \end{bmatrix}. \quad (23)$$

Here, $\tilde{\alpha}_{b_{i+1}}^{b_i}$, $\tilde{\beta}_{b_{i+1}}^{b_i}$, and $\tilde{q}_{b_{i+1}}^{b_i}$ are the measurements obtained after the corrected bias. $\delta \theta_{b_{i+1}}^{b_i}$ is a three-dimensional error state representation of the quaternion, and $[q]_{xyz}$ represents the imaginary part of the extracted quaternion q . In the graph optimization, the state quantities x_i and x_{i+1} need to be optimized, and the corresponding Jacobian matrix is obtained as shown in equations (27)–(29).

4.4. Odometer Measurement Residual. It is assumed that the axle is on the central axis of the robot. The amount of displacement measured by the odometer can be used to

obtain the displacement of the center of the robot. Therefore, the odometer measurement residual is defined as

$$r_D(\tilde{z}_{d_{i+1}}^{d_i}, \chi) = \begin{bmatrix} \Delta R_{d_{i+1}}^{d_i} \\ \Delta p_{d_{i+1}}^{d_i} \end{bmatrix} = \begin{bmatrix} R_w^{d_i} R_{d_{i+1}}^w \text{Exp}(\delta \theta_{d_{i+1}}^{d_i}) \\ -R_w^{d_i} R_{d_j}^w p_w^{d_{i+1}} + p_w^{d_i} + \delta p_{d_{i+1}}^{d_i} \end{bmatrix}. \quad (24)$$

In the graph optimization model, we need to optimize the states x_i and x_{i+1} under two consecutive frames b_i and b_{i+1} and obtain the corresponding Jacobian matrices such as formulas (31) and (32):

$$J_C[0] = \begin{bmatrix} \frac{\partial r_C}{\partial p_{b_i}^w} & \frac{\partial r_C}{\partial q_{b_i}^w} \end{bmatrix} = \begin{bmatrix} R_b^c R_w^{b_j} & -R_b^c R_w^{b_j} R_{b_i}^w \left[R_b^c \frac{1}{\lambda_i} \tilde{p}^{c_i} + p_c^b \right]_{\times} \end{bmatrix}, \quad (25)$$

$$J_C[1] = \begin{bmatrix} \frac{\partial r_C}{\partial p_{b_j}^w} & \frac{\partial r_C}{\partial q_{b_j}^w} \end{bmatrix} = \begin{bmatrix} -R_b^c R_w^{b_j} & R_b^c \left[R_w^{b_j} \left(R_{b_i}^w \left(R_b^c \frac{1}{\lambda_i} \tilde{p}^{c_i} + p_c^b \right) + p_{b_i}^w - p_{b_j}^w \right) \right]_{\times} \end{bmatrix}, \quad (26)$$

$$J_C[2] = \frac{\partial r_C}{\partial \lambda_i} = -R_b^c R_w^{b_j} R_{b_i}^w R_c^b \frac{\tilde{p}^{c_i}}{\lambda_i^2}, \quad (27)$$

$$J_B[0] = \begin{bmatrix} -R_w^{b_i} & [R_w^{b_i} \Delta p]_{\times} & -R_w^{b_i} \Delta t_i & -J_{b_a}^{\alpha} & -J_{b_w}^{\alpha} \\ 0 & -\left[L \left((q_{b_{i+1}}^w)^{-1} \otimes q_{b_i}^w \right) R \left(\tilde{q}_{b_{i+1}}^{b_i} \right) \right] & 0 & 0 & -\left[L \left((q_{b_{i+1}}^w)^{-1} \otimes q_{b_i}^w \otimes \tilde{q}_{b_{i+1}}^{b_i} \right) \right] J_{b_w}^{q_{i+1}^i} \\ 0 & \left[R_w^{b_i} (v_{b_{i+1}}^w - v_{b_i}^w + g^w \Delta t_i) \right]_{\times} & -R_w^{b_i} & -J_{b_a}^{\beta} & -J_{b_w}^{\beta} \\ 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & 0 & -I \end{bmatrix}, \quad (28)$$

$$J_B[1] = \begin{bmatrix} R_w^{b_i} & 0 & 0 & 0 & 0 \\ 0 & \left[L \left(\left(\tilde{q}_{b_{i+1}}^{b_i} \right)^{-1} \otimes (q_{b_i}^w)^{-1} \right) q_{b_{i+1}}^w \right] & 0 & 0 & 0 \\ 0 & 0 & R_w^{b_i} & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix}, \quad (29)$$

where

$$q = [w, x, y, z] = [w, s],$$

$$\Delta p = p_{b_{i+1}}^w - p_{b_i}^w - v_{b_i}^w \Delta t_i + \frac{1}{2} g^w \Delta t_i^2, \quad (30)$$

$$L(q) = \begin{bmatrix} [w]_{\times} & w \\ -w^T & 0 \end{bmatrix} + sI, R(q) = \begin{bmatrix} -[w]_{\times} & w \\ -w^T & 0 \end{bmatrix} + sI,$$

$$J_D[0] = \frac{\partial r_D}{\partial \delta x_i} = \begin{bmatrix} 0 & \left((q_{b_{i+1}}^w)^{-1} \otimes q_{b_i}^w \right) \text{Exp}(\delta \theta_{b_i}^{b_{i+1}}) & 0 & 0 & 0 \\ I & 0 & v_i^w & 0 & 0 \end{bmatrix}, \quad (31)$$

$$J_D[1] = \frac{\partial r_D}{\partial \delta x_{i+1}} = \begin{bmatrix} 0 & \left(\left(\tilde{q}_{b_{i+1}}^{b_i} \right)^{-1} \otimes (q_{b_i}^w)^{-1} \right) \text{Exp}(\delta \theta_{b_i}^{b_{i+1}}) & 0 & 0 & 0 \\ -R_{b_i}^w R_w^{b_{i+1}} & 0 & -R_{b_i}^w R_w^{b_{i+1}} v_{i+1}^w & 0 & 0 \end{bmatrix}. \quad (32)$$

5. Experimental Result Analysis

The experimental scene in this paper is the indoor environment with a Vicon motion capture system, and the robot

platform is Turtlebot3 mobile robot with a camera named Xiaomi. The robot moves in the area of $3m \times 3m$, and the average speed of the whole process is $0.2m/s$. In the experiment, the position and posture of the robot acquired

by the motion capture system in the motion capture coordinate system are taken as the true value of the motion trajectory.

5.1. Scale Error Analysis. The experiment mainly verifies the effects of the initialization state estimation algorithm and the dynamic weighting result proposed in Section 3.4. Compared with the initialization using only IMU, in this paper, the combination of visual, IMU, and odometer can achieve more accurate scale and convergence time. The scale estimation results and time are shown in Table 1, and the root mean square error (RMSE) is used to quantitatively represent the accuracy of the algorithm.

It can be seen from the results in Table 1 that the initialization state estimation algorithm of this paper reduces the RMSE of the scale value by 72.2% and the convergence time RMSE by 62.9% compared with the amplification using only the IMU for initialization. Therefore, the initialization state estimation algorithm proposed has obvious improvement in the accuracy and stability of scale estimation.

5.2. Pose Error Analysis. The algorithm proposed in this paper is an improvement based on VINS_MONO, so the experimental results are compared with the VINS_MONO. The trajectory of mono_imu in the figure represents the result of the VINS_MONO [25] algorithm running on the robot platform. The trajectory and true value comparison of the two algorithms are shown in Figures 4 and 5. The dotted line in the figure is the real trajectory obtained by the motion capture system, and the solid line is the trajectory obtained by the algorithm. As can be seen from the figure, due to the odometer constraint and more accurate initialization scale added to the algorithm in this paper, the accuracy in the positioning process of mobile robots is better than that of VINS_MONO.

To quantitatively analyze the accuracy of the algorithm, the accuracy is measured by estimating the absolute position error (APE) obtained by comparing the pose result with the true motion pose of each time step. The pose and APE and the rotational APE of the algorithm proposed in this paper and the monocular inertial navigation algorithm are shown in Figure 6, respectively. The overall pose APE obtained by the two algorithms is shown in Table 2.

By comparing the two methods, it is not difficult to see that the proposed algorithm is superior to the monocular inertial fusion algorithm in accuracy, which indicates that the odometer information has a certain degree of improvement on the positioning result. Moreover, the better planar constraint of the odometer makes the algorithm consistently stable at 0.03 m in the z^w direction, while the monocular inertial fusion algorithm has a certain degree of trajectory divergence in the z^w direction. Since the center of gravity of the rigid body model built in the Vicon system is 0.03 m, the true value keeps near 0.03 m in the z^w direction. The results are shown in Figure 7.

TABLE 1: Scale values and convergence time.

Algorithm Number	Monocular + IMU		Proposed algorithm	
	Scale (m)	Time (s)	Scale (m)	Time (s)
1	0.07466	2.67605	0.03456	2.31003
2	0.05031	2.31583	0.02535	1.85799
3	0.07570	2.41201	0.02435	2.09317
4	0.03262	2.21847	0.01493	1.68178
5	0.03084	2.34407	0.01459	2.00447
6	0.08114	2.75072	0.01258	1.78724
7	0.01052	1.35351	0.01606	2.00229
8	0.02975	1.55614	0.02438	1.99803
9	0.03340	2.32441	0.01275	2.08879
10	0.07908	2.79579	0.02233	1.89776
Mean	0.04981	2.27470	0.02019	1.97215
RMSE	0.02448	0.45333	0.00679	0.16796

FIGURE 4: Comparison of trajectories and truth values of the two algorithms.

FIGURE 5: Comparison of x -axis and y -axis with true values.

FIGURE 6: Pose APE comparison of the two methods.

TABLE 2: APE statistics results of two methods.

Index	Proposed algorithm	Monocular + IMU
Max pose APE (m)	0.075407	0.311616
Min pose APE (m)	0.002026	0.046728
Mean pose APE (m)	0.024533	0.154703
Pose RMSE (m)	0.027296	0.169879

FIGURE 7: Comparison of z-axis with true values.

6. Conclusions

In this paper, aiming at the problem of low accuracy and robustness of the monocular inertial navigation algorithm in the pose estimation of mobile robots, a multisensor fusion positioning system is designed, including monocular vision, IMU, and odometer, which realizes the initial state estimation of monocular vision and the fusion location of the tightly coupled visual-inertial odometer.

The method of odometer preintegration and dynamic weighted scale parameters, as well as the back-end tightly coupled pose optimization method based on graph optimization, is proposed, which greatly improves the accuracy

and robustness of the positioning system. Then, the effectiveness and feasibility of the proposed algorithm is verified by theoretical derivation and practical experiments.

Although the research in this paper has achieved certain results, there are still some aspects that need to be improved. First, the results are only tested in the laboratory or a small number of complex scenarios, and it cannot fully explain the ability to work robustly and stably for a long time in a complex environment. Second, binocular cameras are superior to monocular cameras in terms of depth calculation and scale estimation. In subsequent solutions, multisensor fusion schemes can be considered based on binocular vision and contribute to future map-based visual navigation studies.

Data Availability

The data are available at <https://figshare.com/s/c0328d437eb86a021e24>, DOI: 10.6084/m9.figshare.9782474.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the NSFC (National Nature Science Foundation of China) under grants nos. 61374128 and 61973278.

References

- [1] H. Kim, D. Kim, and K. Sim, "Simultaneous localization and map building using vision camera and electrical compass," in *Proceedings of the SICE-ICASE International Conference*, IEEE, Busan, Korea, pp. 5915–5918, October 2006.
- [2] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 611–625, 2018.
- [3] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [4] S. T. Wang, *Principle and Design of the Mobile Robot*, People's Daily House, Beijing, China, 2013.
- [5] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE, Nara, Japan, pp. 225–234, November 2007.
- [6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [7] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," *Computer Vision*, vol. 8690, pp. 834–849, 2014.
- [8] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: fast semi-direct monocular visual odometry," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Hong Kong, pp. 15–22, June 2014.
- [9] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.
- [10] C. Kerl, J. Sturm, and D. Creamers, "Dense visual slam for rgbd cameras," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, IEEE, Tokyo, Japan, pp. 2011–2016, November 2013.
- [11] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, IEEE, Roma, Italy, pp. 3565–3572, April 2007.
- [12] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
- [13] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proceedings of the International Conference on Intelligent Robots and Systems*, IEEE, Hamburg, Germany, pp. 298–304, September 2015.
- [14] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [15] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen, "Monocular visual-inertial state estimation for mobile augmented reality," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, IEEE, Nantes, France, pp. 11–21, October 2017.
- [16] T. Qin, P. Li, and S. Shen, "Relocalization, global optimization and map merging for monocular visual-inertial SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, Brisbane, Australia, pp. 1197–1204, May 2018.
- [17] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Visual SLAM: why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.
- [18] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "ICE-BA: incremental, consistent and efficient bundle adjustment for visual-inertial SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, pp. 1974–1982, June 2018.
- [19] Y. Lin, F. Gao, T. Qin et al., "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 4, 2018.
- [20] Z. Yang, F. Gao, and S. Shen, "Real-time monocular dense mapping on aerial robots using visual-inertial fusion," in *Proceedings of the IEEE International Conference on Automation Science & Engineering*, vol. 14, no. 1, pp. 39–51, IEEE, Fort Worth, TX, USA, August 2016.
- [21] D. G.-L. Lopez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [22] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, IEEE, Vancouver, Canada, pp. 24–28, August 1981.
- [23] J. Shi and C. Tomasi, "Good features to track," in computer vision and pattern recognition," in *Proceedings of the CVPR'94 IEEE Computer Society Conference*, IEEE, Seattle, WA, USA, pp. 593–600, June 1994.
- [24] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.
- [25] 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.