

Article

Visual-Inertial RGB-D SLAM with Encoder Integration of ORB Triangulation and Depth Measurement Uncertainties

Zhan-Wu Ma and Wan-Sheng Cheng *

School of Electronic and Information Engineering, University of Science and Technology Liaoning,
Anshan 114051, China

* Correspondence: cws@ustl.edu.cn

Abstract: In recent years, the accuracy of visual SLAM (Simultaneous Localization and Mapping) technology has seen significant improvements, making it a prominent area of research. However, within the current RGB-D SLAM systems, the estimation of 3D positions of feature points primarily relies on direct measurements from RGB-D depth cameras, which inherently contain measurement errors. Moreover, the potential of triangulation-based estimation for ORB (Oriented FAST and Rotated BRIEF) feature points remains underutilized. To address the singularity of measurement data, this paper proposes the integration of the ORB features, triangulation uncertainty estimation and depth measurements uncertainty estimation, for 3D positions of feature points. This integration is achieved using a CI (Covariance Intersection) filter, referred to as the CI-TEDM (Triangulation Estimates and Depth Measurements) method. Vision-based SLAM systems face significant challenges, particularly in environments, such as long straight corridors, weakly textured scenes, or during rapid motion, where tracking failures are common. To enhance the stability of visual SLAM, this paper introduces an improved CI-TEDM method by incorporating wheel encoder data. The mathematical model of the encoder is proposed, and detailed derivations of the encoder pre-integration model and error model are provided. Building on these improvements, we propose a novel tightly coupled visual-inertial RGB-D SLAM with encoder integration of ORB triangulation and depth measurement uncertainties. Validation on open-source datasets and real-world environments demonstrates that the proposed improvements significantly enhance the robustness of real-time state estimation and localization accuracy for intelligent vehicles in challenging environments.



Citation: Ma, Z.-W.; Cheng, W.-S. Visual-Inertial RGB-D SLAM with Encoder Integration of ORB Triangulation and Depth Measurement Uncertainties. *Sensors* **2024**, *24*, 5964. <https://doi.org/10.3390/s24185964>

Academic Editors: Chunhui Zhao and Shuai Hao

Received: 6 August 2024

Revised: 5 September 2024

Accepted: 12 September 2024

Published: 14 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: covariance intersection filter; encoders; RGB-D SLAM; multisensor fusion

1. Introduction

In recent years, with the rapid development of robotics, computer vision, autonomous driving, augmented reality/virtual reality, planetary exploration, and unmanned aerial vehicle navigation [1], visual odometry and SLAM have emerged as prominent areas of research [2]. SLAM technology enables robots to not only achieve real-time self-localization but also to construct and continuously update maps of unknown environments during exploration, significantly enhancing their ability to operate autonomously in such settings [3]. In the field of robotics, intelligent wheeled robots have emerged as a prominent area of research [4]. Based on the types of sensors employed, mainstream SLAM systems are primarily classified into laser-based SLAM, vision-based SLAM, and various sensor-assisted laser/vision multisensor fusion SLAM technologies. Visual SLAM relies on cameras as the primary sensors, and its ability to capture rich information, combined with its lightweight and cost-effective nature, has garnered significant interest from researchers [5]. Within the domain of visual SLAM, ORB-SLAM3 demonstrates superior performance, surpassing other algorithms in terms of accuracy, computational efficiency, and robustness [6]. Currently, ORB-SLAM3 acquires the 3D positions of feature points by directly utilizing RGB-D depth camera measurements, without accounting for the uncertainty in triangulation estimation. Furthermore, ORB-SLAM3 fuses data from the RGB-D depth camera

and the IMU, but it does not integrate wheel encoder data, which are crucial for improving SLAM accuracy and aiding mobile robot localization and navigation. The literature [7] proposes a lightweight multisensor fusion method involving ORB-SLAM, IMU, and wheel odometry for the localization and navigation of an indoor mobile robot in a GPS-denied environment. Experimental results demonstrate that the robot can localize itself with tolerable error and exhibit strong navigation capabilities in specific scenarios. The ORB-SLAM in this study employs a monocular camera to compute the real-time camera position through feature matching. To filter out dynamic objects during localization and mapping, the literature [8] presents a mobile robot localization method using asynchronous data fusion comprising a wheel odometer and a visual odometer within an Extended Kalman Filter (EKF) framework, based on a semantic SLAM system. The approach integrates semantic SLAM based on ORB-SLAM2 with YOLOv3. The asynchronous data fusion for mobile robot localization, comprising visual and wheel odometer data, is implemented at a frequency of approximately 50 Hz using EKF. Experimental results demonstrate that this method effectively improves the accuracy and robustness of the mobile robot. Recently, the literature [9] proposed a visual-inertial-wheel odometry method that provides robust initialization and highly accurate estimates for ground robots. This study utilizes a novel maximum-a-posteriori initialization, coupled with wheel encoder measurements, to address the unobservable scale problem in visual-inertial-only initialization during straight-line motion at the beginning of the trajectory. Experiments on public datasets demonstrate the effectiveness and efficiency of the initialization, the robustness of pose tracking, and the improved accuracy of the entire trajectory. In addition, the literature [10] utilizes a CI filter to fuse inertial navigation sensors and wheel odometry to improve the accuracy of real-time localization and mapping. The effectiveness and feasibility of the algorithm are validated using an experimental platform. The aforementioned literature demonstrates that the use of wheel encoders not only enhances the accuracy and robustness of SLAM but also addresses the initialization failures in mobile robots. Inspired by this literature, this paper proposes a novel tightly coupled visual-inertial RGB-D SLAM with encoder integration of ORB triangulation and depth measurement uncertainties to improve ORB-SLAM3.

Visual odometry (VO) algorithms can be categorized into two main types: feature-based methods and direct methods [11]. However, in both feature-based and direct methods within SLAM, accurately estimating the uncertainty of the camera pose and the 3D positions of feature points is crucial. Such accurate estimations are essential for selecting precise keyframes, particularly in the context of information fusion and active SLAM [12,13]. The primary objective is to minimize uncertainty or entropy, which necessitates a closed-form solution for the uncertainty estimation of the camera pose and the 3D positions of feature points. The uncertainty estimation of the camera pose [1] and the 3D positions of feature points are key parameters for SLAM. This uncertainty can generally be expressed using the covariance matrix [14], information entropy [15], and Fisher's information matrix [13]. The estimations of uncertainty in the camera poses and the 3D positions of feature points are primarily based on filters (e.g., the Extended Kalman Filter) [16] and nonlinear optimization methods (e.g., Bundle Adjustment and graph optimization) [17]. Uncertainty estimations based on filter methods are both convenient and fast, providing direct access to the covariance matrix or expected entropy. In contrast, nonlinear optimization-based methods do not directly estimate uncertainty. As a result, the covariance matrix and information entropy are not optimized as direct parameters. Instead, uncertainty in nonlinear optimization requires rigorous computation [18].

In the study of Vakhitov et al. [19], a PnP(L) solver based on EPnP and DLS is proposed for uncertainty-aware pose estimation by considering both the 3D coordinates and 2D projections of feature points in SLAM. Additionally, the motion-only bundle adjustment is modified to account for the uncertainty in the 3D positions of feature points. Tests performed on the KITTI datasets demonstrate improved accuracy. In Belter et al. (2016) [20], two spatial uncertainty models are proposed, based on experimental tests, to estimate the covariance matrix of the measured feature points. The resulting covariance is then

incorporated into factor graph optimization in the back-end of the SLAM system. While these methods provide error models and uncertainties for 3D points, they cannot be directly applied to more general visual SLAM systems. In Belter et al. (2018) [21], the problem of estimating the uncertainty in the 3D positions of feature points using the feature method is discussed in detail. The effect of point feature uncertainty on trajectory and map uncertainty estimation is investigated through an uncertainty model. A factor graph optimization model is employed to minimize the error in the 3D positions of feature points measured by depth cameras. Experiments conducted on open-source datasets demonstrate that the algorithm can improve the accuracy of camera pose estimation.

Currently, researchers mainly focus on one type of uncertainty study, such as the 3D positions of feature points [22–24] or camera poses [25–28]. Li and Yang [29] present a pose fusion method that accounts for the possible correlations among measurements. The handling of these correlations is based on the theory of Covariance Intersection (CI), where the independent and dependent parts are separated to yield a more consistent result. Few studies have investigated the problem of joint uncertainty estimation for the 3D positions of feature points and camera pose. However, the above methods are unable to provide closed-form solutions for uncertainty estimations, which are crucial for information fusion and active SLAM, as motion prediction and pose evaluation require one or even multiple steps of forward-looking uncertainty propagation. To address the above problems, this paper proposes a closed-form uncertainty estimation algorithm, called CI-TEDM (Covariance Intersection for Triangulation Estimates and Depth Measurements), for fusing uncertainties in triangulation estimates and depth measurements of the 3D positions of feature points. This algorithm is based on the closed-form uncertainty estimation of both camera poses and 3D positions of feature points. On one hand, based on the ORB-SLAM3 system, the covariance matrix of the camera pose and the triangulation of ORB features is estimated separately. On the other hand, the covariance matrix of the 3D positions of feature points obtained from the depth measurements of the RGB-D camera is computed. The 3D positions of feature points estimated by the two approaches are then fused with a Covariance Intersection (CI) filter, which uses the two covariance matrices as weights to achieve optimal fusion [18].

Moreover, the initialization of monocular SLAM can be time-consuming when the system lacks substantial disparity. Binocular SLAM requires more time to process stereo pairs [30]. Therefore, in this paper, RGB-D SLAM is used to achieve robustness and real-time performance. The open-source ORB-SLAM3 system with monocular, stereo, and RGB-D cameras has attracted much attention from scholars for its high accuracy and real-time performance [6]. Inspired by the IMU pre-integration model and the powerful and fast loop closure detection in [31], this paper introduces wheel encoder edges, which optimize the state of each frame to avoid tracking failures in RGB-D cameras and help improve the robustness of the entire RGB-D SLAM system [32]. Although RGB-D cameras can achieve centimeter-level or even higher accuracy, vision SLAM still faces significant challenges in complex environments, such as changes in illumination, moving objects, adverse weather, weak textures, and environmental degradation [33]. Cameras are prone to losing tracking information, which can lead to tracking failures due to fast camera motion, unstable performance of embedded boards, and low frame rates during image capture [34]. In this study, auxiliary sensors, such as IMU and encoders, are integrated to mitigate the problem of temporary tracking failures that may occur in pure-vision SLAM. Visual-inertial SLAM is a current research hotspot and can be categorized into two types: loosely coupled and tightly coupled [35]. Tightly coupled SLAM systems include direct methods [36], keyframe-based visual-inertial SLAM (OKVIS) [37], robust visual-inertial odometry based on EKF [38], and the monocular visual-inertial system (VINS-Mono) [39]. Among these methods, the most accurate results for the EuRoC datasets [40] are achieved by visual-inertial ORB-SLAM3, which is primarily based on the theoretical framework of real-time IMU pre-integration [6]. In this study, encoder and IMU data are fully utilized to enhance the robustness and real-time performance of RGB-D ORB-SLAM3.

The main contribution of this paper lies in the new combination of CI-TEDM and wheel odometry, which seeks to improve the accuracy and robustness of SLAM. Additionally, this paper derives the wheel encoder pre-integration model, error model, rotation-to-state-variable Jacobian matrix, and position-to-state-variable Jacobian matrix, which are used in local BA optimization to enhance SLAM results.

The rest of this paper is organized as follows. Section 2 outlines the framework of the system. Section 3 describes the fusion of ORB-SLAM3 triangulation estimation and depth measurement estimation algorithms. Section 4 derives the encoder pre-integration model, which provides the Jacobian matrix of the rotation-to-state variable and the position-to-state variable to facilitate local BA optimization for improved SLAM results. Section 5 validates the algorithms using open-source datasets and real-world environments, presenting the experimental results. Section 6 concludes the paper.

2. System Overview

The system framework diagram is shown in Figure 1, which consists of the input, function and output of the three main modules. The primary enhancement in the input module is the addition of a new encoder data thread. The system inputs include ORB images and RGB-D images from the RGB-D camera, encoder data from the wheel encoder, and IMU data from the IMU. In the function module, ORB feature points are first extracted and matched from each RGB image. The uncertainty in the camera pose is then derived using these matched point pairs. Given the observation noise of feature points in the RGB image plane, the uncertainty in the 6D camera pose ξ , represented by an element of the Lie algebra $se(3)$, is estimated through implicit differentiation and covariance propagation. Utilizing the resulting camera pose uncertainty, the uncertainty in the triangulation of ORB feature points is computed, leading to the estimation of the uncertainty in the position of each 3D point. Simultaneously, the uncertainty in the depth measurements from the RGB-D camera is propagated to determine the uncertainty in the position of each 3D point. The uncertainties from both triangulation and depth measurements are then fused using a CI filter. Additionally, it is required to synchronize the encoder and IMU data with image frames within a permissible time alignment error. Encoder or IMU data that exceed the time alignment threshold relative to the nearest keyframe are discarded. The encoder and IMU data are then pre-integrated to compute the wheel and inertial odometry between adjacent image frames. The system with CI-TEDM that fuses IMU and encoder data based on ORB-SLAM3 is referred to as VEOS3-TEDM, the system fusing only encoder data is termed VEOS3-TEDM, and the system using CI-TEDM alone is called VOS3-TEDM. An important modification is the addition of pure encoder edges in VEOS3-TEDM, which connect two states, x_i and x_j , using only encoder measurements and are incorporated into the spanning tree. This tree primarily propagates corrected poses to all keyframes in the map. Finally, the CI filter fusion results and odometry results are input into the tracking, local mapping, loop closure, map merging, and full bundle adjustment (BA) modules for processing. In the output module, the map points of the environment and the keyframes of the mobile robot's trajectory are obtained through the aforementioned processing.

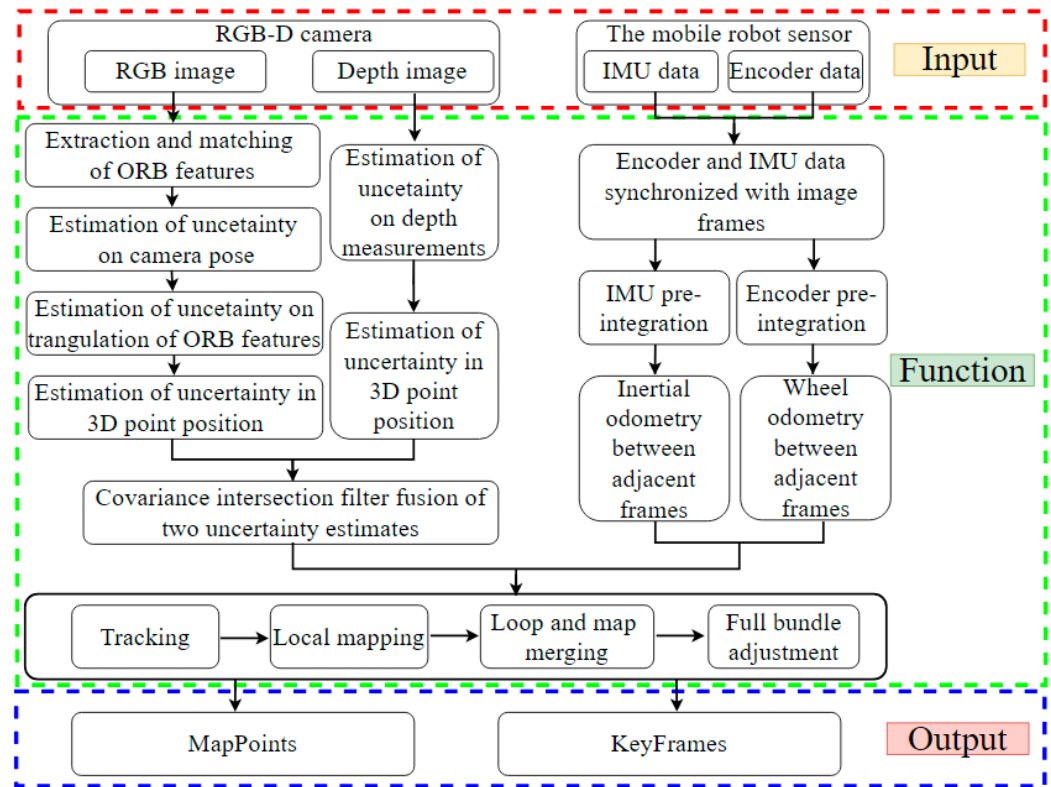


Figure 1. The system framework diagram. The system framework diagram consists of three main modules: input, function, and output.

3. Fusion of ORB-SLAM3 Triangulation and Depth Measurement Uncertainty Estimations

3.1. Uncertainty Estimation in ORB-SLAM3 Triangulation and Depth Measurement

The first step is to estimate the uncertainty of the camera pose. Minimizing the re-projection error of 3D feature points is commonly used to estimate the camera pose, as illustrated in Figure 2. Once the uncertainty of the camera pose is determined, the uncertainty of the 3D feature points can be calculated. Simultaneously, the depth measurement uncertainty is also estimated. The derivation of the solution formula mentioned above can be found in Yuan et al. [18].

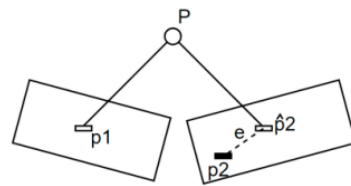


Figure 2. An example diagram of reprojection error. Feature matching indicates that points $p1$ and $p2$ are projections of the same spatial point p , but the camera pose is initially unknown. Initially, there is a certain distance between the projected point, $\hat{p}2$, of P and the actual point, $p2$. The camera pose is then adjusted to minimize this distance.

3.2. Fusion of Two Uncertainty Estimations

It is assumed that the two uncertainty estimates of the 3D feature points obtained using the camera follow a normal distribution, as denoted in Yuan et al. [18].

$$N\left(\hat{p}_{i,Tr}^w, \text{cov}\left(\hat{p}_{i,Tr}^w\right)\right) \text{ and } N\left(\hat{p}_{i,Dep}^w, \text{cov}\left(\hat{p}_{i,Dep}^w\right)\right),$$

where $\hat{p}_{i,Tr}^w$ and $\text{cov}\left(\hat{p}_{i,Tr}^w\right)$ represent the uncertainty concerning the world coordinate system (denoted as w) and the covariance matrix of the 3D feature points triangulated by

the camera, respectively. These estimates are then fused to obtain a high-precision map. According to the derivation formula for triangulation and depth measurement uncertainty estimation, it is evident that a correlation exists between the two normal distributions. In this paper, a CI filter is employed to fuse these two correlated estimates, which allows for the fusion of two estimates with unknown cross-covariance and ensures consistent results. The formula [18] is as follows:

$$\text{cov}\left(\hat{p}_i^w\right) = \left[\omega \text{cov}^{-1}\left(\hat{p}_{i,Tr}^w\right) + (1 - \omega) \cdot \text{cov}^{-1}\left(\hat{p}_{i,Dep}^w\right)\right]^{-1} \quad (1)$$

$$\hat{p}_i^w = \text{cov}\left(\hat{p}_i^w\right) \cdot \left[\omega \text{cov}^{-1}\left(\hat{p}_{i,Tr}^w\right) \cdot \hat{p}_{i,Tr}^w + (1 - \omega) \cdot \text{cov}^{-1}\left(\hat{p}_{i,Dep}^w\right) \cdot \hat{p}_{i,Dep}^w\right] \quad (2)$$

where $\omega \in [0, 1]$ is the weight assigned to the uncertainty in the triangulation estimate and the depth measurement estimate, which can be calculated by minimizing the trace of the fused covariance [18], i.e.,

$$\omega^* = \underset{\omega}{\text{argmin}} \left\{ \text{tr} \left[\text{cov}\left(\hat{p}_i^w\right) \right] \right\}. \quad (3)$$

4. Derivation of the Wheel Encoder Model

4.1. Pre-Integration Model for Wheeled Encoder

There are multiple wheel encoder measurements between two adjacent frames [8]. In this paper, inspired by IMU pre-integration [6], a wheel encoder pre-integration model is derived to compute a low-frequency integration term that provides motion constraints for the two frames. Since the object of study is a ground mobile robot, its motion is approximated to be in a plane. Thus, the motion of the robot can be represented by a rotation angle, $\theta \in R$, around the z -axis and a 2D translation vector, $q \in R^2$, in the $x - y$ plane. The wheel encoder provides the mobile robot with measured values for linear velocity, v , and angular velocity, ω [41]. It is assumed that the rear wheel of the mobile robot is aligned with the positive x -axis of the wheel encoder coordinate system and that the left side of the rear wheel is aligned with the positive y -axis. The wheel encoder coordinate system follows the right-hand rule, and the movement model for the wheel odometry is shown in Figure 3. Figure 3 shows the robot moving from coordinate (x_k, y_k) at time t_k to the coordinate (x_{k+1}, y_{k+1}) at time t_{k+1} . The linear velocity of the mobile robot is measured by an incremental photoelectric encoder, which converts pulse data into the robot's linear velocity equation:

$$v = \frac{\pi d_w}{N \Delta t} n, \quad (4)$$

where d_w is the diameter of the rear wheel of the robot, N is the number of encoders lines, Δt is the sampling time, and n is the number of pulses during the sampling time. The linear velocities of the robot's drive wheels are calculated by Equation (4), and the true velocities of the left and right wheels are obtained by subtracting measurement noise from the observed values as follows [42]:

$$v_k^{el} = \tilde{v}_k^{el} - \eta_k^{eld}, \quad (5)$$

$$v_k^{er} = \tilde{v}_k^{er} - \eta_k^{erd}, \quad (6)$$

where $[\eta_k^{eld} \ \eta_k^{erd}]^T \in \mathbb{R}^2$ is the measurement noise caused by the wheel encoders, which is assumed to follow a zero-mean Gaussian distribution with covariance matrix $[\eta_k^{eld} \ \eta_k^{erd}]^T \in \mathbb{R}^2$. v_k^{el} and v_k^{er} are the estimated instantaneous linear velocities at moment t_k . $\tilde{v}_k^{el}$ and $\tilde{v}_k^{er}$ are the linear velocities converted from the measurements of the left and right wheel encoders at moment t_k . It is assumed that the robot operates in an ideal environment where the surface is flat and the wheels do not slip. The linear velocity of the wheels is denoted by $v_k^e = [\tilde{v}_k^{ex} \ 0 \ 0]^T - \eta_k^{mvd}$, where $\eta_k^{mvd} = [\eta_k^{m\omega x} \ \eta_k^{m\omega y} \ \eta_k^{m\omega z}]^T \in \mathbb{R}^3$ denotes the

measurement noise vector in the direction of x, y, z for the linear velocity. The angular velocity of the wheels is denoted by $\omega_k^e = [0 \ 0 \ \tilde{\omega}_k^{ez}]^T - \eta_k^{m\omega d}$, where $\tilde{\omega}_k^{ez}$ is the angular velocity converted from the measured value of the wheel encoder at moment t_k . $\eta_k^{m\omega d} = [\eta_k^{m\omega x} \ \eta_k^{m\omega y} \ \eta_k^{m\omega z}]^T \in \mathbb{R}^3$ represents the measurement noise vector in the direction of x, y, z for the angular velocity.

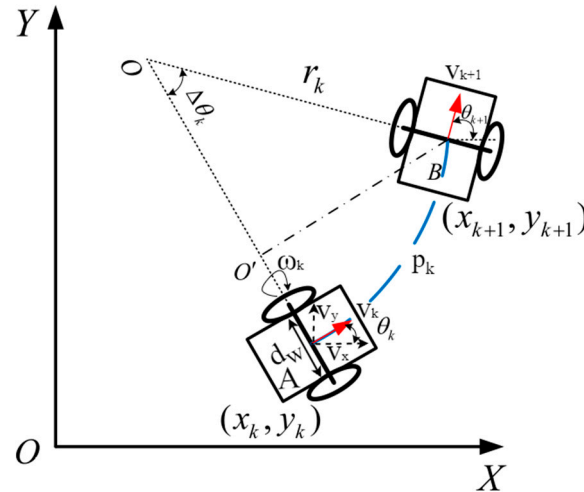


Figure 3. The motion model of the wheeled robot using wheel encoders. The figure illustrates the motion model of a mobile robot using wheel encoders in a 2D plane. The model describes the robot's trajectory between its position at time t_k , denoted as (x_{k+1}, y_{k+1}) , and its position at time t_{k+1} , denoted as (x_{k+1}, y_{k+1}) .

Due to the high frequency of the wheel encoder, the state transition model between two adjacent frames of wheel encoder measurements can be described as:

$$\theta_{k+1} = \theta_k + \omega_k \cdot \Delta t, \quad (7)$$

$$q_{k+1} = q_k + R(\theta_k) \cdot \begin{bmatrix} 0 \\ v_k \cdot \Delta t \end{bmatrix}. \quad (8)$$

Here, the Euler integration method is used, and by calculating Equations (7) and (8) in recursive form, the wheel encoder pre-integration term between two adjacent frames can be obtained. Since the ground robot operates in a 2D $x - y$ plane, the rotation matrix is expressed as:

$$R(\theta_k) = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{bmatrix}. \quad (9)$$

Due to noise in the wheel encoder measurements, the following Equation (10) is used to construct the measurement error model:

$$\begin{aligned} \bar{\omega} &= \omega + n_\omega \\ \bar{v}_x &= v_x + n_x \\ \bar{v}_y &= v_y + n_y \end{aligned} \quad (10)$$

In Equation (10), v_x and v_y denote the forward and left lateral linear velocities of the mobile robot, respectively. Assuming the noise of ω , v_x , and v_y is Gaussian white noise, with Gaussian distributions as follows: $n_\omega \sim N(0, \delta_\omega^2)$, $n_{v_x} \sim N(0, \delta_{v_x}^2)$, $n_{v_y} \sim N(0, \delta_{v_y}^2)$.

A linear recurrence relation for the state error of the pre-integrated term of the wheel encoder can be derived from Equations (7), (8) and (10):

$$\begin{bmatrix} \delta\theta_{k+1} \\ \delta q_{k+1} \end{bmatrix} = P_k \begin{bmatrix} \delta\theta_k \\ \delta q_k \end{bmatrix} + H_k \begin{bmatrix} n_\omega \\ n_v \end{bmatrix}. \quad (11)$$

In Equation (11), n_v denotes the measurement noise of v_x and v_y . The Jacobian matrices P_k and H_k can be calculated from Equations (12) and (13), respectively:

$$P_k = \begin{bmatrix} 1 & 0_{1 \times 2} \\ R(\theta_k) \cdot J \cdot \begin{bmatrix} 0 \\ v_k \cdot \Delta t \end{bmatrix} & I_{2 \times 2} \end{bmatrix}, \quad (12)$$

$$H_k = \begin{bmatrix} \Delta t & 0_{1 \times 2} \\ 0_{2 \times 1} & R(\theta_k) \cdot \Delta t \end{bmatrix}, \quad (13)$$

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (14)$$

After obtaining the linear transition equation for the state error at adjacent moments, the covariance matrix of the wheeled encoder pre-integration term between neighboring image frames can be calculated using the following Equation (15):

$$Q_{k+1} = P_k Q_k P_k^T + H_k \sum_n H_k^T, \quad (15)$$

where Q_k is the covariance matrix of the state quantities θ_k and q_k ; $\sum_n$ is the covariance matrix of the noise, which is calculated by the following Equation (16):

$$\sum_n = \begin{bmatrix} \delta_\omega^2 & 0 & 0 \\ 0 & \delta_{v_x}^2 & 0 \\ 0 & 0 & \delta_{v_y}^2 \end{bmatrix}. \quad (16)$$

4.2. Pre-Integration Error of Wheeled Encoder

Based on the derivation of wheel encoder pre-integration in Section 4.1, the pre-integration error for the wheel encoders is [42]:

$$r_{\Delta\phi_{ij}^e}^e \triangleq \text{Log}[(\Delta\tilde{R}_{ij}^e)^T \Delta R_{ij}^e], \quad (17)$$

$$r_{\Delta p_{ij}^e}^e \triangleq \Delta p_{ij}^e - \Delta\tilde{p}_{ij}^e. \quad (18)$$

In the above equation, ΔR_{ij}^e and Δp_{ij}^e are the predicted values of the wheel encoders pre-integration, and $\Delta\tilde{R}_{ij}^e$ and $\Delta\tilde{p}_{ij}^e$ are the observed values of the wheel encoders pre-integration.

4.2.1. Jacobian Matrix of Rotation to State Variables

Equation (17) represents the rotational error in the pre-integration of the wheel encoders, which does not involve p_{wi}^e and p_{wj}^e . Therefore, the Jacobian matrix with respect to each of these state variables is 0, and its Jacobian matrix with respect to $\delta\phi_i^e$ and $\delta\phi_j^e$ is given by [42]:

$$\frac{\partial r_{\Delta\phi_{ij}^e}^e}{\partial \delta\phi_i^e} = -J_r^{-1}(r_{\Delta\phi_{ij}^e}^e)(R_{wi}^{eT} R_{wj}^e R_{ce})^T, \quad (19)$$

$$\frac{\partial r_{\Delta\phi_{ij}^e}^e}{\partial \delta\phi_j^e} = J_r^{-1}(r_{\Delta\phi_{ij}^e}^e) R_{ce}^T. \quad (20)$$

In the above equation, R_{ce} is the rotation matrix from the wheel encoder coordinate system to the camera coordinate system, which can be obtained from the intelligent vehicle model built in ROS.

4.2.2. Jacobian Matrix of Position to State Variables

Equation (18) represents the pre-integrated position error of the wheel encoders. Its Jacobian matrix with respect to p_{wi}^e , p_{wj}^e , $\delta\phi_i$ and $\delta\phi_j$ is given by [42]:

$$\frac{\partial r_{\Delta p_{ij}}^e}{\partial \delta p_{wi}^e} = -R_{ce}^T R_{wi}^{eT}, \quad (21)$$

$$\frac{\partial r_{\Delta p_{ij}}^e}{\partial \delta p_{wj}^e} = R_{ce}^T R_{wi}^{eT}, \quad (22)$$

$$\frac{\partial r_{\Delta p_{ij}}^e}{\partial \delta \phi_i^e} = R_{ce}^T (R_{wi}^{eT} (R_{wj}^e p_{ce} + p_{wj}^e - p_{wi}^e))^{\wedge}, \quad (23)$$

$$\frac{\partial r_{\Delta p_{ij}}^e}{\partial \delta \phi_j^e} = -R_{ce}^T R_{wi}^{eT} R_{wj}^e p_{ce}^{\wedge}. \quad (24)$$

In the above equation, p_{ce} is the translation matrix from the wheel encoder coordinate system to the camera coordinate system, which can be obtained from the intelligent vehicle model built in ROS.

5. Experimental Analysis of the VEOS3-TEDM Algorithm

5.1. Experimental Analysis of Open-Source Datasets

The combination of the three datasets—RGB-D, IMU, and wheel encoders—is not currently available in some public datasets. To validate the effectiveness of the VEOS3-TEDM algorithm proposed in this paper, open-source datasets from the literature [42] are used. These datasets are applied to a differential wheeled robot equipped with a Kinect v2 RGB-D camera, two-wheel encoders, and an IMU sensor. The ground truth data for these datasets are provided via a total station, which generates the true trajectory of the robot at approximately 10 Hz by measuring the position of a prismatic reflector fixed to the robot, with an accuracy close to 1 mm. These datasets were recorded using a mobile robot on an experimental platform, with captured scenes including long straight corridors and laboratories. To validate the deviation of the estimated trajectories from the true trajectories, a public benchmarking tool [43] was used to evaluate the Absolute Trajectory Error (ATE), which reflects the SLAM accuracy by calculating the Root-Mean-Squared Error (RMSE) of the system output. For evaluating algorithm accuracy, only the translation error is considered, and thus the Average Translation Error is defined as follows:

$$ATE_{trans} = \sqrt{\frac{1}{N} \sum_{i=1}^N \|T_{trans}(T_{gt,i}^{-1} T_{esti,i})\|_2^2}. \quad (25)$$

In the above equation, $T_{gt,i}$ and $T_{esti,i}$ represent the true and estimated values, respectively. T_{trans} denotes the translation portion of the variable inside the parentheses. The RMSE values in Tables 1 and 2 were obtained using an EVO (Evaluation of Odometry, version 1.29.0) tool. The VEOS3-TEDM algorithm was evaluated using two datasets: one from a long straight corridor scene and another from a laboratory scene. The frame rate of the camera was 15 Hz, while the IMU and wheel encoders operated at 200 Hz. The corridor dataset contains 897 frames, and the laboratory dataset contains 875 frames. Table 1 shows the results of the algorithm comparison for the corridor dataset, while Table 2 presents the results for the laboratory dataset. An “X” in the table indicates that the value could not be computed, meaning that the algorithm failed to track or complete initialization. The pose estimation ratio is defined as the number of frames successfully tracked by the algorithm divided by the total number of frames in the dataset.

Table 1. The comparison of algorithm results on the corridor dataset.

Algorithm Name	Sensors	RMSE (m)	Average Tracking Time per Frame (ms)	Pose Estimation Ratio (%)
ORB-SLAM2	Encoders	0.463	1	100
	RGB-D	X	X	45
ORB-SLAM3	RGB-D	X	X	94
	RGB-D + IMU	X	X	5
VEORB-SLAM3	RGB-D + Encoders	0.094	17	100
VIEORB-SLAM3	RGB-D + IMU + Encoder	0.114	26	100
VOS3-TEDM	CI-TEDM	X	X	96
VEOS3-TEDM	CI-TEDM + Encoders	0.083	14	100
VIEOS3-TEDM	CI-TEDM + IMU + Encoders	0.107	23	100

Table 2. The comparison of algorithm results on the laboratory dataset.

Algorithm Name	Sensors	RMSE (m)	Average Tracking Time per Frame (ms)	Pose Estimation Ratio (%)
ORB-SLAM2	Encoders	0.382	1	100
	RGB-D	X	X	92
ORB-SLAM3	RGB-D	0.1167	22	100
	RGB-D + IMU	X	X	5
VEORB-SLAM3	RGB-D + Encoders	0.1104	21	100
VIEORB-SLAM3	RGB-D + IMU + Encoders	0.128	33	100
VOS3-TEDM	CI-TEDM	0.1054	20	100
VEOS3-TEDM	CI-TEDM + Encoders	0.087	16	100
VIEOS3-TEDM	CI-TEDM + IMU + Encoders	0.115	29	100

The texture of the corridor in those datasets is very weak. During the turn of the intelligent vehicle, the camera is very close to the white wall surface, resulting in a very small number of feature points or even none at all. This causes purely visual SLAM tracking to fail, so both the ORB-SLAM2 and ORB-SLAM3 algorithms using the RGB-D camera fail to track. In Table 1, it can be observed that two types of algorithms are unable to compute the RMSE and the average tracking time per frame. However, ORB-SLAM3, with its multiple map modes, achieves a final pose estimation ratio that is twice as high as that of the ORB-SLAM2 algorithm. When ORB-SLAM3 utilizes the combined form of RGB-D and IMU, the IMU pre-initialization requires the intelligent vehicle to perform sufficient motion to achieve the necessary excitation for IMU initialization. During this time, visual tracking must not fail for an extended period. Nevertheless, visual tracking tends to fail during the vehicle's turning process. The IMU was unable to complete initialization within the dataset. While the intelligent vehicle's pose estimation can be achieved using only the wheel encoder, this method results in the shortest average tracking time per frame but produces a relatively high RMSE, which does not meet practical requirements. As shown in Table 1, the VEOS3-TEDM algorithm proposed in this paper outperforms the other algorithms in the corridor datasets. Specifically, in terms of RMSE, it achieves reductions of 38 cm, 1.1 cm, 3.1 cm, and 2.4 cm compared to the encoders and VEORB-SLAM3, VIEORB-SLAM3, and VIEOS3-TEDM algorithms, respectively. Regarding average tracking time per frame, the VEOS3-TEDM algorithm is 3 ms, 12 ms, and 9 ms faster than the VEORB-SLAM3, VIEORB-SLAM3, and VIEOS3-TEDM algorithms, respectively. These results indicate that the VEOS3-TEDM algorithm offers superior performance in both RMSE and average tracking time per frame within the corridor dataset.

The laboratory dataset is rich in texture information, but the high-speed motion of the intelligent vehicle causes tracking failures in ORB-SLAM2. After adding the IMU, ORB-SLAM3 has more stringent initialization requirements, which the intelligent vehicle platform struggles to meet. This results in the inability to calculate the RMSE and average tracking time per frame. However, the final pose estimation ratios of ORB-SLAM3 and ORB-SLAM2 are nearly identical in these datasets. Table 2 shows that, in the laboratory dataset, the VEOS3-TEDM algorithm proposed in this paper achieves RMSE reductions of 29.5 cm, 2.97 cm, 2.34 cm, 4.1 cm, 1.84 cm, and 2.8 cm compared to the encoders and ORB-SLAM3, VEORB-SLAM3, VIEORB-SLAM3, VOS3-TEDM, and VIDEOS3-TEDM algorithms, respectively. In terms of average tracking time per frame, VEOS3-TEDM is 6 ms, 5 ms, 17 ms, 4 ms, and 13 ms faster than ORB-SLAM3, VEORB-SLAM3, VIEORB-SLAM3, VOS3-TEDM, and VIDEOS3-TEDM, respectively. These results imply that the VEOS3-TEDM algorithm outperforms other algorithms in both RMSE and average tracking time per frame in the laboratory dataset.

Tables 1 and 2 show that the ORB-SLAM3 algorithm fails to track both in the corridor and laboratory datasets when implemented using RGB-D and IMU fusion. This failure is attributed to the lack of feature points, fast motion, and light reflections. However, the VEOS3-TEDM algorithm proposed in this paper effectively addresses these temporary failure issues, achieving satisfactory accuracy and successfully recovering the entire trajectory. The results demonstrate that the VEOS3-TEDM algorithm generally achieves better accuracy and stability due to the fusion of ORB triangulation estimation with RGB-D depth measurement and the integration of encoder information. This approach aids in identifying outliers, enhancing prediction, and improving pose estimation. The improvements are particularly noticeable in datasets with higher motion speeds.

The process of running the VEOS3-TEDM algorithm on the corridor and laboratory datasets is illustrated in Figure 4. During the execution, the pose of all keyframes is based on the world coordinate system, which is established at the frame where initialization is completed. After initialization, map points are created and their 3D coordinates are relative to the world coordinate system. The map points are categorized into two types: local map points and global map points. Local map points assist in localization during the tracking process. In Figure 4, the blue frames represent keyframes, red frames represent initial keyframes, and green frames represent current frames. The VEOS3-TEDM algorithm achieves satisfactory accuracy and successfully recovers the entire trajectory. This demonstrates that the VEOS3-TEDM algorithm, which integrates ORB triangulation estimation, RGB-D depth measurement, and encoder information, effectively fulfills the primary goal of this paper.

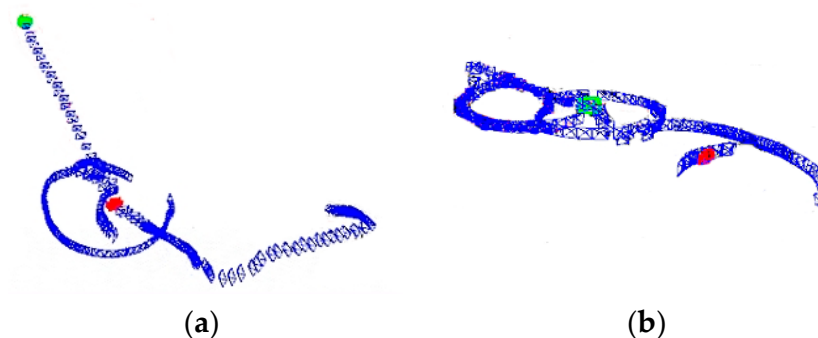


Figure 4. The process of running datasets in the VEOS3-TEDM algorithm: (a) corridor scene and (b) laboratory scene. The blue frames represent keyframes, red frames represent initial keyframes, and green frames represent current frames.

The tracking process of the VEOS3-TEDM algorithm on the corridor and laboratory datasets is shown in Figure 5, where the green boxes indicate the recognized 3D feature points. In Figure 5, it is evident that angular feature points and those with significant changes in gray value are well identified. This demonstrates that the VEOS3-TEDM algo-

rithm can stably track the image without failure. This stability is achieved by incorporating encoder information into the algorithm. Even if the visual tracking fails, the encoder data can compensate, allowing the algorithm to connect the two states and maintain continuous stable tracking.

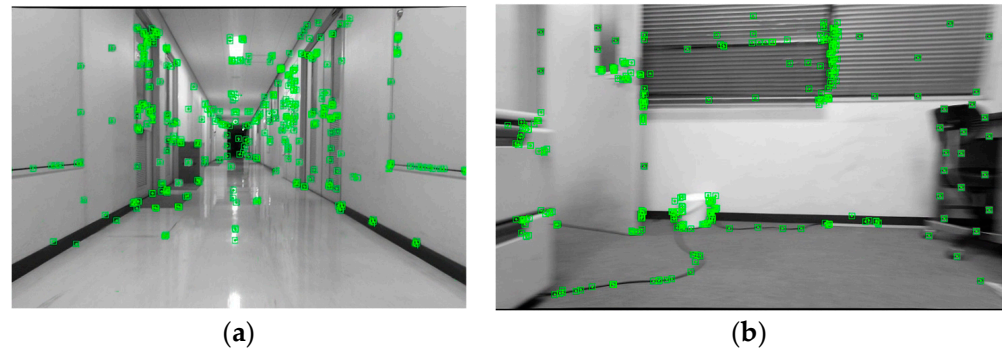


Figure 5. The process of tracking datasets in the VEOS3-TEDM algorithm: (a) corridor scene and (b) laboratory scene. The green boxes in the figure represent key feature points detected by VEOS3-TEDM algorithm.

When the VEOS3-TEDM algorithm finishes running, it generates the pose information of the keyframes. Figure 6 shows a comparison between the pose estimates generated by the VEOS3-TEDM algorithm using the open-source software EVO and the true trajectories. As seen in Figure 6, the estimated trajectories closely match the true trajectories on flat and straight roads, with only minor errors observed in the corridor scene. However, there is some deviation in the vehicle's trajectory at turns, where the camera is very close to the white wall, resulting in few or no feature points. This causes the pure vision ORB-SLAM3 tracking to fail. Nonetheless, the VEOS3-TEDM algorithm is able to track continuously and stably due to the compensation provided by encoder data. In the laboratory scene, the estimated trajectory of the vehicle closely matches the true trajectory, as the scene is rich in texture information, enabling accurate recognition of 3D feature points, and thus producing a more accurate estimated trajectory.

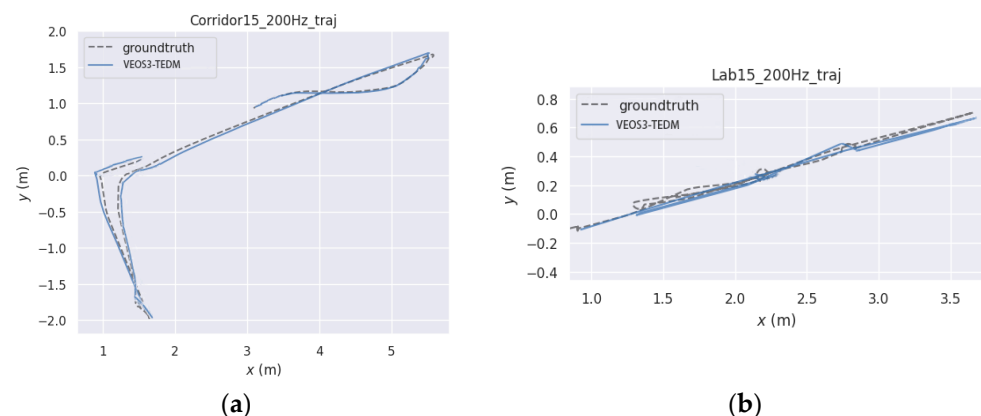


Figure 6. The comparison between estimated and true trajectory in the VEOS3-TEDM algorithm: (a) corridor scene and (b) laboratory scene.

As shown in Figure 7, the trajectory estimation errors in the x -axis and y -axis directions are relatively small in both the corridor and laboratory scenes. However, the errors in the z -axis direction are comparatively larger. This is due to the fact that encoder information is recorded more accurately in the horizontal direction, while larger errors are observed in the vertical direction. This suggests that further improvements are needed in the algorithms to enhance the accuracy of depth value processing. Additionally, the trajectory estimation errors in the x -axis and y -axis directions are smaller in the corridor scene compared to

the laboratory scene. This is likely because the corridor scene is more open, with fewer obstacles, leading to a more accurate trajectory estimation in the horizontal directions.

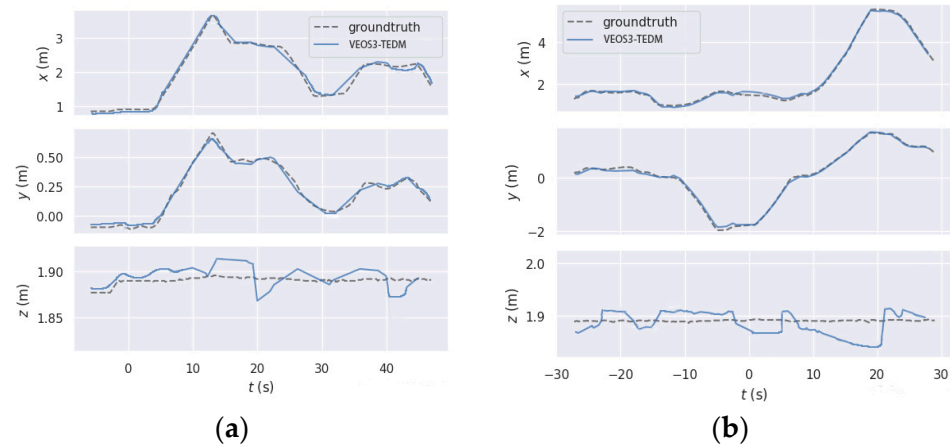


Figure 7. The comparison between true and estimated trajectories in x , y and z directions, using the VEOS3-TEDM algorithm: (a) corridor scene and (b) laboratory scene.

Finally, the intelligent vehicle successfully constructed 3D point cloud maps of the corridor and laboratory scenes, as shown in Figure 8. The constructed 3D point cloud maps are consistent with the actual scenes. For approximate navigation, the positional accuracy of obstacles, which are defined as points located between the ground plane and the height of the robot, is considered acceptable. However, some redundancies remain due to inevitable errors introduced by the pure encoders, which can be minimized through more closed-loop detection, image measurements, and better prior calibration. The point cloud representation of the map points is mainly used to highlight visible errors in the estimated trajectories of the keyframes. However, for ROS navigation, future research should focus on using a grid map form that incorporates advanced filtering techniques [43].

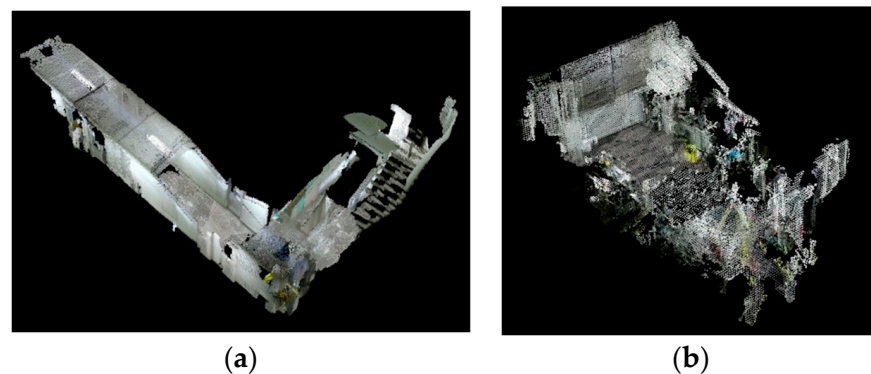


Figure 8. 3D point cloud maps: (a) corridor scene and (b) laboratory scene.

5.2. Experimental Analysis of Real-World Environments

The experimental platform is shown in Figure 9. The mobile robot is divided into a bottom layer, an intermediate layer, and an upper layer. The bottom layer contains the lithium battery, lower computer, motor driver, software-based emergency stop button, servo motor, and motor. The middle layer houses the single-wire LiDAR, power switch, and emergency stop button. The upper layer contains the upper computer, RGB-D camera, mechanical arm, voltage reduction module, and power switch for the upper computer. The locations of the devices held in the bottom and upper layers of the mobile robot are shown in Figure 10.

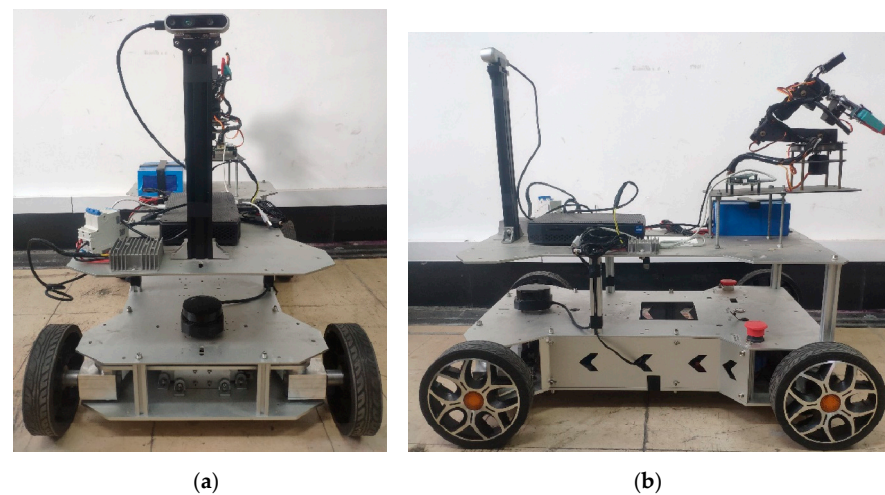


Figure 9. Images of the experimental platform: (a) front view and (b) left view.

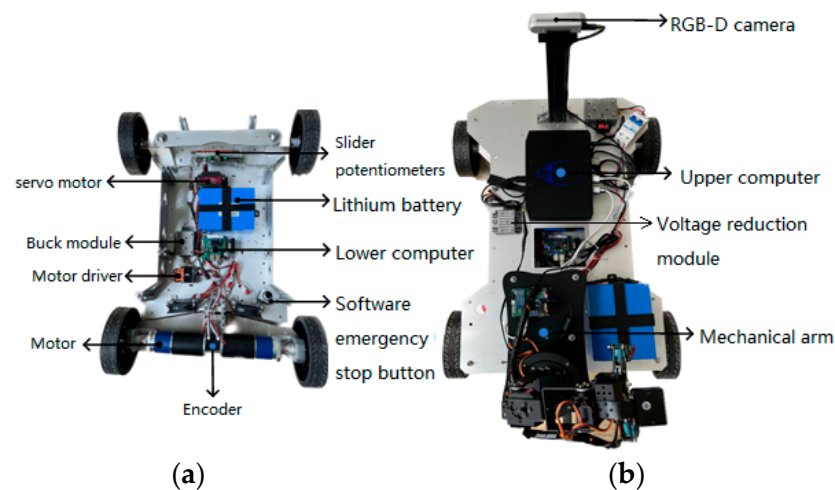


Figure 10. The location of various components on the mobile robot: (a) bottom level and (b) upper level.

The proposed VEOS3-TEDM algorithm is first ported to a mobile robot. Afterward, the algorithm is validated using the mobile robot. In this paper, data sequences of real-world indoor scenes are captured by an experimental platform. For the collected indoor scene data, it is challenging to obtain the ground truth for each moment on the experimental platform. Instead, the dataset is collected by controlling the motion of the experimental platform so that the data sequence forms a large closed loop at the end. Specifically, the experimental platform starts at the initial point, completes a full circle, and returns to the starting point. Thus, the start and end segments of the data sequence are in the same scenario.

The RGB-D camera used in the process has a frame rate of 30 Hz and an image resolution of 640×480 . The IMU operates at 200 Hz, and the wheel encoders operate at 100 Hz. The data from the wheel encoders are collected by the lower computer and transmitted to the upper computer via serial communication. The operating environments include the laboratory, a long straight corridor, and a hall. These environments are relatively large, and the tracking process of the experimental platform during the experiment is shown in Figure 11. Figure 11 indicates that the experimental platform captures a relatively large number of 3D feature points in the laboratory, corridor, and hall scenes, including distinct angular points and points with significant changes in gray value. This demonstrates that the VEOS3-TEDM algorithm achieves the desired accuracy and stability. However, the 3D feature points obtained in weakly textured scenes (e.g., white walls) are relatively sparse, suggesting that texture richness directly impacts the accuracy and stability of localization

and map building. Therefore, to achieve high localization and map-building accuracy, the mobile robot should operate in texture-rich scenes.

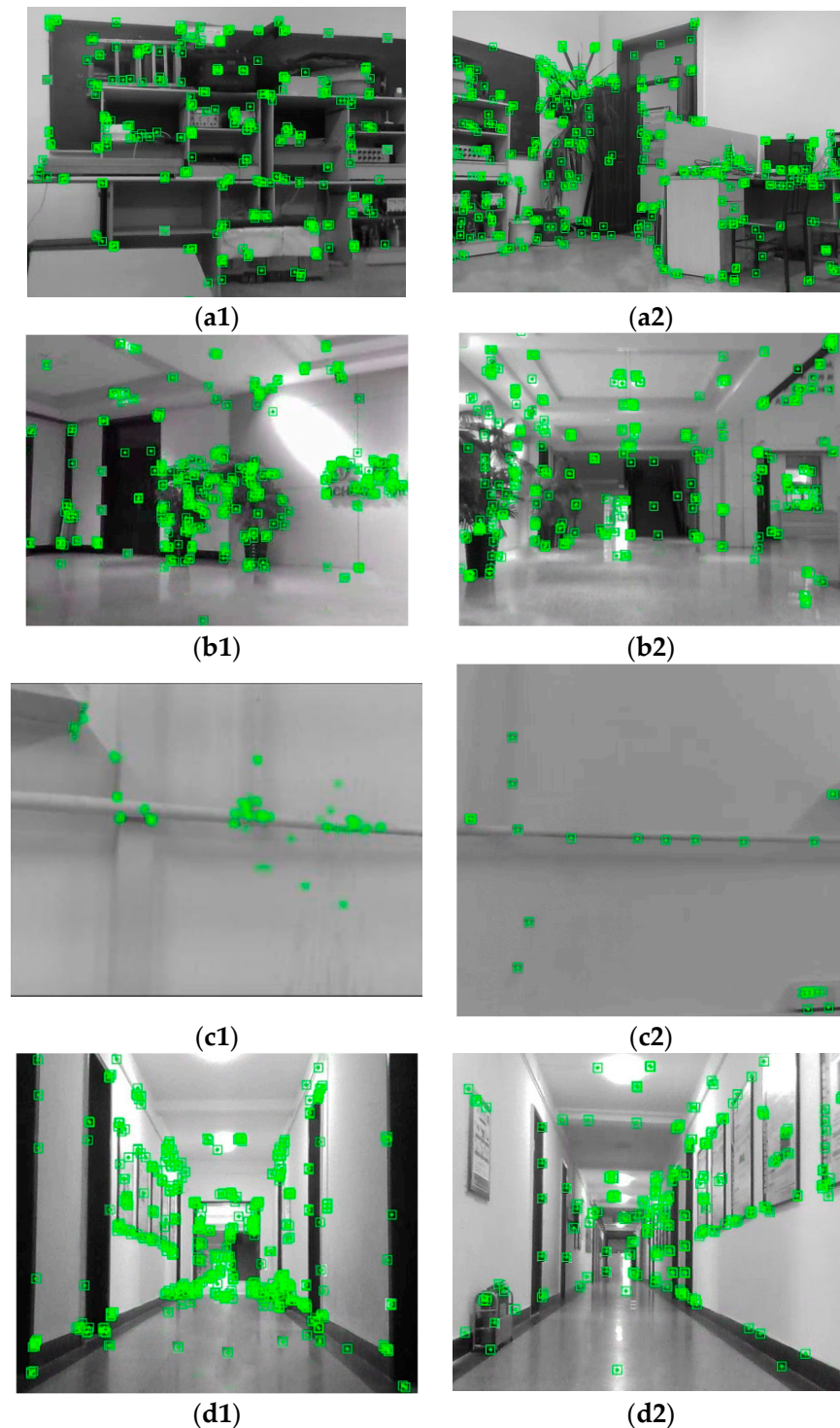


Figure 11. The process of tracking real-world environments in the VEOS3-TEDM algorithm: (a1,a2) laboratory, (b1,b2) hall, (c1,c2) weak texture scene, (d1,d2) long straight corridor. The green boxes in the figure represent key feature points detected by VEOS3-TEDM algorithm.

Figure 12 illustrates the estimated trajectory of VEOS3-TEDM running in a real-world environment compared to the ground truth, with an obtained RMSE of 0.091. The effectiveness and robustness of the proposed VEOS3-TEDM algorithm were validated in real-world

environments. Additionally, the estimated trajectory closely aligns with the ground truth, achieving the research objectives outlined in this paper.

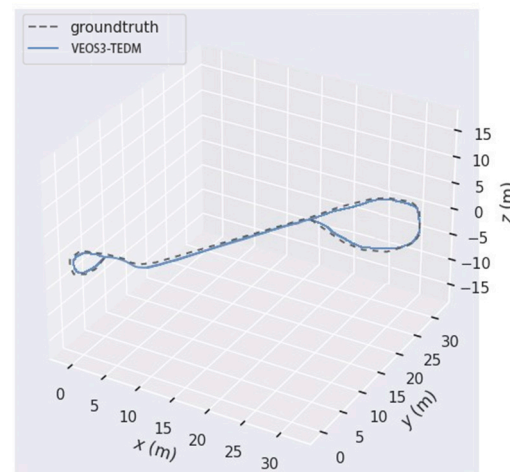


Figure 12. A comparison of estimated and true trajectories in real-world environments using the VEOS3-TEDM algorithm.

6. Conclusions

This paper introduces wheel odometry into the classical ORB-SLAM3 framework. The main contribution is the proposal of a novel combination of CI-TEDM and wheel odometry, offering a new approach to enhancing the accuracy and robustness of SLAM. Additionally, this paper derives the wheel encoder pre-integration model, error model, rotation-to-state-variable Jacobian matrix, and position-to-state-variable Jacobian matrix, all of which are employed for local BA optimization to improve SLAM results. Through these advancements, a more accurate and robust VEOS3-TEDM algorithm is achieved. Experiments on open-source datasets and in real-world environments demonstrate that the proposed VEOS3-TEDM algorithm surpasses state-of-the-art methods in both positioning and mapping accuracy. Furthermore, the proposed method is extensible to other optimization-based indirect visual SLAM or visual-inertial SLAM systems.

Author Contributions: Conceptualization, Z.-W.M. and W.-S.C.; methodology, Z.-W.M. and W.-S.C.; software, Z.-W.M.; validation, Z.-W.M.; formal analysis, Z.-W.M. and W.-S.C.; resources, Z.-W.M. and W.-S.C.; data curation, Z.-W.M.; writing—original draft preparation, Z.-W.M.; writing—review and editing, Z.-W.M. and W.-S.C.; visualization, Z.-W.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, C.H.; Fan, B.; Hu, J.; Pan, Q.; Xu, Z. Homography-based camera pose estimation with known gravity direction for UAV navigation. *Sci. China Inf. Sci.* **2021**, *64*, 1–3. [\[CrossRef\]](#)
2. Wang, K.; Zhao, G.; Lu, J. A Deep Analysis of Visual SLAM Methods for Highly Automated and Autonomous Vehicles in Complex Urban Environment. *IEEE Trans. Intell. Transp. Syst.* **2024**, *4*, 1–18. [\[CrossRef\]](#)
3. Zhao, Y.L.; Hong, Y.T.; Huang, H.P. Comprehensive Performance Evaluation between Visual SLAM and LiDAR SLAM for Mobile Robots: Theories and Experiments. *Appl. Sci.* **2024**, *14*, 3945. [\[CrossRef\]](#)
4. Hu, X.; Zhu, L.; Wang, P.; Yang, H.; Li, X. Improved ORB-SLAM2 mobile robot vision algorithm based on multiple feature fusion. *IEEE Access* **2023**, *11*, 100659–100671. [\[CrossRef\]](#)

5. Al-Tawil, B.; Hempel, T.; Abdelrahman, A.; Al-Hamadi, A. A review of visual SLAM for robotics: Evolution, properties, and future applications. *Front. Robot. AI* **2024**, *11*, 1347985. [\[CrossRef\]](#)
6. Campos, C.; Elvira, R.; Rodríguez, J.J.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [\[CrossRef\]](#)
7. Lin, J.; Peng, J.; Hu, Z.; Xie, X.; Peng, R. Orb-slam, imu and wheel odometry fusion for indoor mobile robot localization and navigation. *Acad. J. Comput. Inf. Sci* **2020**, *27*, 131–141.
8. Lee, C.; Peng, J.; Xiong, Z. Asynchronous fusion of visual and wheel odometer for SLAM applications. In Proceedings of the 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Boston, MA, USA, 6–9 July 2020; pp. 1990–1995.
9. Zhou, W.; Pan, Y.; Liu, J.; Wang, T.; Zha, H. Visual-Inertial-Wheel Odometry With Wheel-Aided Maximum-a-Posteriori Initialization for Ground Robots. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4814–4821. [\[CrossRef\]](#)
10. Anousaki, G.; Gikas, V.; Kyriakopoulos, K. INS-Aided Odometry and Laser Scanning Data Integration for Real Time Positioning and Map-Building of Skid-Steered Vehicles. In Proceedings of the International Symposium on Mobile Mapping Technology, Padua, Italy, 29–31 May 2007.
11. Zhou, W.; Zhou, R. Vision SLAM algorithm for wheeled robots integrating multiple sensors. *PLoS ONE* **2024**, *19*, e0301189. [\[CrossRef\]](#)
12. Cabrera-Ávila, E.V.; da Silva, B.M.; Gonçalves, L.M. Nonlinearly Optimized Dual Stereo Visual Odometry Fusion. *J. Intell. Robot. Syst.* **2024**, *110*, 56. [\[CrossRef\]](#)
13. Chen, Y.; Huang, S.; Fitch, R. Active SLAM for mobile robots with area coverage and obstacle avoidance. *IEEE/ASME Trans. Mechatron.* **2020**, *25*, 1182–1192. [\[CrossRef\]](#)
14. Chen, Y.; Huang, S.; Zhao, L.; Dissanayake, G. Cramér–Rao bounds and optimal design metrics for pose-graph SLAM. *IEEE Trans. Robot.* **2021**, *37*, 627–641. [\[CrossRef\]](#)
15. Mu, B.; Giamou, M.; Paull, L.; Agha-mohammadi, A.A.; Leonard, J.; How, J. Information-based active SLAM via topological feature graphs. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 5583–5590.
16. Li, N.; Zhou, F.; Yao, K.; Hu, X.; Wang, R. Multisensor Fusion SLAM Research Based on Improved RBPF-SLAM Algorithm. *J. Sens.* **2023**, *2023*, 3100646. [\[CrossRef\]](#)
17. Lin, X.; Huang, Y.; Sun, D.; Lin, T.Y.; Englot, B.; Eustice, R.M.; Ghaffari, M. A Robust Keyframe-Based Visual SLAM for RGB-D Cameras in Challenging Scenarios. *IEEE Access* **2023**, *11*, 97239–97249. [\[CrossRef\]](#)
18. Yuan, J.; Zhu, S.; Tang, K.; Sun, Q. ORB-TEDM: An RGB-D SLAM approach fusing ORB triangulation estimates and depth measurements. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–15. [\[CrossRef\]](#)
19. Vakhitov, A.; Ferraz, L.; Agudo, A.; Moreno-Noguer, F. Uncertainty-aware camera pose estimation from points and lines. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4659–4668.
20. Belter, D.; Nowicki, M.; Skrzypczyński, P. Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1279–1284.
21. Belter, D.; Nowicki, M.; Skrzypczyński, P. Modeling spatial uncertainty of point features in feature-based RGB-D SLAM. *Mach. Vis. Appl.* **2018**, *29*, 827–844. [\[CrossRef\]](#)
22. Zhang, L.; Zhang, Y. Improved feature point extraction method of ORB-SLAM2 dense map. *Assem. Autom.* **2022**, *42*, 552–566. [\[CrossRef\]](#)
23. Lee, S.W.; Hsu, C.M.; Lee, M.C.; Fu, Y.T.; Atas, F.; Tsai, A. Fast point cloud feature extraction for real-time slam. In Proceedings of the 2019 International Automatic Control Conference (CACs), Keelung, Taiwan, 13–16 November 2019; pp. 1–6.
24. Zhou, F.; Zhang, L.; Deng, C.; Fan, X. Improved point-line feature based visual SLAM method for complex environments. *Sensors* **2021**, *21*, 4604. [\[CrossRef\]](#)
25. Gomez-Ojeda, R.; Moreno, F.A.; Zuniga-Noël, D.; Scaramuzza, D.; Gonzalez-Jimenez, J. PL-SLAM: A stereo SLAM system through the combination of points and line segments. *IEEE Trans. Robot.* **2019**, *35*, 734–746. [\[CrossRef\]](#)
26. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Volume 8690, pp. 834–849.
27. Eudes, A.; Lhuillier, M. Error propagations for local bundle adjustment. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2411–2418.
28. Barfoot, T.D.; Furgale, P.T. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robot.* **2014**, *30*, 679–693. [\[CrossRef\]](#)
29. Li, L.; Yang, M. Joint localization based on split covariance intersection on the Lie group. *IEEE Trans. Robot.* **2021**, *37*, 1508–1524. [\[CrossRef\]](#)
30. Mur-Artal, R.; Montiel, J.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [\[CrossRef\]](#)
31. Wang, Y.; Ng, Y.; Sa, I.; Parra, A.; Rodriguez, C.; Lin, T.J.; Li, H. Mavis: Multi-camera augmented visual-inertial slam using se2 (3) based exact imu pre-integration. *arXiv* **2023**, arXiv:2309.08142.

32. Wang, Z.; Peng, Z.; Guan, Y.; Wu, L. Manifold regularization graph structure auto-encoder to detect loop closure for visual SLAM. *IEEE Access* **2019**, *7*, 59524–59538. [[CrossRef](#)]
33. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [[CrossRef](#)]
34. Kazerooni, I.A.; Fitzgerald, L.; Dooley, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [[CrossRef](#)]
35. Cai, D.; Li, R.; Hu, Z.; Lu, J.; Li, S.; Zhao, Y. A comprehensive overview of core modules in visual SLAM framework. *Neurocomputing* **2024**, *590*, 127760. [[CrossRef](#)]
36. Usenko, V.; Engel, J.; Stückler, J.; Cremers, D. Direct visual-inertial odometry with stereo cameras. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1885–1892.
37. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [[CrossRef](#)]
38. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust visual inertial odometry using a direct EKF-based approach. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.
39. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [[CrossRef](#)]
40. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163.
41. Cho, B.S.; Moon, W.S.; Seo, W.J.; Baek, K.R. A dead reckoning localization system for mobile robots using inertial sensors and wheel revolution encoding. *J. Mech. Sci. Technol.* **2011**, *25*, 2907–2917. [[CrossRef](#)]
42. Zhu, Z.; Kaizu, Y.; Furuhashi, K.; Imou, K. Visual-inertial RGB-D SLAM with encoders for a differential wheeled robot. *IEEE Sens. J.* **2021**, *22*, 5360–5371. [[CrossRef](#)]
43. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.