

Article

# Positioning of Quadruped Robot Based on Tightly Coupled LiDAR Vision Inertial Odometer

Fangzheng Gao <sup>1,\*</sup>, Wenjun Tang <sup>1</sup>, Jiacai Huang <sup>2</sup> and Haiyang Chen <sup>1</sup><sup>1</sup> School of Automation, Nanjing Institute of Technology, Nanjing 211167, China; y00450200319@njit.edu.cn (W.T.); haiyangchen@njit.edu.cn (H.C.)<sup>2</sup> Industrial Center, Nanjing Institute of Technology, Nanjing 211167, China; huangjc@njit.edu.cn

\* Correspondence: gaofz@njit.edu.cn

**Abstract:** Quadruped robots, an important class of unmanned aerial vehicles, have broad potential for applications in education, service, industry, military, and other fields. Their independent positioning plays a key role for completing assigned tasks in a complex environment. However, positioning based on global navigation satellite systems (GNSS) may result in GNSS jamming and quadruped robots not operating properly in environments sheltered by buildings. In this paper, a tightly coupled LiDAR vision inertial odometer (LVIO) is proposed to address the positioning inaccuracy of quadruped robots, which have poor mileage information obtained through legs and feet structures only. With this optimization method, the point cloud data obtained by 3D LiDAR, the image feature information obtained by binocular vision, and the IMU inertial data are combined to improve the precise indoor and outdoor positioning of a quadruped robot. This method reduces the errors caused by the uniform motion model in laser odometer as well as the image blur caused by rapid movements of the robot, which can lead to error-matching in a dynamic scene; at the same time, it alleviates the impact of drift on inertial measurements. Finally, the quadruped robot in the laboratory is used to build a physical platform for verification. The experimental results show that the designed LVIO effectively realizes the positioning of four groups of robots with high precision and strong robustness, both indoors or outdoors, which verify the feasibility and effectiveness of the proposed method.

**Keywords:** positioning of quadruped robot; lidar visual inertial odometer; tightly coupled nonlinear optimization



**Citation:** Gao, F.; Tang, W.; Huang, J.; Chen, H. Positioning of Quadruped Robot Based on Tightly Coupled LiDAR Vision Inertial Odometer. *Remote Sens.* **2022**, *14*, 2945.  
<https://doi.org/10.3390/rs14122945>

Academic Editors: María García Fernández and Guillermo Álvarez-Narciandi

Received: 5 May 2022

Accepted: 17 June 2022

Published: 20 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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/>).

## 1. Introduction

Quadruped robots have broad potential for applications in education, service, industry, military, and other fields, due to their advantages, such as good flexibility and strong environmental adaptability [1]. The independent positioning of quadruped robots is key factor in completing assigned tasks in a complex environment and has become a research hotspot [2]. At present, the mainstream positioning technologies include map matching [3], mileage estimation [4], satellite and beacon [5], and so on. The positioning based on map matching requires expensive high-precision sensors to establish prior maps [6]. In studies, satellite-based positioning accuracy is not high, and the signal under the shelter of buildings is weak [7]. Beacon-based positioning is also limited to the layout of positioning base stations [8]. As compared to the aforementioned methods, location-positioning based on mileage estimation has advantages in both cost and the scope of application.

The design of the current odometer has mostly been applied to wheeled robots, whose mileage data are obtained by measuring the rotation of the wheel using a built-in encoder to achieve accurate positioning information. However, the wheel odometer has been shown as idle on smooth ground and cannot measure wheel “skidding”, which adversely affects the positioning accuracy [9]. However, as opposed to wheeled robots, legged robots are driven by motorized leg joints, which cannot provide accurate mileage data when relying

solely on its own structure. Consequently, the mainstream non-wheel and non-contact odometer solutions have been based on vision and laser measurements [10].

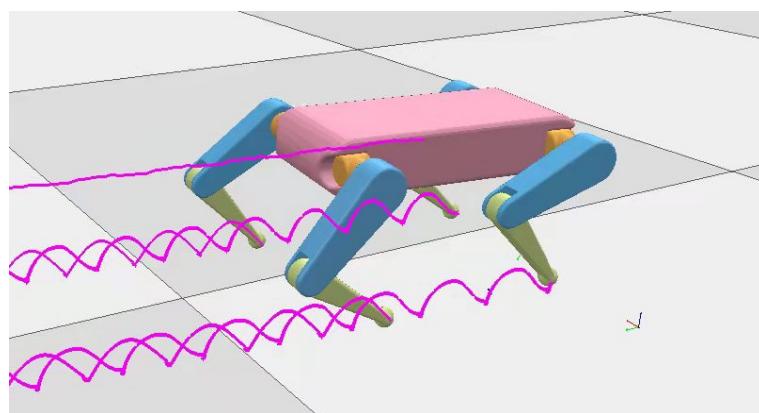
In recent years, with the continuous development of vision, inertial measurement, and multi-sensor fusion technology, visual inertial odometers (VIOs) have employed simultaneous localization and mapping (SLAM) [11]. Image feature information has been added to the tightly coupled vector state method so that the visual and inertial data can provide complementary advantages, and this has become a focus of mainstream research [12]. With the fusion of visual and inertial data, tightly coupled VIO has been categorized as either a filtering (such as MSCKF [13]) or an optimization method (such as ORB-SLAM [14] and VINS-Mono [15]).

The filtering method adopts extended Kalman filter (EKF), which is simple in form and widely used, but it has many limitations. Firstly, the state at moment  $k$  in the filter method is only relevant to the moment  $k - 1$ . The nonlinear optimization method, on the other hand, tends to use all historical data, so that more information is used. Secondly, the filtering method has nonlinear errors, which has been a significant challenge [16]. In optimization, although linearized approximation may also be performed, the Taylor expansion has to be recalculated after each iteration when the state estimate has changed [17]. The optimization method can be applied even when the state changes are significant. Finally, the EKF must store the mean and variance of the state values, as well as maintain and update them. A large number of road signs, for example, in visual SLAM can increase the storage volume considerably. Therefore, EKF is not suitable for such scenarios. MSCKF [13] missed details in its observations when working in nonlinearized models, which resulted in information loss, such as inaccurate continuous-time correlations when performing pose estimation and inaccurate estimations of the sensor bias when handling IMU measurements alone. ORB-SLAM2 [14] and VINS-Mono [15] have high requirements for the initial values of the variables, poor robustness and accuracy for low texture environments, and are overly sensitive to dynamic movements (e.g., track lost when rotating). Under a constant speed, the IMU does not have an objective degree of freedom, causing drift. Most importantly, it is more dependent on a closed loop, and its performance degrades more after closing the loop. The significant challenges are the acceleration of feature point extraction, processing objects in a dynamic environment, and the optimization and loopback of the trajectory for front-end data. These will be discussed in this paper.

A visual odometer is easily affected by illumination changes in terms of attitude estimation. With the development of multi-line LiDAR, odometers using LiDAR equipment have become a focus of research. Due to its robustness to illumination changes [18], LiDAR has been widely used for unmanned navigation. A laser odometer includes direct matching as well as feature-based (such as LOAM [19]), and multi-sensor fusion (such as LIO-SAM [20]). LOAM employs novel feature extraction methods (e.g., edge points and plane points). It uses timestamps for motion compensation and incorporates scan-to-scan odometry and map-to-map comparisons. However, there is no back-end optimization, and it cannot handle large-scale rotational transformations. The direct matching algorithms have included the iterative closest point (ICP) [21] algorithm and the normal distribution transform (NDT) [22] point cloud registration algorithm. The ICP algorithm is a classical data registration algorithm that is based on a simple principle and has good accuracy. However, ICP relies on the initial position of the registration point cloud and has a slow calculation speed due to the iterative operation. If the initial value is incorrect, the algorithm may rely on local optimum [23]. The NDT registration algorithm converts the reference point cloud into a normal distribution of multidimensional variables for registration. In the process, it does not use the feature calculation and matching of corresponding points, which reduces the time and resource costs involved. Moreover, this algorithm has little correlation with the initial value [24]. However, NDT does not work in degraded scenes and cannot remove the noise generated by dynamic obstacles in the map. Furthermore, it has no loopback detection, which can result in drift in large environments. The algorithm proposed in this paper performed a depth alignment of feature points by integrating

laser and visual odometry and compensated for drift by incorporating re-localization and loopback detection.

As shown in Figure 1, the moving track of the quadruped robot is not on a plane. It also needs to maintain its balance through relatively intense movements such as lateral movement and autochthonous rotation [25]. These factors have led to drift in IMU data, blurred camera images, matching errors, feature point loss, and pose calculation failures due to excessive motion speed, as well as other problems [26].



**Figure 1.** The moving track of quadruped robot.

To ensure high stability and strong robustness in odometer-based location data for quadruped robots, nonlinear optimization is combined with point cloud data obtained by 3D LiDAR, the image characteristics obtained from binocular vision and IMU inertial data, and loopback detection is used for repositioning to eliminate any cumulative error. In addition, this study proposed a tightly coupled LiDAR vision inertia odometer (LVIO) algorithm that could effectively resolve the errors caused by uniform motion models in a LiDAR odometer, such as mismatching, feature loss, pose calculation failures, and data drift, among others. Precise positioning with strong robustness is achieved for quadruped robots, both indoors and outdoors.

This paper is arranged as follows. Section 2 introduces the LVIO algorithm and describes the algorithmic flow of the proposed method. In Section 3, the experimental platform and environment are briefly introduced, followed by the calibration of relevant sensors, and then the experiments were conducted and analyzed. Section 4 discusses the experimental results. Finally, Section 5 summarizes the experiment and conclusions as well as provides direction for future research.

## 2. Method

In this section, we introduce the tightly coupled LiDAR visual inertia odometer algorithm. Specifically, the front-end data processing algorithm will be introduced, including feature extraction and tracking image data acquired by binocular camera through optical flow method, pre-integration of acceleration and angular velocity data from IMU, and alignment processing of LiDAR point cloud data. The current state quantity, including position ( $p$ ), speed ( $v$ ), and attitude ( $q$ ), can be obtained. Then the tightly coupled back-end nonlinear optimization is performed using a sliding window, and the optimization variables are derived by minimizing the edge, visual, and IMU residuals to obtain  $p v q$  for all frames of the sliding window. Finally, loopback detection and repositioning are performed, and the LiDAR odometry, visual odometry, and IMU pre-integration constraints, as well as the loopback detection constraint are jointly optimized for the global poses.

### 2.1. Optical Flow Feature Tracking

The front-end of visual odometer (VO) adopts a Lucas–Kanade (LK) optical flow for feature tracking [27]. Optical flow is the projection of a moving object, which describes the

moving direction and speed of the corresponding pixels of the object over time, and can be used to track the movement of pixels in an image.

1. On the premise of constant brightness, short distance movement, and spatial consistency (similar motion of adjacent pixels), the brightness of a pixel point on the image is constant at the moment, and the basic equation of optical flow can be obtained as follows:

$$I(x(t), y(t), t) = c \quad (1)$$

2. Let the increments of coordinates be  $d_x$ ,  $d_y$ , and the increments of time be  $d_t$ , we attain the following:

$$I(x, y, t) = I(x + d_x, y + d_y, t + d_t) \quad (2)$$

3. With small motion, the position does not change as much with time; then the Taylor series of the image at  $I(x, y, t)$  is as follows:

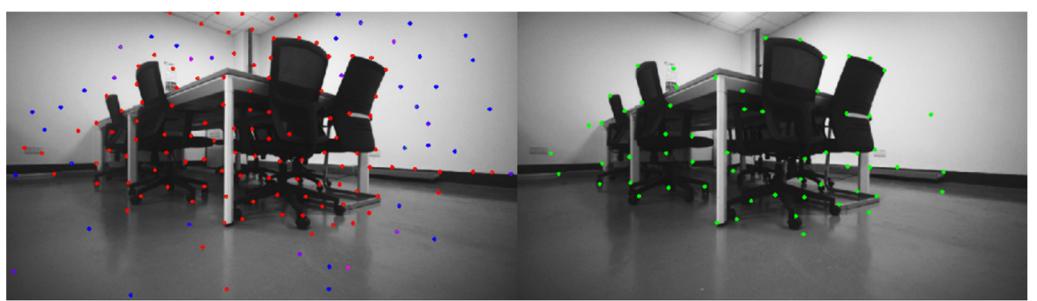
$$\begin{aligned} I(x, y, t) &= I(x, y, t) + I_x d_x + I_y d_y + I_t d_t \\ I_x u + I_y v + I_t &= 0 \end{aligned} \quad (3)$$

where  $I_x$  and  $I_y$  represent the image gradient,  $I_t$  is the time gradient,  $u = d_x/d_t$ , and  $v = d_y/d_t$  represents the horizontal and vertical velocity components of the optical flow at the pixel point.

4. The least squares method is used to solve the basic optical flow equation for all the pixels in the neighborhood, so that the movement of each pixel in the time interval  $\Delta_t$  of the two frames can be calculated. The formula is as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n I_{ix}^2 & \sum_{i=1}^n I_{ix} I_{iy} \\ \sum_{i=1}^n I_{ix} I_{iy} & \sum_{i=1}^n I_{iy}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^n I_{ix} I_t \\ -\sum_{i=1}^n I_{iy} I_t \end{bmatrix} \quad (4)$$

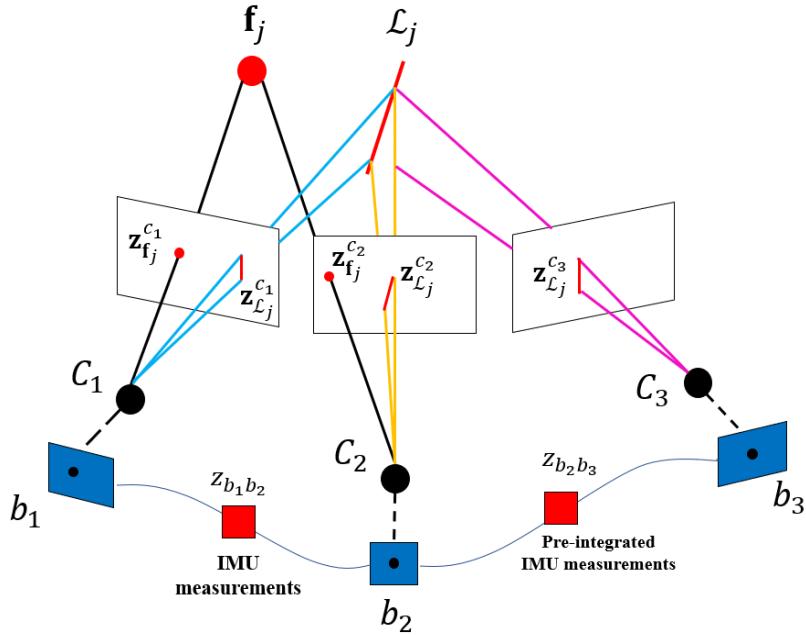
The effect of using LK optical flow algorithm to track the image feature points is shown in Figure 2.



**Figure 2.** Feature tracking by optical flow method for binocular camera (where the green points represent the characteristic points of tracking, the red points represent the low tracking times, and the blue points represent the high tracking times).

## 2.2. IMU Pre-Integration

IMU pre-integration adopts the median method, that is, the pose of two adjacent moments is calculated by means of the average value of IMU measurements at two moments [28]. The camera-IMU model is shown in Figure 3.



**Figure 3.** Camera-IMU model.

The traditional IMU recursive calculation method obtains the state quantity  $p v q$  at the current moment by using the measured linear acceleration  $\bar{a}$  and angular velocity  $\bar{\omega}$  (the real values are  $a$  and  $\omega$ ) through integral operation [29].

1. Calculation formula of IMU measurement is as follows:

$$\begin{aligned}\bar{\omega} &= \omega^b + b^g + n^g \\ \bar{a} &= q_{bw}(a^w + g^w) + b^a + n^a\end{aligned}\quad (5)$$

where  $w$  is the world coordinate system, superscript  $b$  is the IMU coordinate system,  $q_{bw}$  represents the rotation quaternion (from world to IMU coordinates),  $g$  represents the gravity vector, and  $b$  and  $n$  denote bias and noise of IMU accelerometer and gyroscope, respectively.

2. The derivative of  $p v q$  with respect to time can be written as follows:

$$\begin{aligned}\dot{p}_{b_t} &= v_t^w \\ \dot{v}_t &= a_t^w \\ \dot{q}_{wb_t} &= q_{wb_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\omega^{b_t} \end{bmatrix}\end{aligned}\quad (6)$$

3. By integrating the measured value of IMU for the state quantity at the  $i$  moment, the value at the  $j$  moment is as follows:

$$\begin{aligned}p_{b_j}^w &= p_{b_i}^w + v_i^w \Delta t + \int_{t \in [i,j]} (q_{wb_t} a^{b_t} - g^w) \delta t^2 \\ v_j^w &= v_i^w + \int_{t \in [i,j]} (q_{wb_t} a^{b_t} - g^w) \delta t \\ q_{wb_j} &= \int_{t \in [i,j]} q_{wb_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\omega^{b_t} \end{bmatrix} \delta t\end{aligned}\quad (7)$$

4. In VIO based on nonlinear optimization, to avoid repeated integration when absolute poses are optimized, IMU needs to be pre-integrated [30], that is, the integration model is converted into a pre-integration model, and the formula is as follows:

$$q_{wb_t} = q_{wb_i} \otimes q_{b_i b_t} \quad (8)$$

5. When  $q_{wb_i}$  is separated from  $q_{wb_t}$  outside the integral operation, the integral term of the attitude quaternion in the integral formula of  $p v q$  becomes the attitude  $q_{b_i b_t}$  relative to the  $i$  moment. According to the acceleration and angular velocity obtained by IMU sensor,  $p_t v_t q_t$  can be obtained by continuous integration on the basis of initial  $p_0 v_0 q_0$ . Equation (7) can be written as follows:

$$\begin{aligned} p_{b_j}^w &= p_{b_i}^w + v_t^w \Delta t - \frac{1}{2} g^w \Delta t^2 + q_{wb_t} \iint_{t \in [i, j]} \left( q_{b_i b_t} a^{b_t} \right) \delta t^2 \\ v_j^w &= v_i^w - g^w \Delta t + q_{wb_t} \int_{t \in [i, j]} \left( q_{b_i b_t} a^{b_t} \right) \delta t \\ q_{wb_j} &= q_{wb_i} \int_{t \in [i, j]} q_{b_i b_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \omega^{b_t} \end{bmatrix} \delta t \end{aligned} \quad (9)$$

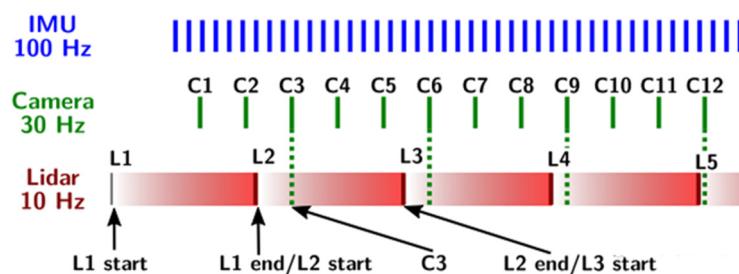
6. In the process of each optimization iteration, the attitude adjustment is adjusted relative to the world coordinate system, that is,  $q_{wb_j}$  is adjusted outside the integral while  $q_{b_i b_t}$  is relatively unchanged, so as to avoid the problem of repeated calculation of integral. In Equation (9):

$$\begin{aligned} \alpha_{b_i b_j} &= \iint_{t \in [i, j]} \left( q_{b_i b_t} a^{b_t} \right) \delta t^2 \\ \beta_{b_i b_j} &= \int_{t \in [i, j]} \left( q_{b_i b_t} a^{b_t} \right) \delta t \\ q_{b_i b_j} &= \int_{t \in [i, j]} q_{b_i b_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \omega^{b_t} \end{bmatrix} \delta t \end{aligned} \quad (10)$$

we find the pre-integral of IMU, corresponding to position ( $p$ ), velocity ( $v$ ), and attitude ( $q$ ) respectively.

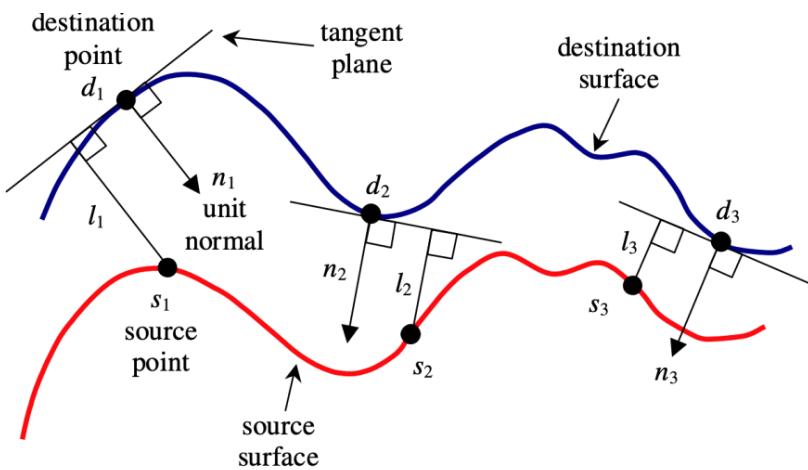
### 2.3. LiDAR Point Cloud Registration

Before registration, feature primitives were extracted from the LiDAR point cloud and synchronized with camera frames and IMU timestamps to optimize all sensors at once [31]. The processing includes point cloud distortion removal, timestamp synchronization, filtering, primitive extraction, and tracking. The motion distortion compensation and timestamp synchronization of the laser frame point cloud are shown in Figure 4.



**Figure 4.** Motion distortion compensation and timestamp synchronization of laser frame point cloud.

The point cloud registration adopts ICP algorithm to transform the point convergence into the point convergence with similar local geometric features. The traditional point-to-point ICP algorithm [32] has the following disadvantages: noise or abnormal data may lead to algorithm convergence or error. The selected initial iteration value will have an important influence on the final registration result, and if the initial iteration value is not properly selected, the algorithm may be limited to the local optimum. Point-to-plane ICP algorithm [33], as shown in Figure 5, uses the distance from the measuring point to the plane as the objective function. It converges faster than point-to-point and approximates nonlinear problems through linear optimization.



**Figure 5.** Point-to-plane ICP algorithm.

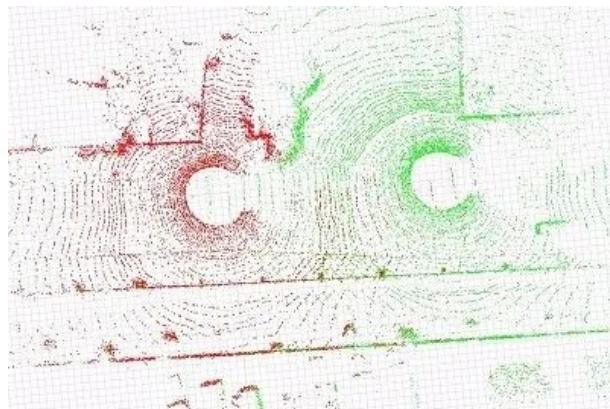
As shown in the figure, two points converged (source surface and destination surface), source point  $s_i$  and target point  $d_i$ , as well as target plane.  $n_i$  is the normal vector of the target plane, and  $l_i$  is the distance from the source point to the target plane. The objective function is shown as follows:

$$M_{opt} = \operatorname{argmin}_M \sum ((M * s_i - d_i) * n_i)^2 \quad (11)$$

where  $M$  is a  $4 \times 4$  3D rigid body transformation matrix composed of rotation ( $R$ ) and shift ( $T$ ), which is shown as follows:

$$M = T(t_x, t_y, t_z)R(\alpha, \beta, \gamma) \quad (12)$$

Point cloud data were collected by 3D LiDAR. We calculate a point-to-plane ICP algorithm. Red was the source point clustering, and green was the target point clustering. Algorithm results are shown in Figure 6.



**Figure 6.** Point-to-plane ICP algorithm.

After point cloud registration, a sparse depth map is created. By stacking several frames together, a dense depth map can be obtained for the visual feature points [34].

#### 2.4. Tightly Coupled Nonlinear Optimization

As opposed to the loosely coupled method, which assumes image processing as a black box and integrates IMU data until visual odometry (VO) is calculated, the tightly coupled method adds image feature information into the state vector. The filter-based tightly coupled VIO state vector has a lower dimension as its calculation method is relatively simple. The calculation consisted of two parts: one predicts and updates the status data,

and the other measures the data from the other sensors. The positioning accuracy is relatively low [35].

The tightly coupled back-end nonlinear optimization based on sliding windows can achieve high accuracy. An objective function is used to unify visual constraints, IMU constraints, and closed-loop constraints to implement bundle adjustment (BA), minimize edge residuals, visual residuals, and IMU residuals, and calculate the Jacobian matrix by differentiating the optimization variables. The bias, external parameters from IMU to camera, and  $p_{vq}$  of all frames in the sliding window were obtained.

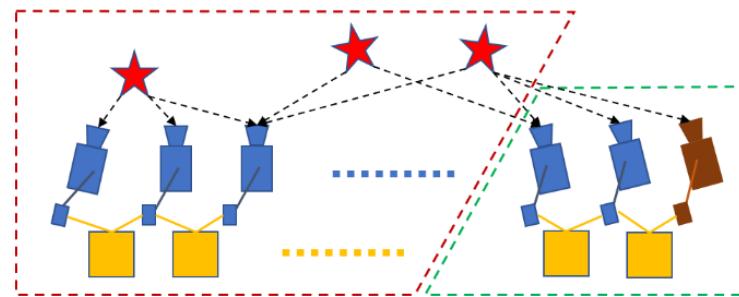
$$\min_{\mathcal{X}} \left\{ \|r_p - H_p \mathcal{X}\|^2 + \sum_{k \in \mathcal{B}} \left\| r_B(\hat{z}_{b_{k+1}}^{b_k}, \mathcal{X}) \right\|_{p_{b_{k+1}}}^2 + \sum_{(l,j) \in \mathcal{C}} \rho \left\| r_C(\hat{z}_l^{c_j}, \mathcal{X}) \right\|_{p_l^{c_j}}^2 \right\} \quad (13)$$

The above formula consists of three objective items, namely, marginalization residual, IMU measurement residual, and visual reprojection error.  $\mathcal{X}$  is all states in the sliding window.  $x_k$  is the IMU state ( $p_{vq}$  and acceleration bias  $b_a$ , gyroscope bias  $b_g$ ) captured in the image of frame  $k$ .  $x_c^b$  is the external parameter of the camera, which is expressed as follows:

$$\begin{aligned} \mathcal{X} &= [x_0, x_1, \dots, x_n, x_c^b, \lambda_0, \lambda_1, \dots, \lambda_m] \\ x_k &= [p_{b_k}^w, v_{b_k}^w, q_{b_k}^w, b_a, b_g], k \in [0, n] \\ x_c^b &= [p_c^b, q_c^b] \end{aligned} \quad (14)$$

where  $n$  is the number of key frames, and  $m$  is the total number of all observed landmark points in the sliding window.  $r_p$  in the optimization term is the marginal residual, and  $H_p$  is the new information matrix after Schur complement, where  $p$  refers to prior, that is, the measurement information is transformed into prior information.

1. Marginalization residual: Briefly, to remove the information of pose and feature point constraints in the sliding window and retain the image information constraints for optimization through Ceres Solver (nonlinear optimization library) [36], as shown in Figure 7.



**Figure 7.** Schematic diagram of sliding window marginalization. represents the old key frame, represents the latest frame, represents IMU constraint, represents visual feature, represents fixed state, and represents estimated state.

Marginalization ensures the sparsity of the system and the continuity of pre-integration, ensures enough parallax between key frames, and can triangulate enough feature points.

2. IMU measurement residual: It is generated by IMU data between adjacent frames, including state propagation prediction and residuals of IMU pre-integration, namely  $r_B$  in the optimization term. Optimization variables are IMU state ( $p_{vq}$  and bias). The calculation of  $r_B$  is shown as follows:

$$r_B(z_{b_{k+1}}^{b_k}, \mathcal{X}) = \begin{bmatrix} \delta\alpha_{b_k}^{b_k} \\ \delta\beta_{b_k}^{b_k} \\ \delta\theta_{b_k}^{b_k} \\ \delta b_a \\ \delta b_g \end{bmatrix} = \begin{bmatrix} R_w^{b_k} \left( p_{b_{k+1}}^w - p_{b_k}^w + \frac{1}{2}g^w \Delta t_k^2 - v_{b_k}^w \Delta t_k \right) - \alpha_{b_{k+1}}^{b_k} \\ R_w^{b_k} \left( v_{b_{k+1}}^w + g^w \Delta t_k - v_{b_k}^w \right) - \beta_{b_{k+1}}^{b_k} \\ 2 \left[ q_{b_k}^{w^{-1}} \otimes q_{b_{k+1}}^w \otimes (\hat{\gamma}_{b_{k+1}}^{b_k})^{-1} \right]_{xyz} \\ b_{ab_{k+1}} - b_{ab_k} \\ b_{wb_{k+1}} - b_{wb_k} \end{bmatrix} \quad (15)$$

3. Visual residual: The visual reprojection error of the feature point in the sliding window is the error between the observed value and the estimated value of the same landmark point in the normalized camera coordinate system, namely  $r_C$  in the optimization term. The calculation formula is as follows:

$$r_C = \left( \frac{x}{z} - u, \frac{y}{z} - v \right)^T \quad (16)$$

The feature quantity to be estimated is the three-dimensional space coordinate  $(x, y, z)^T$  of the feature point, and the observed value  $(u, v)^T$  is the coordinate of the feature point under the camera normalized plane. Since the observation depth of feature points may be large, it is difficult to optimize, so inverse depth parameterization is performed to align with the Gaussian system and reduce the parameter variables of actual optimization. Coordinate relations of feature points in the normalized camera coordinate system are as follows:

$$(x, y, z)^T = \frac{1}{\lambda} (u, v, 1)^T \quad (17)$$

where  $\lambda = 1/z$  is called inverse depth.

4. Jacobian residual matrix: The value of the feature point in frame  $i$  projected to the camera coordinate system in frame  $j$  is:

$$\begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = T_{bc}^{-1} T_{wb_j}^{-1} T_{wb_i} T_{bc} \begin{bmatrix} \frac{1}{\lambda} u_{c_i} \\ \frac{1}{\lambda} v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \quad (18)$$

Its three-dimensional coordinate form is:

$$f_{c_j} = \begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \end{bmatrix} = R_{bc}^\top R_{wb_j}^\top R_{wb_i} R_{bc} \frac{1}{\lambda} \begin{bmatrix} u_{c_i} \\ v_{c_i} \\ 1 \end{bmatrix} + R_{bc}^\top \left( R_{wb_j}^\top \left( (R_{wb_i} P_{bc} + P_{wb_i}) - P_{wb_j}^\top - P_{wb_i} \right) - P_{bc} \right) \quad (19)$$

To solve the Jacobian matrix is to differentiate the above state variables by visual residuals. The derivative of visual residual to reprojected 3D point  $f_{c_j}$  is as follows:

$$\frac{\partial r_c}{\partial f_{c_j}} = \begin{bmatrix} \frac{1}{z_{c_j}} & 0 & -\frac{x_{c_j}}{z_{c_j}^2} \\ 0 & \frac{1}{z_{c_j}} & -\frac{y_{c_j}}{z_{c_j}^2} \end{bmatrix} \quad (20)$$

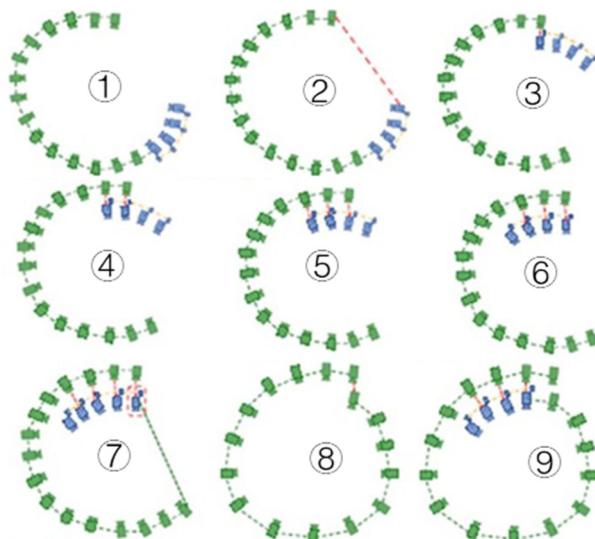
The derivative of each optimization variable of reprojected 3D point  $f_{c_j}$  is as follows:

$$\begin{aligned} J[0]^{3 \times 7} &= \left[ \frac{\partial f_{c_j}}{\partial p_{b_i}^w}, \frac{\partial f_{c_j}}{\partial q_{b_i}^w} \right] \\ J[1]^{3 \times 7} &= \left[ \frac{\partial f_{c_j}}{\partial p_{b_j}^w}, \frac{\partial f_{c_j}}{\partial q_{b_j}^w} \right] \\ J[2]^{3 \times 7} &= \left[ \frac{\partial f_{c_j}}{\partial p_c^b}, \frac{\partial f_{c_c}}{\partial q_c^b} \right] \\ J[3]^{3 \times 1} &= \frac{\partial f_{c_j}}{\partial \lambda_l} \end{aligned} \quad (21)$$

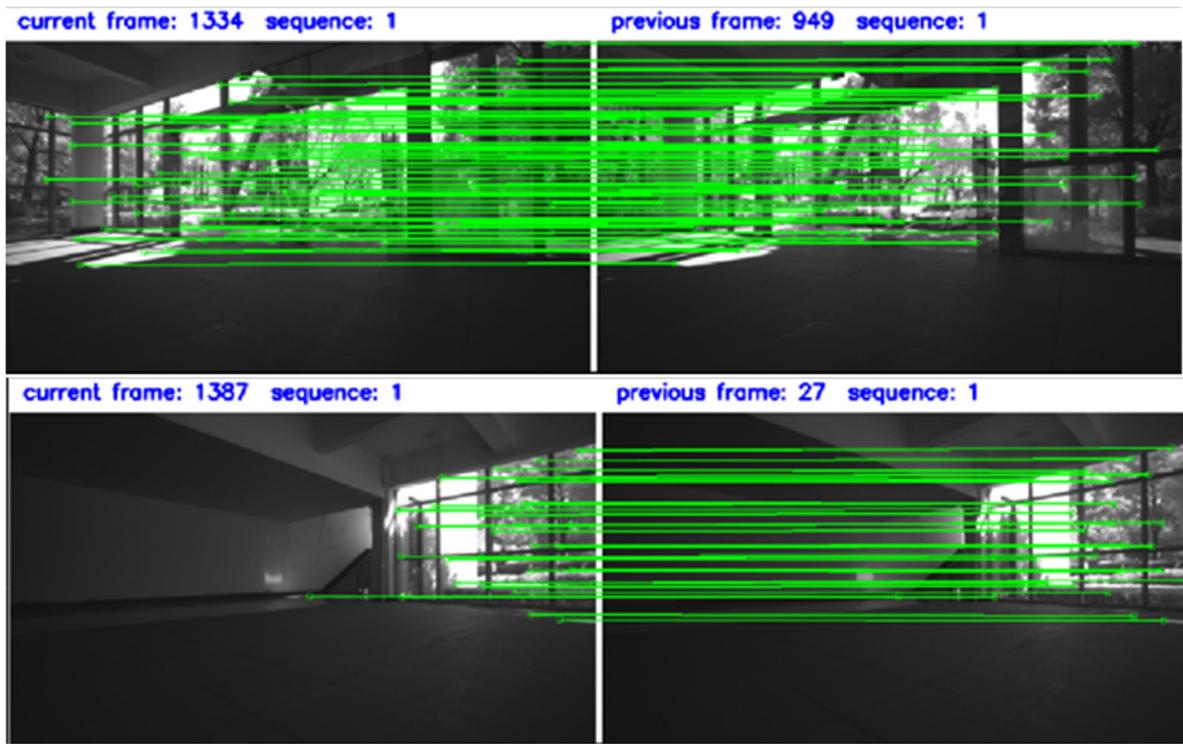
## 2.5. Relocation

Even though marginalization and sliding windows reduce computational complexity, the cumulative drift errors of the system are still inevitable where the errors are the global position (XYZ coordinates) and yaw angle (rotation around the direction of gravity). The purpose of introducing relocation is to move the local sliding window to align the current pose with a past pose [37]. Specific steps include loopback detection and feature matching between loopback candidate frames and tightly coupled relocation, as shown in Figure 8.

1. First, visual word bag position recognition method (DBoW2) [38] is used for loop detection. During this period, only pose estimation is performed (blue part), and the past state (green part) is always recorded. After time and space consistency test, the visual word bag returns loopback detection candidate frames, and the red dotted line in the figure represents loopback association.
2. If loopback is detected in the latest frame, the multi-constraint relocation is initiated, the corresponding relationship is matched by BRIEF descriptors [39], and the outer points are removed by two-step geometric elimination method (2D–2D, 3D–2D) [40]. When the inner point exceeds a certain threshold, the candidate frame is regarded as the correct loopback detection and relocation is performed. Feature matching of loopback candidate frames is shown in Figure 9.



**Figure 8.** Loopback detection and feature matching between loopback candidate frames and tightly coupled relocation.



**Figure 9.** Feature matching between loopback candidate frames.

As shown in Figure 9, loopback is detected in frame 1334 and frame 1387, relocation is started, and feature matching of loopback candidate frame is performed in frame 949 and frame 27, respectively. The matching effect is good without feature matching failure. The specific calculation method is as follows:

$$\begin{aligned} \min_{\mathcal{X}} & \left\{ \|r_p - H_p \mathcal{X}\|^2 + \sum_{k \in \mathcal{B}} \left\| r_B \left( \hat{\mathbf{z}}_{b_k}, \mathcal{X} \right) \right\|_{p_{b_k+1}}^2 + \sum_{(l,j) \in \mathcal{C}} \rho \left( \left\| r_C \left( \hat{\mathbf{z}}_l, \mathcal{X} \right) \right\|_{p_{l_j}^{c_j}}^2 \right) \right. \\ & \left. + \sum_{(l,v) \in \mathcal{L}} \rho \left( \left\| r_C \left( \hat{\mathbf{z}}_l, \mathcal{X}, q_v^w, p_v^w \right) \right\|_{p_i^{cv}}^2 \right) \right\} \end{aligned} \quad (22)$$

The difference between the above formula and the previous tightly coupled nonlinear optimization model lies in the addition of the loopback term, that is, the loopback frame attitude obtained from the pose diagram.

3. Finally, key frames were added to the pose map, and the global 4-DOF (coordinate XYZ and yaw angle) pose optimization of the past pose and closed loop image frames are performed by minimizing the cost function:

$$\min_{p, \psi} \left\{ \sum_{(i,j) \in \mathcal{S}} \|r_{i,j}\|^2 + \sum_{(i,j) \in \mathcal{L}} \rho \left( \|r_{i,j}\|^2 \right) \right\} \quad (23)$$

where  $\mathcal{S}$  is the set of all sequential edges (the relative transformation between two key frames in the local slide window directly obtained from VIO),  $\mathcal{L}$  is the set of all loopback edges (the connection between the latest marginalized key frame  $i$  with loop connection and the previous key frame  $j$ ), and another Huber norm  $\rho(\cdot)$  is added to the loopback

edge. The impact of error loops can be further reduced.  $r_{i,j}$  represents the minimal residual between  $i$  and  $j$ :

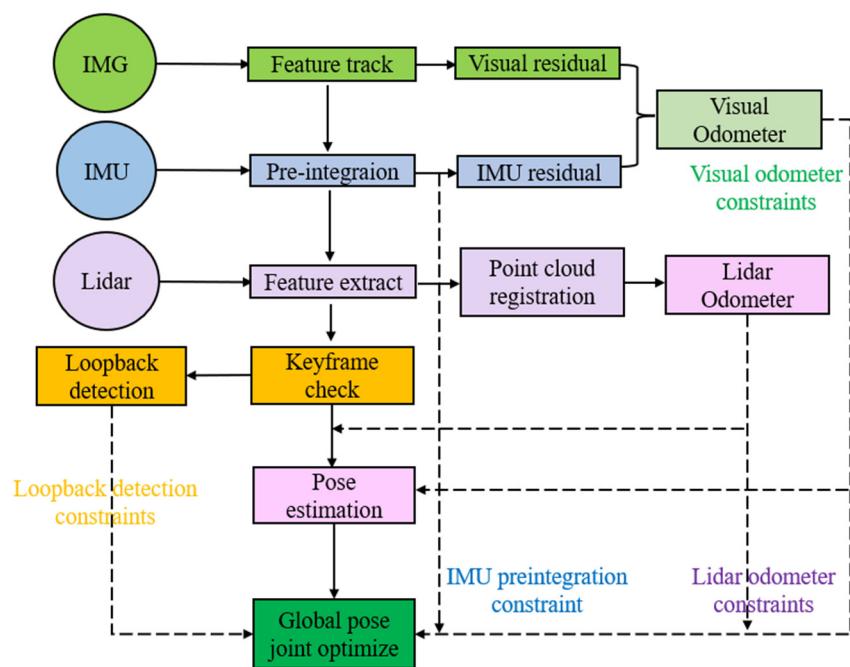
$$r_{i,j}(p_i^w, \psi_i, p_j^w, \psi_j) = \begin{bmatrix} R(\hat{\phi}_i, \hat{\theta}_i, \psi_i)^{-1}(p_j^w - p_i^w) - p_{ij}^i \\ \psi_j - \psi_i - \hat{\psi}_{ij} \end{bmatrix} \quad (24)$$

where  $\hat{\phi}_i$  and  $\hat{\theta}_i$  are the estimation of roll angle and pitch angle obtained directly from VIO, respectively.

## 2.6. Algorithm Flow

The flow of LiDAR vision inertial odometer (LVIO) algorithm based on tight coupling optimization is shown in Figure 10. The program inputs are image data obtained by binocular camera, IMU data, and point cloud data obtained by LiDAR. The specific algorithm flow is as follows:

1. The input image data is used for feature point tracking by optical flow method, that is, the motion of the previous frame is used to estimate the initial position of feature points in the current frame. The positive and negative matching of both eyes was performed by LK optical flow method, and its error must be less than 0.5 pixel; otherwise, it is regarded as outlier. Then, the fundamental matrix ( $F$ ) is calculated by using tracking points, and outliers are further removed using polar constraint [41]. If feature points are insufficient, corner points are used to replace them, and the tracking times of feature points are updated. Then the normalized camera coordinates of feature points and the moving speed relative to the previous frame are calculated. Finally, the feature point data of the current frame (including normalized camera plane coordinates, pixel coordinates, and normalized camera plane moving speed) are saved.
2. After IMU data is input, IMU median integration is performed to calculate the current frame pose and update  $p_{vq}$ . The specific steps are using the IMU data between the previous frame and the current frame; if there is no initialization, the IMU acceleration between this frame is averaged, and the gravity is aligned to obtain the initial IMU attitude. Then the pose of the previous frame is applied it to IMU integral to obtain the pose of the current frame.
3. The LiDAR odometer is obtained by extracting point cloud features and matching them with visual features. The feature graph is optimized in real time by sliding window.
4. The estimation results of the laser odometer were used for initialization, and the depth information of the visual features was optimized using the LiDAR measurement results. The visual odometer was obtained by minimizing the visual residuals, IMU measurement residuals and marginalization residuals, and the pose estimation was performed.
5. Key frame check: The criteria for confirming key frames is that there are many new feature points or the parallax of feature points in the first two frames is large enough [42]. After the keyframe check, the visual word bag method is used for loop detection.
6. Global pose joint optimization: The whole state estimation task of the system is expressed as a maximum estimation posterior probability problem, and the global pose joint optimization is performed by IMU pre-integral constraint, visual odometer constraint, radar odometer constraint, and loopback detection constraint.

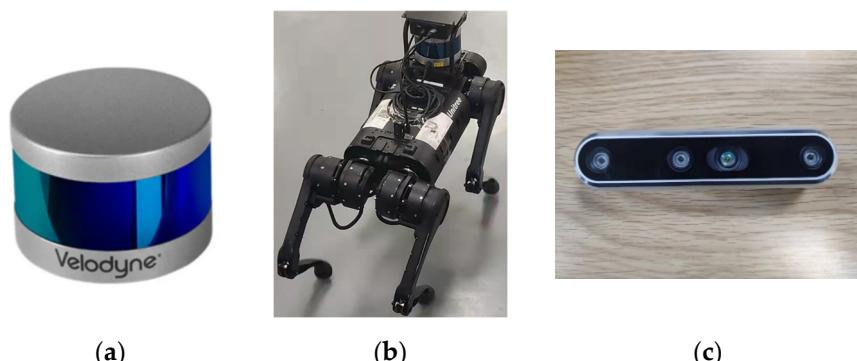


**Figure 10.** LiDAR vision inertial odometer algorithm flow chart.

### 3. Results

#### 3.1. Experimental Platform and Environment

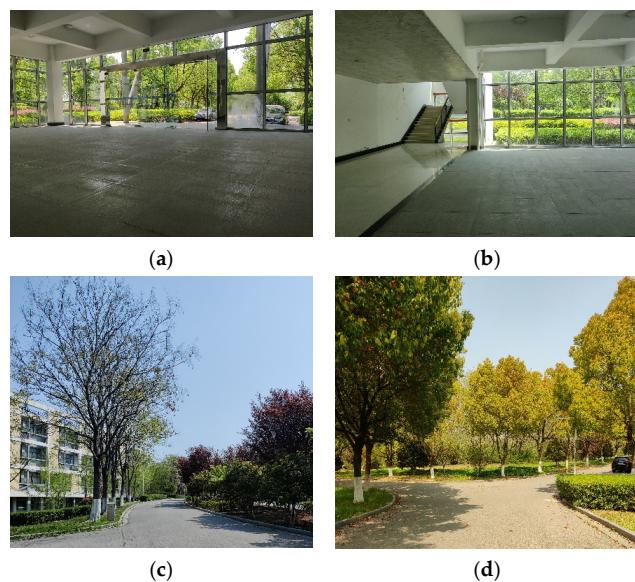
The experimental platform is a quadruped robot in our laboratory. The processor is an embedded controller (Jetson Xavier NX), and the software environment is Ubuntu operating system and Robot Operating System (ROS). The LiDAR used is Velodyne 16-line three-dimensional LiDAR VLP-16. The camera used is Intel sense-depth camera (RealSense D455). They are shown in Figure 11.



**Figure 11.** Experimental platform, (a) VLP-16 LiDAR, (b) quadruped robot, and (c) RealSense D455 camera.

Among them, the image of the RealSense D455 camera includes the global shutter and the integrated IMU (BME055). The experimental image and IMU data are processed by time synchronization. The image acquisition frequency is 30 Hz, the resolution is  $848 \times 480$ , and the IMU data acquisition frequency is 100 Hz.

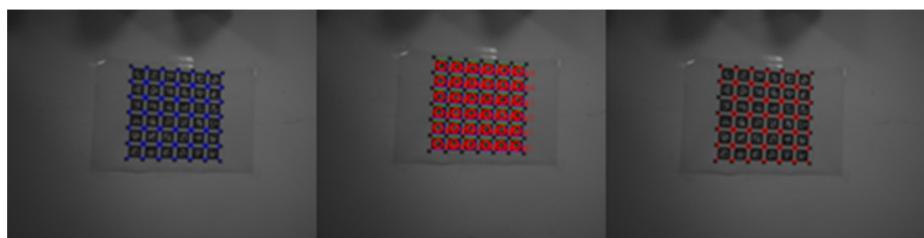
The indoor and outdoor experimental environment is shown in Figure 12.



**Figure 12.** The indoor and outdoor experimental environment: (a,b) the indoor experimental environment and (c,d) the outdoor experimental environment.

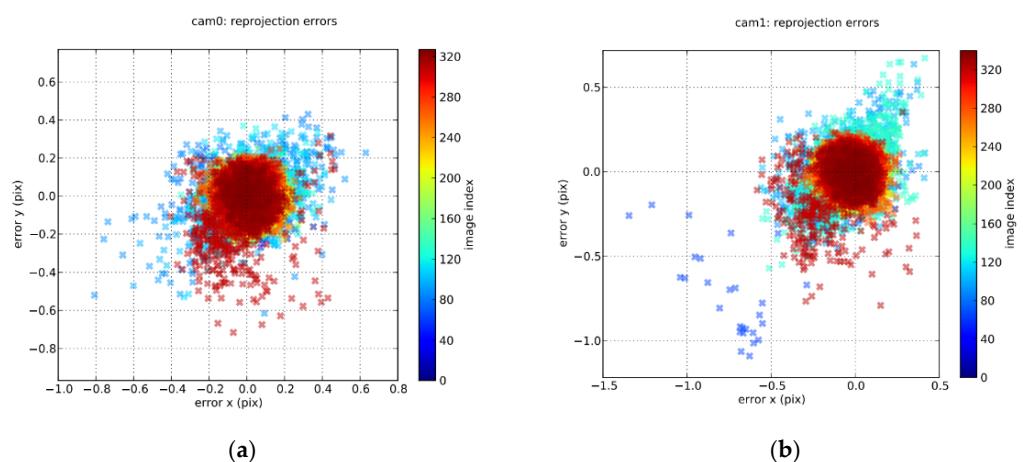
### 3.2. The Sensor Calibration

To obtain the internal parameters, such as the distortion parameters of camera, the noise density, and the random walk parameters of the IMU as well as the rotation and translation matrix of binocular camera to IMU, Kalibr (visual inertial calibration toolbox) is used to perform the joint calibration. The process is shown in Figure 13.



**Figure 13.** Joint calibration process of binocular camera and IMU.

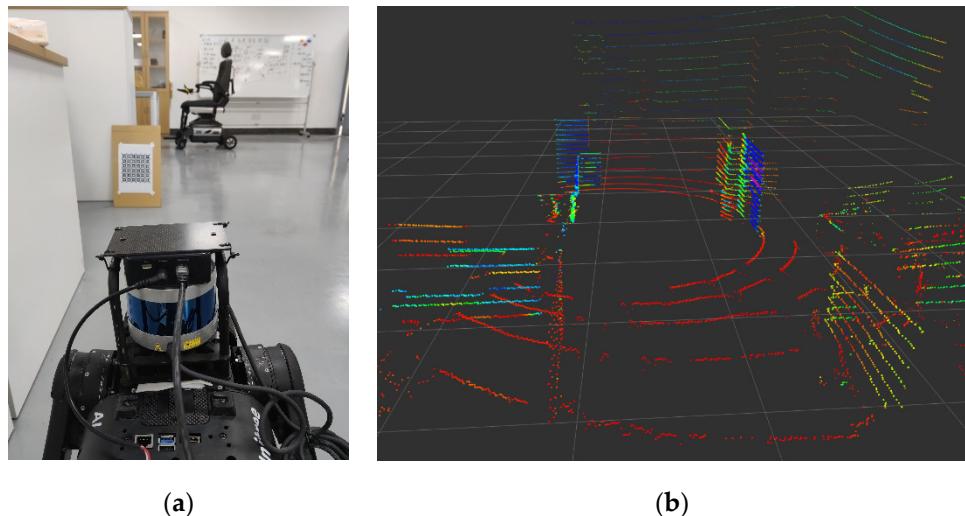
After calibration, the reprojection error of binocular camera is shown in Figure 14.



**Figure 14.** Calibration reprojection error of binocular camera. (a) Reprojection error of the left-eye camera and (b) reprojection error of the right-eye camera.

As shown in the figure, the reprojection errors of both the left and right target are within one pixel; therefore, the calibration effect is good.

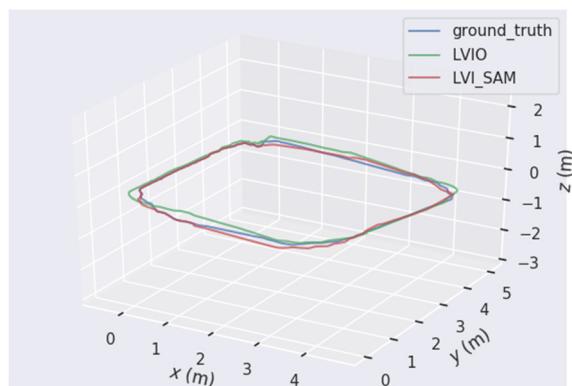
The calibration of the external rotation parameters of the 3D LiDAR and binocular camera is a key factor as the calibration results would directly affect the stability and positioning accuracy of LVIO. In this study, the LiDAR–camera joint calibration is performed using the calibration toolbox Autoware. After calibration, we obtained parameters such as “camera extrinsic mat”, “camera mat”, “distcoeff”, and reprojection error. The calibration process is shown in Figure 15.



**Figure 15.** The LiDAR–camera joint calibration process: (a) the real environment of the calibration and (b) the LiDAR point cloud of the calibration.

### 3.3. The Indoor Experiment

In the indoor environment, the proposed LVIO method is conducted, which involved marching the quadruped robot in the hall for one week. The total distance of the loop is 18 m. At the same time, LVI-SAM [43], which is the latest LiDAR vision inertial odometer calculation method, is used in a comparative experiment. The trajectory is shown in Figure 16.



**Figure 16.** Odometer operating trajectory in indoor environment.

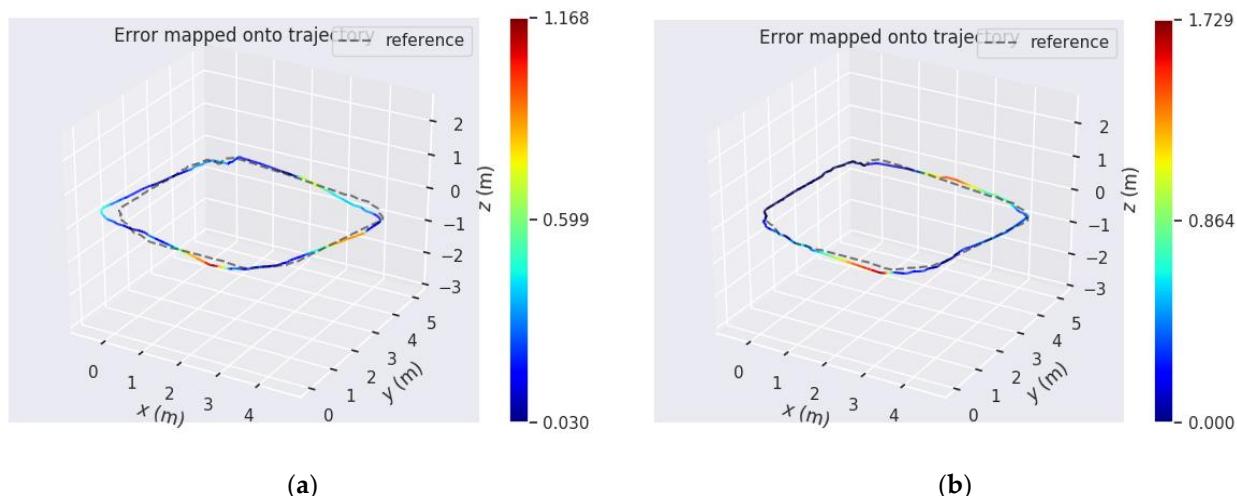
The blue line is the trajectory drawn by a wheeled odometer, the green is drawn by our algorithm, and the red is drawn by LVI-SAM. The trajectory error referred to the absolute error between the estimated trajectory and the output trajectory of the wheeled odometer, which is shown in Table 1.

**Table 1.** The operating trajectory error of the proposed LVIO algorithm and comparison algorithm in indoor environment.

Algorithm	LVIO Algorithm	Comparison Algorithm
Minimum error/m	0.030335	0.000000
Maximum error/m	1.168070	1.728709
Average error/m	0.290409	0.306697
Median error/m	0.219200	0.131016
Root mean square error/m	0.358962	0.523203
Standard error/m	0.210989	0.423884

The above results may have different operating effects in different situations, depending on the operating environment and motion model of the quadruped robot.

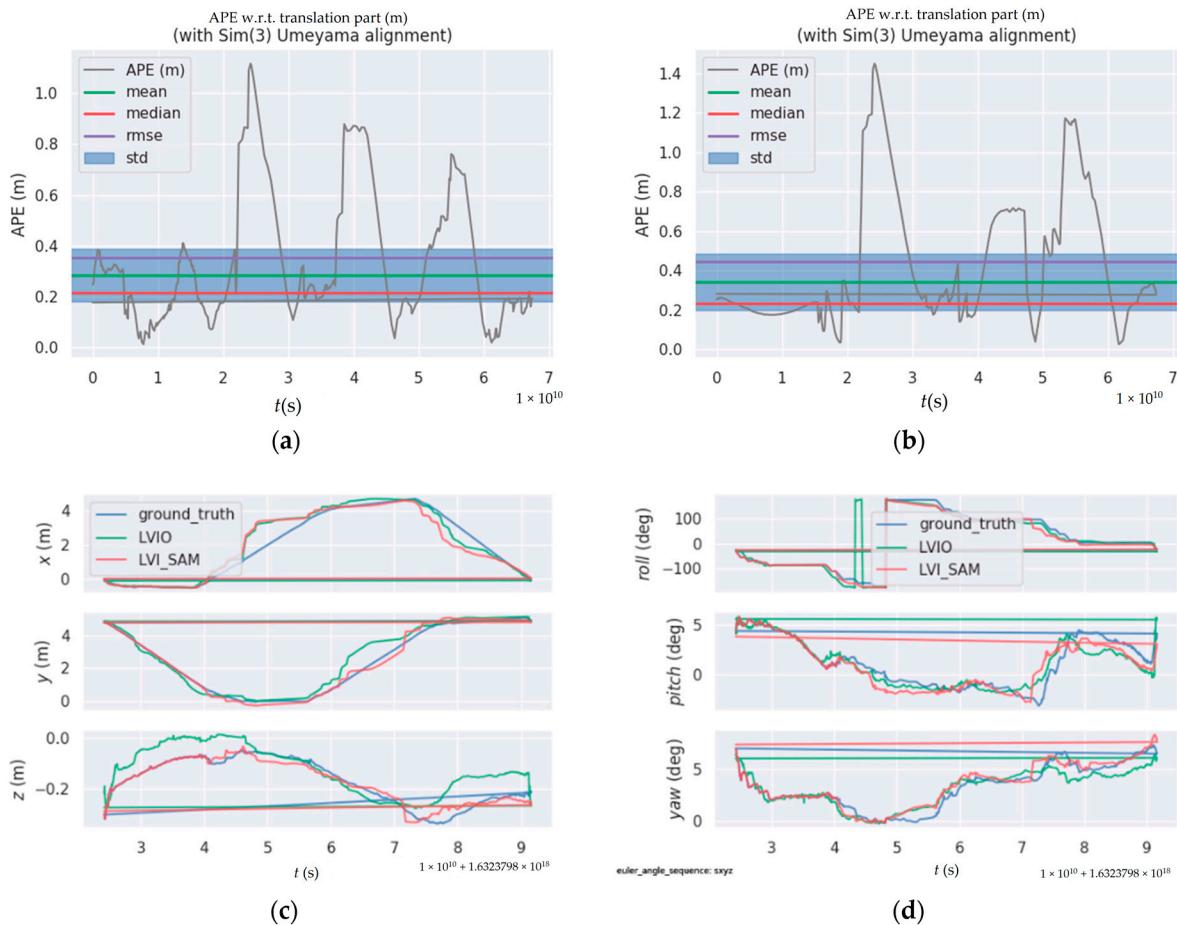
In this study, absolute positional error (APE) is used to measure the accuracy of the trajectory. The track errors of the LVIO algorithm and LVI-SAM algorithm are shown in Figure 17.



**Figure 17.** The track error visualizations of the LVIO algorithm and LVI-SAM algorithm: (a) the trajectory error of the LVIO algorithm and (b) the trajectory error of LVI-SAM algorithm.

The specific mean error, median error, root-mean-square error, standard error, and the error of the trajectory in the x, y, and z directions, as well as the error of the trajectory in pitch, roll, and yaw direction angles, are shown in Figure 18.

Positioning accuracy is an important criterion for the operational effectiveness of the LVIO algorithm. To evaluate the measurement accuracy of the visual inertial odometry, the mean error and root-mean-square error are selected to measure the trajectory error. As shown in Table 1 and Figure 15, the average error between the estimated trajectory and the actual trajectory of the method proposed is 0.290409 m, and the root-mean-square error is 0.358962 m. The average error of the trajectory of the LVI-SAM algorithm is 0.306697 m, and the root-mean-square error is 0.523203 m. The proposed method has a higher localization accuracy. The significant error in the figure is due to the lack of texture on the indoor white wall, which resulted in the deviation of feature matching, as shown in Figure 19.



**Figure 18.** Specific error curve: (a) mean error, median error, RMSE error, and STD error of the proposed method; (b) mean error, median error, RMSE error, and STD error of LVI-SAM algorithm; (c) the error trajectory in the x, y, and z directions; and (d) the error trajectory in pitch, roll, and yaw direction angle.

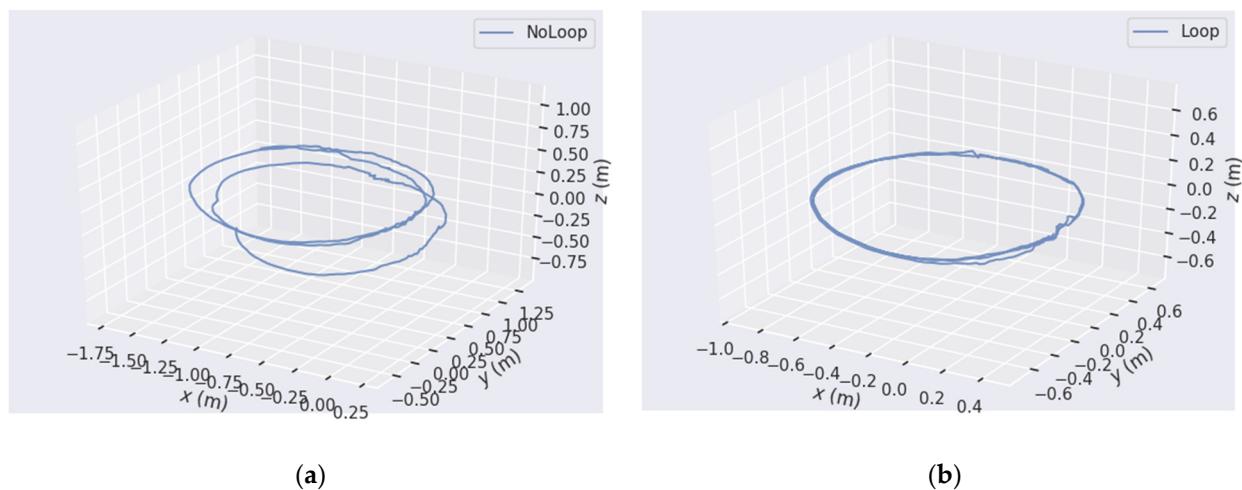


**Figure 19.** The lack of texture in an interior white wall.

In the indoor environment, the light intensity is moderate, and the variation is weak, so our algorithm operates as expected. Though there is some deviation in feature matching caused by the lack of wall texture, the algorithm still shows good robustness and accuracy.

### 3.4. The Loop Detection Experiment

In an indoor environment, the quadruped robot travels several laps around a circular path with a certain radius. The operation effect of the algorithm is shown in Figure 20.

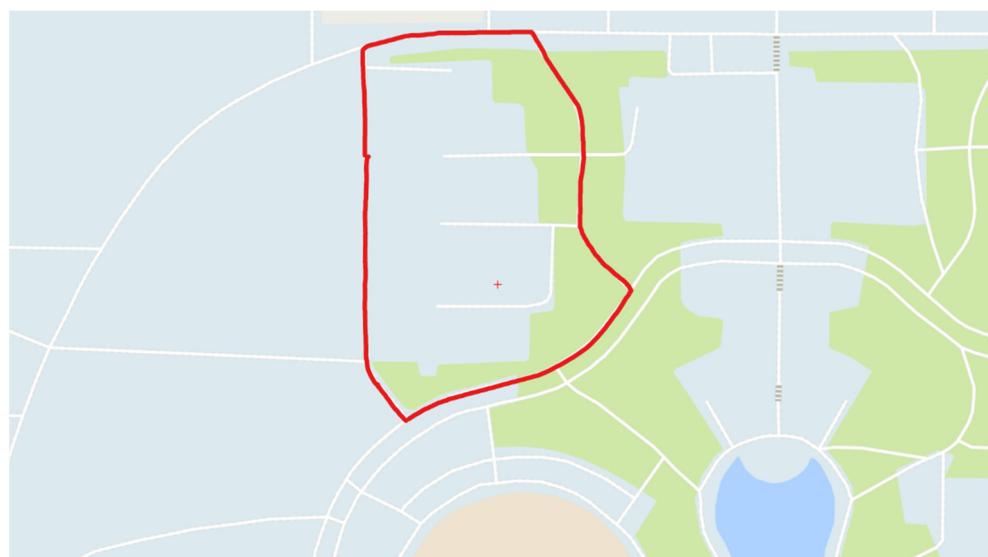


**Figure 20.** LVIO loop experiment running track in indoor environment: (a) before adding loopback detection and (b) after adding loopback detection.

Before loopback detection was added, the trajectory of the quadruped robot deviated from the established path due to the accumulated drift error of the system after several laps. However, after loopback detection is added and the loopback candidate frame matching is performed, the accumulated drift error can always be eliminated in time after the track drift occurs, so as to realize relocation and form a closed loop.

### 3.5. The Outdoor Experiments

To verify the positioning effect of our algorithm when applied to the quadruped robot, relevant experiments need to be performed in an environment with high outdoor lighting intensity and unstable lighting conditions. In the outdoor test, the surrounding environment of the building is used as the experimental object, and the quadruped robot marches around it. Using ROS and Leaflet (open-source interactive map for mobile devices), the trajectory output by our algorithm was displayed in an offline map through a series of coordinate transformations. The total distance of the loop is 550 m. The effect is shown in Figure 21.



**Figure 21.** Outdoor LVIO operation trajectory (the red line).

In the experiment, in addition to the high light intensity and unstable light conditions, there is also the interference of the dynamic environment (e.g., pedestrians and vehicles

passing, etc.). However, the performance of our visual inertial odometer algorithm shows good accuracy and robustness.

#### 4. Discussion

Since the trajectory of a quadruped robot is not in a plane, and the robot must maintain its balance during rigorous motion, the robustness and stability of the visual inertial odometer are required. The LiDAR vision inertial odometer calculation used in this experiment has a good effect on the quadruped robot platform. There is no feature point loss or pose calculation failure occurring under either a static indoor environment or a dynamic outdoor environment. After adding loopback detection and relocation, the track forms a closed loop, which illustrates the improved positioning effect. Although the experiments show that the proposed method works well both indoors or outdoors, we should clearly recognize that there are also some fluctuations in the trajectories of LVIO due to the limitations of sensor accuracy and degraded environments (e.g., no texture, highlight changes, etc.) and dynamic environments. As a result, how to overcome such undesirable phenomenon by improving the accuracy of the sensor and increasing the constraints on the degradation and dynamic environments can be a topic of our future work.

#### 5. Conclusions

In this study, based on nonlinear optimization, point cloud data obtained by LiDAR, image characteristics of binocular vision, and IMU inertial data are effectively integrated, leading to the LVIO algorithm. A loopback detection and repositioning method is also used to eliminate accumulated errors and improve the positioning accuracy and stability. Finally, the average error of indoor positioning is 0.290409 m. In comparison, the positioning error of LVI-SAM [43], which is the latest LiDAR vision inertial odometer calculation method, is 0.306697 m. The experimental results demonstrate the feasibility and effectiveness of the proposed method. The next step is to study the cumulative drift error of IMU and further improve the real-time performance as well as the large range of positioning accuracy in our algorithm. We plan to use the state-of-the-art MEMS inertial sensors (e.g., Xsens) in future research, and we will improve the feature extraction algorithm, either by adding quadtree or fusing point-line features, to ensure the uniformity and robustness of feature extraction. We will integrate the absolute positioning information of GPS and Beidou into the system to compensate for the cumulative errors of relative positioning in large scenes.

**Author Contributions:** Conceptualization, F.G.; Investigation, J.H.; Methodology, F.G., W.T., J.H. and H.C.; Software, F.G., W.T., J.H. and H.C.; Supervision, F.G.; Validation, W.T. and H.C.; Writing—original draft, F.G.; Writing—review and editing, W.T., J.H. and H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the National Natural Science Foundation of China under grants 61873120 and 62073187, the National Natural Science Foundation of Jiangsu Province under grants BK20201469 and BE2021016-5, the Major Program of Natural Science Research of Colleges and Universities in Jiangsu Province under Grant 20KJA510007 and the Qing Lan project of Jiangsu Province.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

- Meng, X.; Wang, S.; Cao, Z.; Zhang, L. A review of quadruped robots and environment perception. In Proceedings of the Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 6350–6356.
- Li, Z.; Ge, Q.; Ye, W.; Yuan, P. Dynamic balance optimization and control of quadruped robot systems with flexible joints. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *46*, 1338–1351. [[CrossRef](#)]
- Prasad, S.K.; Rachna, J.; Khalaf, O.I.; Le, D.-N. Map matching algorithm: Real time location tracking for smart security application. *Telecommun. Radio Eng.* **2020**, *79*, 1189–1203. [[CrossRef](#)]

4. Mowafy, A.; Kubo, N. Integrity monitoring for positioning of intelligent transport systems using integrated RTK-GNSS, IMU and vehicle odometer. *IET Intell. Transp. Syst.* **2018**, *12*, 901–908. [[CrossRef](#)]
5. Li, F.; Tu, R.; Hong, J.; Zhang, H.; Zhang, P.; Lu, X. Combined Positioning Algorithm Based on BeiDou Navigation Satellite System and Raw 5G Observations. *Measurement* **2022**, *190*, 110763. [[CrossRef](#)]
6. Jagadeesh, G.R.; Srikanthan, T. Online map-matching of noisy and sparse location data with hidden Markov and route choice models. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2423–2434. [[CrossRef](#)]
7. Nikolaev, D.; Chetiy, V.; Dudkin, V.; Davydov, V. Determining the location of an object during environmental monitoring in conditions of limited possibilities for the use of satellite positioning. In *IOP Conference Series Earth and Environ-Mental Science*; IOP Publishing: Bristol, UK, 2020; Volume 578, p. 012052.
8. Subedi, S.; Kwon, G.R.; Shin, S.; Hwang, S.; Pyun, J. Beacon based indoor positioning system using weighted centroid localization approach. In Proceedings of the IEEE Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 1016–1019.
9. Nemeč, D.; Šimák, V.; Janota, A.; Hruboš, M.; Bubeníková, E. Precise localization of the mobile wheeled robot using sensor fusion of odometry, visual artificial landmarks and inertial sensors. *Robot. Auton. Syst.* **2019**, *112*, 168–177. [[CrossRef](#)]
10. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 25–30 May 2015; pp. 2174–2181.
11. Gui, J.; Gu, D.; Wang, S.; Hu, H. A review of visual inertial odometry from filtering and optimisation perspectives. *Adv. Robot.* **2015**, *29*, 1289–1301. [[CrossRef](#)]
12. Taketomi, T.; Uchiyama, H.; Ikeda, S. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 1–11. [[CrossRef](#)]
13. Sun, K.; Mohta, K.; Pfrommer, B.; Watterson, M.; Liu, S.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robot. Autom. Lett.* **2018**, *3*, 965–972. [[CrossRef](#)]
14. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.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](#)]
15. 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](#)]
16. Heo, S.; Park, C.G. Consistent EKF-based visual-inertial odometry on matrix Lie group. *IEEE Sens. J.* **2018**, *18*, 3780–3788. [[CrossRef](#)]
17. Ma, S.; Bai, X.; Wang, Y.; Fang, R. Robust stereo visual-inertial odometry using nonlinear optimization. *Sensors* **2019**, *19*, 3747. [[CrossRef](#)]
18. Gargoum, S.; Basyouny, K. Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation. In Proceedings of the IEEE International Conference on Transportation Information and Safety (ICTIS), Banff, AB, Canada, 8–10 August 2017; pp. 563–574.
19. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. *Robot. Sci. Syst.* **2014**, *2*, 1–9.
20. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020; pp. 5135–5142.
21. Shi, X.; Liu, T.; Han, X. Improved Iterative Closest Point (ICP) 3D point cloud registration algorithm based on point cloud filtering and adaptive fireworks for coarse registration. *Int. J. Remote Sens.* **2020**, *41*, 3197–3220. [[CrossRef](#)]
22. Li, L.; Yang, F.; Zhu, H.; Li, D.; Li, Y.; Tang, L. An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells. *Remote Sens.* **2017**, *9*, 433. [[CrossRef](#)]
23. Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing ICP variants on real-world data sets. *Auton. Robot.* **2013**, *34*, 133–148. [[CrossRef](#)]
24. Lee, S.; Kim, C.; Cho, S.; Myoungho, S.; Jo, K. Robust 3-Dimension Point Cloud Mapping in Dynamic Environment Using Point-Wise Static Probability-Based NDT Scan-Matching. *IEEE Access* **2020**, *8*, 175563–175575. [[CrossRef](#)]
25. Hutter, M.; Gehring, C.; Jud, D.; Lauber, A.; Bellicoso, C.D.; Tsounis, V.; Hwangbo, J.; Bodie, K.; Fankhauser, P.; Bloesch, M. Anymal—a highly mobile and dynamic quadrupedal robot. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 38–44.
26. Forster, C.; Carbone, L.; Dellaert, F.; Scaramuzza, D. On-manifold pre-integration for real-time visual-inertial odometry. *IEEE Trans. Robot.* **2016**, *33*, 1–21. [[CrossRef](#)]
27. Plyer, A.; Le, B.G.; Champagnat, F. Massively parallel Lucas Kanade optical flow for real-time video processing applications. *J. Real-Time Image Process.* **2016**, *11*, 713–730. [[CrossRef](#)]
28. Deray, J.; Solà, J.; Andrade, C.J. Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration. In Proceedings of the IEEE European Conference on Mobile Robots (ECMR), Prague, Czech Republic, 4–6 September 2019; pp. 1–6.
29. Chang, L.; Niu, X.; Liu, T. GNSS/IMU/ODO/LiDAR-SLAM integrated navigation system using IMU/ODO pre-integration. *Sensors* **2020**, *20*, 4702. [[CrossRef](#)] [[PubMed](#)]

30. Yuan, Z.; Zhu, D.; Chi, C.; Tang, J.; Liao, C.; Yang, X. Visual-inertial state estimation with pre-integration correction for robust mobile augmented reality. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 1410–1418.
31. Le, G.C.; Vidal, C.T.; H, S. 3d lidar-imu calibration based on upsampled preintegrated measurements for motion distortion correction. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–26 May 2018; pp. 2149–2155.
32. Li, P.; Wang, R.; Wang, Y.; Tao, W. Evaluation of the ICP algorithm in 3D point cloud registration. *IEEE Access* **2020**, *8*, 68030–68048. [[CrossRef](#)]
33. Kim, H.; Song, S.; Myung, H. GP-ICP: Ground plane ICP for mobile robots. *IEEE Access* **2019**, *7*, 76599–76610. [[CrossRef](#)]
34. Shen, Y.; Feng, C.; Yang, Y.; Tian, D. Mining point cloud local structures by kernel correlation and graph pooling. In Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4548–4557.
35. Sirtkaya, S.; Seymen, B.; Alatan, A.A. Loosely coupled Kalman filtering for fusion of Visual Odometry and inertial navigation. In Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey, 9–12 July 2013; pp. 219–226.
36. Von, S.L.; Usenko, V.; Cremers, D. Direct sparse visual-inertial odometry using dynamic marginalization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2510–2517.
37. Yang, L.; Ma, H.; Wang, Y.; Xia, J.; Wang, C. A Tightly Coupled Li-DAR-Inertial SLAM for Perceptually Degraded Scenes. *Sensors* **2022**, *22*, 3063. [[CrossRef](#)] [[PubMed](#)]
38. Zhang, Q.; Xu, G.; Li, N. Improved SLAM closed-loop detection algorithm based on DBow2. *J. Phys. Conf. Series. IOP Publ.* **2019**, *1345*, 042094. [[CrossRef](#)]
39. Tafti, A.P.; Baghaie, A.; Kirkpatrick, A.B.; Holz, J.D.; Owen, H.A.; D’Souza, R.M.; Yu, Z. A comparative study on the application of SIFT, SURF, BRIEF and ORB for 3D surface reconstruction of electron microscopy images. *Comput. Methods Biomed. Bio-Med. Eng. Imaging Vis.* **2018**, *6*, 17–30. [[CrossRef](#)]
40. Ferraz, L.; Binefa, X.; Moreno, N.F. Very fast solution to the PnP problem with algebraic outlier rejection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 501–508.
41. Guan, B.; Yu, Y.; Su, A.; Shang, Y.; Yu, Q. Self-calibration approach to stereo cameras with radial distortion based on epipolar constraint. *Appl. Opt.* **2019**, *58*, 8511–8521. [[CrossRef](#)]
42. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [[CrossRef](#)]
43. Shan, T.; Englot, B.; Ratti, C.; Rus, D. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, 30 May–5 June 2021; pp. 5692–5698.