




## Article

# A Spatiotemporal Calibration Algorithm for IMU–LiDAR Navigation System Based on Similarity of Motion Trajectories

Yunhui Li , Shize Yang, Xianchao Xiu  and Zhonghua Miao \* 

School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China

\* Correspondence: zhhmiao@shu.edu.cn

**Abstract:** The fusion of light detection and ranging (LiDAR) and inertial measurement unit (IMU) sensing information can effectively improve the environment modeling and localization accuracy of navigation systems. To realize the spatiotemporal unification of data collected by the IMU and the LiDAR, a two-step spatiotemporal calibration method combining coarse and fine is proposed. The method mainly includes two aspects: (1) Modeling continuous-time trajectories of IMU attitude motion using B-spline basis functions; the motion of the LiDAR is estimated by using the normal distributions transform (NDT) point cloud registration algorithm, taking the Hausdorff distance between the local trajectories as the cost function and combining it with the hand–eye calibration method to solve the initial value of the spatiotemporal relationship between the two sensors' coordinate systems, and then using the measurement data of the IMU to correct the LiDAR distortion. (2) According to the IMU preintegration, and the point, line, and plane features of the lidar point cloud, the corresponding nonlinear optimization objective function is constructed. Combined with the corrected LiDAR data and the initial value of the spatiotemporal calibration of the coordinate systems, the target is optimized under the nonlinear graph optimization framework. The rationality, accuracy, and robustness of the proposed algorithm are verified by simulation analysis and actual test experiments. The results show that the accuracy of the proposed algorithm in the spatial coordinate system relationship calibration was better than  $0.08^\circ$  ( $3\sigma$ ) and 5 mm ( $3\sigma$ ), respectively, and the time deviation calibration accuracy was better than 0.1 ms and had strong environmental adaptability. This can meet the high-precision calibration requirements of multisensor spatiotemporal parameters of field robot navigation systems.

**Keywords:** multisensor fusion; multisensor calibration; factor graph optimization; state estimation



**Citation:** Li, Y.; Yang, S.; Xiu, X.; Miao, Z. A Spatiotemporal Calibration Algorithm for IMU–LiDAR Navigation System Based on Similarity of Motion Trajectories. *Sensors* **2022**, *22*, 7637. <https://doi.org/10.3390/s22197637>

Academic Editors: Luis Payá, Oscar Reinoso García and Helder Jesus Araújo

Received: 30 August 2022

Accepted: 4 October 2022

Published: 9 October 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

Multisensor fusion is an important way to aid in the environmental perception and navigation of robots in complex environments. It can fully use the advantages of different sensors, and they can then compensate each other, and achieve more accurate and robust perception and localization performance than that of a single sensor. Therefore, in recent years, it has received extensive attention, such as in driverless cars, field robots, high-precision map construction, and target tracking [1–4]. LiDAR is widely used in robot navigation due to its high measurement accuracy, insensitivity to light, and good reliability. Its fusion with the environment-independent IMU can achieve robust perception and localization in complex environments. The calibration of the spatiotemporal relationship between sensors is the premise of realizing multisensor information fusion, which mainly includes the calibration of the spatial transformation relationship between sensor coordinate systems and the calibration of the time deviation of sensor acquisition data.

To realize the calibration of the multisensor spatial transformation relationship in a robot navigation system, many methods have emerged in recent years that are mainly divided into methods based on calibration equipment and self-calibration methods. The method based on calibration equipment refers to the estimation of spatial transformation

parameters using prior information provided by external sensors, rotary tables, pendulums, calibration targets, and other equipment. The calibration accuracy of this method is related to the calibration equipment [5–8]. For example, Liu et al. used a precision turntable, a standard plane, and spherical and cylindrical targets to estimate the motion of LiDAR, modeled the calibration problem as a maximal likelihood estimation problem, and solved it, but this method is time-consuming and labor-intensive, and requires expensive calibration equipment. This severely limits the scope of application of this method [9]. Since the self-calibration method does not have special calibration equipment constraints, calibration is convenient, fast, and widely popular because of its good adaptability. This is the current state of the art calibration scheme. For example, Geiger et al. proposed a motion-based calibration method, and its accuracy is related to the uncertainty of the motion estimation of the sensor to be calibrated [10]. Michael et al. constructed a camera–IMU spatiotemporal calibration problem as a nonlinear optimization problem, and solved the spatiotemporal calibration parameters by obtaining the trajectories of each sensor. This method assumes that the motion trajectory of the sensor has second-order differentiability, and needs to obtain the three-dimensional (3D) information of the environment in advance to realize the Euclidean calibration of the translation parameters [11]. Another study [12] used Gaussian process regression to model the IMU measurement, and combined the IMU preintegration method to obtain the measurement value of the LiDAR sampling time. In the probability framework, the minimal distance from the point to the plane is used as the constraint condition to construct a method. A nonlinear optimization problem solves the spatial transformation parameters, but the method requires some known planar features as a map. Furgale et al. developed the Kalibr calibration software. It is a method that jointly estimates the time deviation and the external parameters between sensors on the basis of a maximal likelihood estimation framework, and derives and realizes the estimation of the spatiotemporal parameters between multiple sensors on the basis of the B-spline basis function. This is an offline calibration method that also requires the participation of the calibration target [13]. Rehder et al. adopted the same strategy to calibrate a camera–IMU–lidar system. First, they calibrated the coordinate system transformation relationship between the camera and the IMU on the basis of continuous-time nonlinear optimization, and then calibrated the external parameters between the LiDAR and the IMU according to the calibration results [14]. Lv et al. proposed a method for calibrating the transformation parameters between LiDAR and IMU coordinate systems on the basis of B-spline continuous-time motion trajectory modeling. The B-spline curve was used to obtain IMU measurement data at the time of LiDAR sampling. The calibration parameters were solved by means of nonlinear optimization [15]. Furthermore, in recent years, methods for multisensor calibration using fuzzy logic and neural networks have also emerged. In [16], the authors proposed a method performing line feature-based factor graph optimization and producing fuzzy inference systems to adapt the covariance of the factor graph and implement the self-calibration of external parameters between multiple lidars. The authors in [17] modeled the calibration process of camera–IMU extrinsic calibration using model-free deep reinforcement learning to derive a policy that guides the motions of a robotic arm holding the sensors to efficiently produce measurements. The method automatically generates favorable and calibrated motion trajectories through the learning strategy, which can greatly improve the calibration efficiency and has high calibration accuracy.

Another problem that arises when dealing with the data fusion of multisensor systems is data synchronization. To calibrate the multisensor time offset, development has been rapid in recent years. For example, Mair et al. proposed an initialization method using cross-entropy or phase-consistent multisensor spatiotemporal calibration to separate the calibration variable from other variables and establish an analytical time deviation model. This can provide better prior information for the calibration algorithm without being affected by other variables [18]. Kelly et al. aligned the rotation curve of the camera and the IMU, and used the variant ICP method to match the rotation curve to realize the offline calibration of the time offset [19]. Li et al. proposed an online temporal offset calibration motion-estimation method based on the Multi-

State Constraint Kalman Filter (MSCKF) framework for the problem of temporal offset estimation in vision–IMU systems, which has the advantage of low computational complexity [20,21]. Qin jointly optimized the time deviation, camera state, and IMU state to realize the online calibration of the time deviation between the camera and IMU data [22]. In [23], aiming at the problem of time synchronization between the camera and IMU data, combined with the extended Kalman filter state estimator, an online time deviation calibration method was proposed on the basis of improving the perspective projection model. In general, there are many calibration methods for multiple sensors, but there are relatively few unified parameters of time and space between sensors to estimate online simultaneously. On the other hand, the environment faced by robots' autonomous operation is complex and changeable, and the unified parameters are likely to change with time, so it is urgent to develop an online self-calibration method for the spatiotemporal unified multisensor parameters.

To realize the online self-calibration of the spatiotemporal unified parameters of the multisensor fusion navigation system composed of LiDAR and IMU, a two-step calibration strategy is proposed. The method first uses the B-spline basis function to model the data collected by the IMU. Then, it solves the initial value of sensor spatiotemporal parameter calibration on the basis of the motion similarity principle and hand–eye calibration method. Lastly, within the framework of Bayesian maximal likelihood estimation, the objective function of multisensor calibration parameter optimization is constructed on the basis of IMU preintegration and lidar multifeatures, and the calibration problem is solved on the basis of graph optimization theory.

## 2. Navigation System Calibration Problem Description

As shown in Figure 1, the robot navigation system consists of LiDAR and IMU, and the two are rigidly connected. If an arbitrary fixed coordinate system is assumed to be the world coordinate system  $W$ , and its  $z$  axis is parallel to the direction of gravity but opposite, the LiDAR coordinate system is  $L$ , the IMU coordinate system is  $I$ , the rigid body transformation relationship between the two coordinate systems is  $x_{calib}$ , the 3D point cloud collected by the LiDAR in time is  $t$ , which represents the point cloud of each frame, the angular velocity measured by the IMU in time is  $\omega = \{\omega_0, \dots, \omega_j, \dots, \omega_M\}$ ,  $\omega_j$  represents the angular velocity measured at different times, the gyroscope bias is  $b_g$ , the measured linear acceleration is  $A = (a_0, \dots, a_j, \dots, a_M)$ ,  $a_j$  represents the angular velocity measured at different times, the acceleration bias is  $b_a$ . In addition, the time deviation of the LiDAR and IMU data collection is  $\Delta t$ , the pose of the IMU in the world coordinate system at time  $t_j$  is  $\{q_{WtoIj}, p_{WinIj}\}$ , then the online spatiotemporal calibration of the sensors of the navigation system can be described as follows:

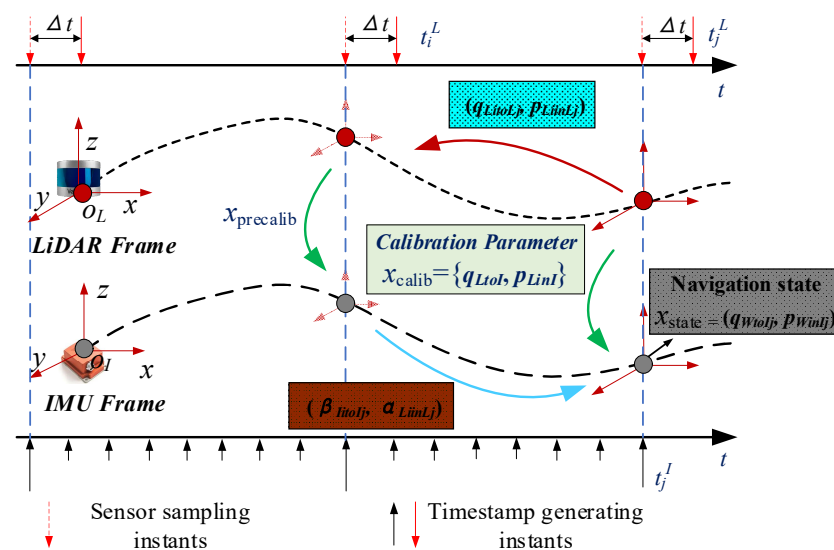


Figure 1. Multisensor spatiotemporal calibration problem description.

Estimating the state variable  $X = (x_{calib}, x_{state})$  of the navigation system according to the known measurement information  $Z = (P, \omega, A)$  and its initial value, and updating it dynamically, where  $x_{state} = (q_{I_0 to I_0}, p_{I_0 to I_0}, b_a^0, b_g^0, \dots, q_{I_N to I_0}, p_{I_N to I_0}, b_a^N, b_g^N)$ ,  $x_{calib} = (q_{L to I}, p_{Lin I}, \Delta t)$ . If the measurements conform to a Gaussian distribution and are independent of each other, the above problem description can be expressed as the following maximal posterior probability estimation form:

$$\begin{aligned}\bar{X} &= \operatorname{argmax} P(X|Z) = \operatorname{argmax} P(x_{1:i}|z_{1:i}) \\ &= \operatorname{argmax} \prod_{i=1}^N P(x_i|x_{i-1})P(z_{i-1} = z|x_{i-1})\end{aligned}\quad (1)$$

where  $z_{1:i}$  represents all measurement data from  $t_1$  to  $t_i$ ,  $x_{1:i}$  represents all navigation system state quantities from  $t_1$  to  $t_i$ , and when  $i = 1$ , set  $P(x_1|x_0) = P(x_1)$ ,  $P(z_0|x_0) = 1$ .

### 2.1. Inertial Measurement Unit Model

The inertial measurement unit is composed of a gyroscope and an accelerometer. If the angular velocity measured by the gyroscope is  $\tilde{\omega}_t$  and the acceleration measured by the accelerometer is  $\tilde{a}_t$ , the inertial measurement unit can be modeled as follows according to the description in the literature [24]:

$$\begin{cases} \tilde{\omega}_t = \omega_t + b_t^g + n_t^g \\ \tilde{a}_t = R_t^{IW}(a_t - g) + b_t^a + n_t^a \end{cases}, \quad (2)$$

where  $\tilde{\omega}_t$  and  $\tilde{a}_t$  represent the measurement of the IMU at time  $t$ ,  $b_t^g$  and  $b_t^a$  represent the offset of  $\tilde{\omega}_t$  and  $\tilde{a}_t$ , respectively,  $n_t^g$  and  $n_t^a$  represent the white noise interference of  $\tilde{\omega}_t$  and  $\tilde{a}_t$ , respectively, and  $R_t^{IW}$  represent the rotation matrix from the world coordinate system to the IMU coordinate system,  $g$  represents the gravitational acceleration in the world coordinate system.

To use the IMU measurement data to estimate the motion of the robot body in the world coordinate system, the following dynamic model of the IMU can be established according to the IMU model:

$$\begin{cases} {}^I_W \dot{q}(t) = \frac{1}{2} \Omega(\omega(t)) {}^I_W q \\ {}^W \dot{p}(t) = {}^W v_I(t), {}^W \dot{v}(t) = {}^I_W R^T a(t) \\ \dot{b}_g(t) = n_g, \dot{b}_a(t) = n_a \end{cases}, \quad (3)$$

where  $\Omega(\bullet)$  represents an antisymmetric matrix, and  ${}^I_W q$  represents the quaternion of the rotation matrix  ${}^I_W R$ . The pose increment can be obtained by integrating the above formula at the time interval  $t \in [t_k, t_{k+1}]$ .

### 2.2. 3D LiDAR Ranging System Model

As shown in Figure 2, according to the literature [25], the mechanical rotation LiDAR measurement is described as  $(\rho, \theta, \phi)$  in the spherical coordinate system; then, the measurement  $(x_d, y_d, z_d)$  in the LiDAR Cartesian coordinate system can be expressed as:

$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} \rho \cos \theta \sin \phi \\ \rho \cos \theta \cos \phi \\ \rho \sin \theta \end{bmatrix}, \quad (4)$$

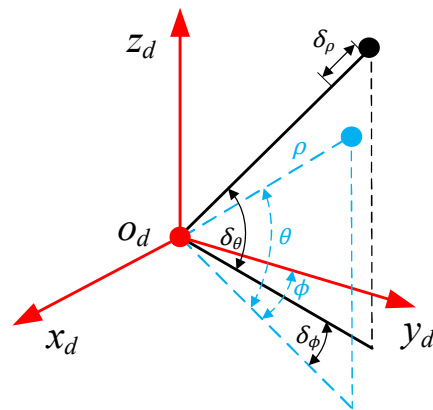


Figure 2. Schematic diagram of LiDAR ranging system.

If the measurement error correction amount is set as  $(\delta_\rho, \delta_\theta, \delta_\phi)$ , the corrected LiDAR measurement can be expressed as Equation (4) in the Cartesian coordinate system. In addition, since the LiDAR measurement is the coordinate value  $P(x_d, y_d, z_d)$  in the Cartesian coordinate system, it is necessary to first convert  $P(x_d, y_d, z_d)$  into the spherical coordinate system, perform error compensation, and then transform it into the Cartesian coordinate system.

$$\begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \begin{bmatrix} (\rho + \delta_\rho) \cos(\delta_\theta) \sin(\phi - \delta_\phi) \\ (\rho + \delta_\rho) \cos(\delta_\theta) \cos(\phi - \delta_\phi) \\ (\rho + \delta_\rho) \sin(\delta_\theta) \end{bmatrix}, \quad (5)$$

### 2.3. Geometric Model of Robot Motion Trajectory

According to the literature [26], when the robot motion trajectory control nodes are uniformly distributed, the B-spline basis function in the form of the following matrix can be used for modeling:

$$p(t) = \sum_{j=0}^d u^T M_{(j)}^{(d+1)} p_{i+j}, \quad (6)$$

where  $d$  is the order of the B-spline basis function, which means that the continuous motion trajectory  $p(t)$  in the  $t \in [t_i, t_{i+1})$  interval is only determined by the control point  $p_i, p_{i+1}, \dots, p_{i+d}$  corresponding to time  $t_i, t_{i+1}, \dots, t_{i+d}$ ;  $u^T = [1, u, \dots, u^d]$ ,  $u = (t - t_i) / (t_{i+1} - t_i)$ ,  $M_{(j)}^{(d+1)}$  represents the  $j$ -th column of the spline matrix  $M^{(d+1)}$ ; therefore, the accumulation of motion trajectories can be expressed as:

$$p(t) = p_i + \sum_{j=1}^d u^T \tilde{M}_{(j)}^{(d+1)} (p_{i+j} - p_{i+j-1}), \quad (7)$$

The authors in [27] modeled the uniformly distributed unit quaternion control point  $q_i$  in  $SO(3)$  space:

$$q(t) = q_i \otimes \prod_{j=1}^d \text{Exp}\left(u^T \tilde{M}_{(j)}^{(4)} \text{Log}\left(q_{i+j-1}^{-1} \otimes q_{i+j}\right)\right), \quad (8)$$

where  $\text{Exp}(\bullet)$  represents the mapping from Lie algebras to Lie groups, and  $\text{Log}(\bullet)$  represents the mapping from Lie groups to Lie algebras.

This paper uses the cubic spline basis function to model the motion trajectory in which the spline matrix and the cumulative spline matrix can be expressed as:

$$M^{(4)} = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \quad \tilde{M}^{(4)} = \frac{1}{6} \begin{bmatrix} 6 & 5 & 1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix}, \quad (9)$$

### 3. Spatiotemporal Calibration of Multisensor Navigation System

Since the IMU can only sense the acceleration and angular velocity information of the sensor itself, LiDAR can only measure the distance information of the surrounding environment in the sensor coordinate system. Therefore, this paper proposes a two-step spatiotemporal parameter calibration method with first coarse and then fine. First, LiDAR odometry is used. The relative motion trajectories of LiDAR are obtained by calculating, combined with the IMU motion perception information and trajectory similarity evaluation criteria, the calibration of the initial values of spatiotemporal unified parameters between IMU and LiDAR. Then, according to the model of the navigation system calibration problem in the previous section, the Bayesian factor graph model is used to construct the optimization model of the spatiotemporal calibration parameters of the navigation system, the initial value of the calibration is refined and dynamically updated, and the high-precision spatiotemporal unified parameter calibration results are obtained online. Its principle is shown in Figure 3.

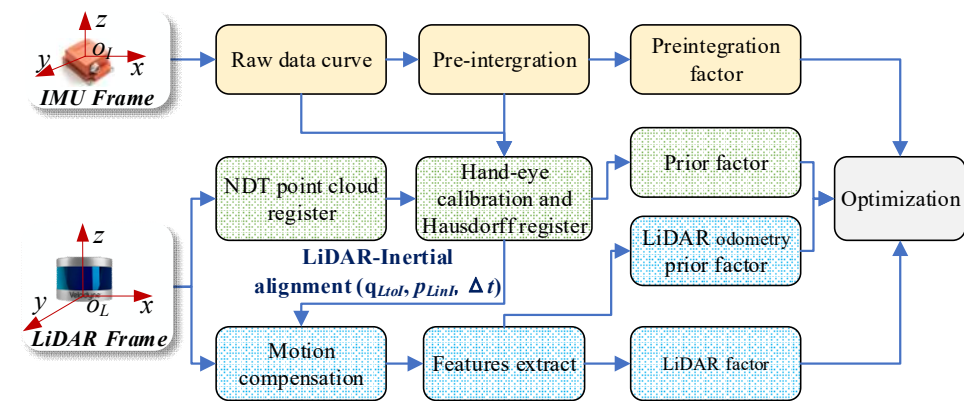


Figure 3. Block diagram of multisensor spatiotemporal calibration.

#### 3.1. Initial Multisensor Calibration Value Based on Local Trajectory Similarity

As shown in Figure 1, let the motion trajectory of the IMU relative to the world coordinate system be  $\Gamma_I = \{q_{WtoI}(t_1), p_{WinI}(t_1)\}$ , and the motion trajectory of the LiDAR relative to the initial position is  $\Gamma_L = \{q_{L_0toL_i}(t_2), p_{L_0inL_i}(t_2)\}$ , where  $t_1$  and  $t_2$  represent the time stamps of IMU data acquisition and LiDAR data acquisition time stamp, respectively. The time deviation can be expressed as  $\Delta t = t_2 - t_1$ ,  $q_{WtoI}$  represents the attitude rotation matrix from the IMU coordinate system to the world coordinate system,  $p_{WinI}$  represents the translation vector of the IMU coordinate system in the world coordinate system, and  $q_{L_0toL_i}$  represents the rotation matrix of LiDAR coordinate system to the LiDAR initial coordinate system,  $p_{L_0inL_i}$  represents the translation vector of the LiDAR coordinate system in the LiDAR zero coordinate system. If the trajectory estimation accuracy of the two rigidly connected sensors is high enough, and there is no time deviation between the sensors, then the shapes of their attitude motion trajectories  $\Gamma_I(q)$  and  $\Gamma_L(q)$  are similar, so points  $^Iq$  and  $^Lq$  at the same moment on the two trajectories can be arbitrarily selected to calculate the two transformation relationship between trajectories is as follows:

$$\tilde{q}_{ItoL} q_{WtoI} q_{I_0toW} \tilde{q}_{LtoI} = q_{L_0toL_i}, \quad (10)$$



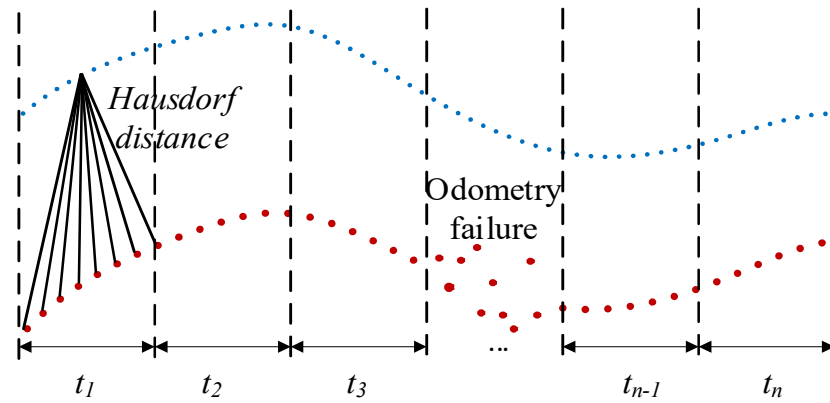
where the attitude trajectory  $q_{WtoI_i}$  of the IMU can be obtained by solving Equation (6).

Using the transformation relationship  $\tilde{q}_{LtoI}$  between the trajectories obtained according to Equation (8) to transform the IMU estimated motion estimation to obtain the trajectory  ${}^I\Gamma_L(q)$ , and because the Hausdorff distance can describe the similarity between two discrete trajectories, it can be expressed according to Equation (11) as Hausdorff distance that solves the similarity between  ${}^I\Gamma_L(q)$  and  ${}^I\Gamma_I(q)$ .

$$H_x = \max(h(x_i, x_j), h(x_j, x_i)), \quad (11)$$

$$\begin{aligned} h(x_i, x_j) &= \max_{x_i \in {}^I\Gamma_L^x} \min_{x_j \in {}^I\Gamma_I^x} \|x_i(t) - x_j(t - \Delta t)\| \\ h(x_j, x_i) &= \max_{x_j \in {}^I\Gamma_I^x} \min_{x_i \in {}^I\Gamma_L^x} \|x_j(t) - x_i(t - \Delta t)\| \end{aligned} \quad (12)$$

where  $H$  represents the similarity between trajectories. To eliminate the failure area of odometry estimation, the trajectory is segmented as shown in Figure 4, and the Hausdorff distance within the trajectory segment is calculated separately. If the similarity between trajectories is less than the set threshold, the estimated trajectory can be used for initial parameter calibration.



**Figure 4.** Trajectory similarity selects initial calibration trajectory segment. The red dots represent the lidar odometry localization data; the blue dots represent the IMU odometer localization data.

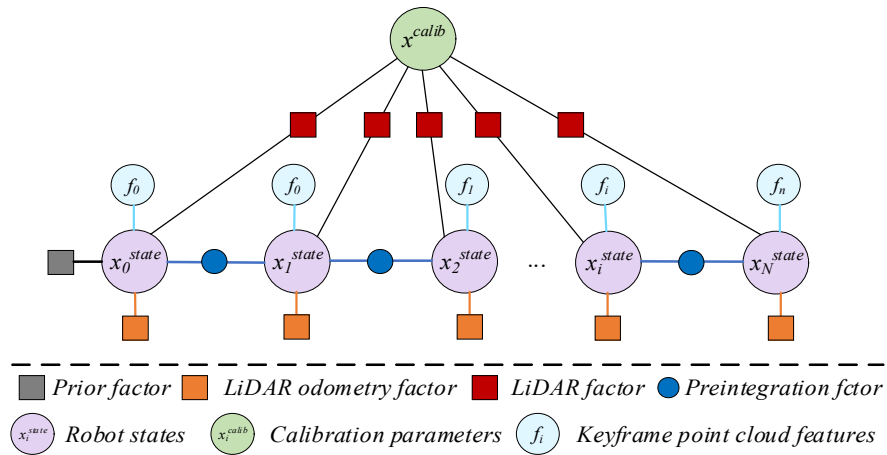
In addition, inspired by the literature [9], according to Equation (8), the angular velocity measured by the gyroscope is used to estimate the continuous time motion trajectory  ${}^I\Gamma_I(t)$  of the IMU attitude, which can avoid the accumulation of errors caused by the measurement error on the results of the integration method, and is estimated by the NDT point cloud matching algorithm. The relative motion trajectory  ${}^L\Gamma_L$  of the LiDAR, and the corresponding IMU motion trajectory point is obtained according to the timestamp of each point on the LiDAR motion trajectory. Lastly, each corresponding timestamp on the two sensor trajectories is transformed according to Equation (10), and the nonlinear least-squares problem is constructed:

$$f(\tilde{q}_{ItoL}, \Delta t) = ar \cos(\tilde{q}_{ItoL} q_{ItoI_0}(i + \Delta t) \tilde{q}_{LtoI} q_{L_0toL_i}), \quad (13)$$

For all the trajectory points that meet the requirements, the cost is calculated with the above formula, the Levenberg–Marquardt (LM) nonlinear optimization algorithm is used to solve the above nonlinear least squares optimization problem, and the initial values  $q_{LtoI}$  and  $\Delta t$  of the spatiotemporal unified parameters are obtained. After the initial parameters are obtained by calibration, the preintegration operation is used to integrate the IMU measurement, the motion at the moment of LiDAR frame data acquisition is obtained by preintegration operation, the motion distortion of LiDAR data acquisition is compensated accordingly, and the above process is repeated to obtain the final initial calibration parameters.

### 3.2. Multisensor Spatiotemporal Parameter Calibration Nonlinear Optimization

Since the above initial value calibration results do not consider the bias effect of the IMU, and the data acquisition frame rate of LiDAR is low, the initial value calibration accuracy of the above multisensor spatiotemporal unified parameters is still insufficient. Therefore, in this section, the initial calibration results of external parameters as the initial value, according to the description of the state estimation problem in this paper by Equation (1), use the factor graph model to model it, fully using the sensor motion sensing information of the navigation system, and building the factor graph model shown in Figure 5, which mainly includes: prior factor, IMU preintegration factor, LiDAR factor, LiDAR odometer prior factor. The nonlinear optimization algorithm is used to solve the above model to obtain high-precision calibration results.



**Figure 5.** Multisensor spatiotemporal calibration factor graph model.

#### 3.2.1. IMU Preintegration Factor

To avoid the repeated integration problem after the linearization point changes, the literature [28] proposed a preintegration method to integrate the motion state in the interval. According to the preintegration formula, this paper uses the preintegration amount between the two states of the IMU as the constraint factor to constrain the navigation state between the two moments:

$$\begin{cases} e_p^w = q_{b_i w} \left( p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2 \right) - \alpha_{b_i b_j} \\ e_q^w = 2 \left[ q_{b_i b_j} \otimes \left( q_{b_i w} \otimes q_{wb_j} \right) \right]_{xyz} \\ e_v^w = q_{b_i w} \left( v_j^w - v_i^w + g^w \Delta t \right) - \beta_{b_i b_j} \\ e_{b_a}^w = b_j^a - b_i^a \\ e_{b_g}^w = b_j^g - b_i^g \end{cases}, \quad (14)$$

where  $q_{bw}$  and  $p_{wb_j}$  represent the attitude transformation quaternion and translation vector between the world coordinate system and the IMU coordinate system, respectively;  $v^w$  represents the velocity vector of the IMU coordinate system in the world coordinate system;  $g^w$  represents the gravitational acceleration in the world coordinate system; and  $b_a$  and  $b_g$  represent the accelerometer and gyroscope bias errors, respectively.

#### 3.2.2. LiDAR Odometry Prior Factor

Because LiDAR odometry was used to estimate the motion state  $T_{L_0 to L_i}$  of the unmanned system in the initial calibration stage of external parameters, in order to fully use the LiDAR point cloud information to estimate the motion state of the unmanned system and the external structure parameters of the sensor, the LiDAR odometer is introduced in each frame to firstly estimate the motion state  $T$  of the unmanned system. In addition,



considering that the point cloud collected by LiDAR is seriously distorted when the unmanned system moves violently, the accuracy of the above NDT-based LiDAR odometer is low. Therefore, this paper uses the B-spline interpolation algorithm to process the above preintegrated data. Solve the pose of the LiDAR coordinate system relative to the LiDAR initial zero coordinate system (defined as the map coordinate system) at the time of LiDAR point collection.

$$P_{i+\tau}^{L_0} = T_{L_0I}^{-1} \cdot T_{WtoI_0} \cdot T_{WtoI_i}^{-1} \cdot T_{I_i to I_{i+\tau}}^{-1} \cdot T_{L_0I} P_{i+\tau}^{L_{i+\tau}}, \quad (15)$$

where  $T_{L_0I}$  represents the external parameters between sensors;  $T_{WtoI_0}$  represents the pose of the IMU zero coordinate system in the world coordinate system;  $T_{WtoI_i}$  represents the pose of the IMU coordinate system in the world coordinate system at time  $t_i$ ;  $T_{I_i to I_{i+\tau}}$  represents the IMU pose of the LiDAR point sampling time between  $t_{i+\tau}$  and  $t_i$  that can be obtained with the interpolation of Equations (7) and (8).  $T_{L_0 to L_i}$  is recalculated after removing the distortion.

### 3.2.3. LiDAR Odometry Factor

#### 1. Point Cloud Feature Extraction;

To extract the features of 3D point clouds, in 1999, Mass, Vosselman, et al. proposed to use the invariant spatial moment of the point cloud to represent the geometric characteristics of the point cloud [29]. In 2009, Jutzi, Cross, et al. used a 3D covariance matrix (i.e., the 3D structure tensor) composed of the invariant moments of the point cloud to describe the geometric features of the 3D point cloud in this range [30]. This paper uses the structure tensor to extract the geometric features of the point cloud. It can be expressed as follows.

First, the 3D structure tensor (covariance matrix) composed of point cloud spatial moment is calculated.

$$S = \frac{1}{k} \sum_{i \in P} (p_i - \bar{p})(p_i - \bar{p})^T, \quad (16)$$

where  $P$  and  $\bar{p}$  represent the  $k$ -nearest neighbor set and the center point of the set  $P$ , respectively.

Second, matrix decomposition is performed on the structure tensor to obtain eigenvalues and eigenvectors.

$$S = U \Sigma V^T, \Sigma = \text{diag}(\lambda_1, \lambda_2, \lambda_3), \quad (17)$$

Lastly, compute geometric description features from eigenvalues.

$$\text{linearity} : \frac{\lambda_1 - \lambda_2}{\lambda_1}, \text{planarity} : \frac{\lambda_2 - \lambda_3}{\lambda_1}, \quad (18)$$

Only point clouds whose linearity and planarity are within a certain threshold range are retained as the extracted line features and plane features. On the other hand, considering the sparseness of LiDAR point cloud, this paper extracts point cloud features of points, lines, and planes in the environment according to the method of the literature [31].

#### 2. LiDAR Factor;

First, the invariant moment of the 3D point cloud within a certain range is calculated for each frame of point cloud, and the covariance matrix  $\Sigma = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$  is constructed to describe the geometric characteristics of the 3D point cloud within this range. The geometric description features of the point cloud are obtained, and the point, line, and plane features are extracted accordingly. In addition, planar features are extracted by evaluating the curvature of adjacent points, and points with small curvature values are classified as planar features. Cross-validation checks are performed on the geometric features extracted by the above methods to obtain the final point, line, and plane features. The following

constraint factors are constructed according to the distance between point-to-point, line, and plane:

$$\begin{cases} e^{p_o \rightarrow p_o} = {}^M q_i - (R_{L_i to M} p_i + t_{L_i in M}), {}^M q_i, p_i \in F_{i+1}^{p_o} \\ e^{p_o \rightarrow l_i} = {}^M v_j \times ({}^M q_j - (R_{L_i to M} p_j + t_{L_i in M})), {}^M q_j, p_j \in F_{i+1}^{l_i} \\ e^{p_o \rightarrow p_l} = {}^M n_k^T ({}^M q_k - (R_{L_i to M} p_k + t_{L_i in M})), {}^M q_k, p_k \in F_{i+1}^{p_l} \end{cases}, \quad (19)$$

where  $e^{p_o \rightarrow p_o}$ ,  $e^{p_o \rightarrow l_i}$ , and  $e^{p_o \rightarrow p_l}$  represent the distance between points, point-to-lines, and point-to-planes, respectively; and  ${}^M v_j$  and  ${}^M n_k$  represent the direction vector and normal vector in the feature line and plane in the map coordinate system, respectively.  $[R_{L_i to M}, t_{L_i in M}]$  represents the pose between the current LiDAR coordinate system and the map coordinate system.

### 3.2.4. Objective Function of Nonlinear Optimization

To realize the unified parameter calibration of sensor data, under the conditional of the measurement noise and the process noise conform to Gaussian distribution, the following nonlinear optimization problem is constructed:

$$\min_X \left\{ E_{INS}^{prior} + \sum_{i \in I} \lambda_{l_i} E_i^{l_o} + \sum_{(i,i+1) \in B} \lambda_{imu} \|e_{imu}^i\|_{\Sigma_{e_{imu}^i}}^2 + \sum_{p \in F_i} \lambda_{p_o, p_l, l_i} \|e_i^{p_o \rightarrow p_o, p_l, l_i}\|_{\Sigma_{e_i^{p_o, p_l, l_i}}}^2 \right\}, \quad (20)$$

where  $E_{INS}^{prior}$  is the system prior information,  $E_i^{l_o}$  is the LiDAR odometry prior factor;  $\Sigma_{e_i}$  is the covariance matrix of the factor, and  $\lambda(\bullet)$  is the adjustment parameter  $(3 \times N \times M^{-1} \times \text{mean}(\Sigma_{e_i^{p_o, p_l, l_i}}^{-1}) \times \text{mean}(\text{tr}(\Sigma_{e_{imu}^i})))$ . The proposed algorithm is as follows (Algorithm 1).

---

#### Algorithm 1: LiDAR/IMU spatiotemporal calibration.

---

**Input:** IMU set  $I$  [1], point cloud set  $P \{P_1, \dots, P_N\}$

**Output:** External spatial parameter  $T_{il}$ , temporal parameter  $\Delta t$

**while** (True) **do**

Modeling Gyro measurement data set  $I_i$  with B-splines

Solving the motion between lidar frames by NDT algorithm

Solving external parameters  $T_{il0}$  using hand-eye calibration

Time offset  $\Delta t_0$  solving to minimize the Hausdorff distance

Preintegration using IMU measurement

**for**  $i = 1, \dots, K$  **do**

Distortion correction of point cloud by initial calibration

Construct IMU preintegration factor, lidar prior factor and lidar factor

Perform nonlinear optimization on the data in the window to solve the state  $x_i$

**end**

**end**

---

## 4. Simulation Analysis

To verify the feasibility and statistical properties of the algorithm, the following simulation experiments were designed in which the sampling frequency of IMU was 200 Hz, the sampling frequency of LiDAR was 10 Hz, the number of LiDAR lines was 16, the resolution of horizontal angle was  $0.25^\circ$ , the resolution of vertical angle was  $2^\circ$ , and the gravitational acceleration vector was  $(0, 0, 9.81) \text{ m/s}^2$ . The noise of the sensor was simulated with Gaussian white noise in which the LiDAR point cloud noise conformed to the mean value of 0, variance of 2 cm, the bias of the gyroscope was  $10^{-5}$ , the bias of the acceleration was  $10^{-4}$ , the error of the gyroscope was 0.00015, and the error of the acceleration was 0.00019. The simulation environment is shown in Figure 6. The plane height in the environment was 3 m, the simulation time was 123 s, and the stabilization time of 3 s was set at the start

time. The rotation angle relationship between lidar and IMU was  $(Ax, Ay, Az) = (0, 180, 0)^\circ$ , and the translation relationship was  $(Tx, Ty, Tz) = (0, 40, -60)/\text{mm}$ . The motion trajectory was obtained by cubic spline interpolation as shown in Table 1 for the control point data. The simulation of time offset is realized by translating the timestamps of LiDAR trajectory by 5, 10, 15, 20, and 30 ms. All simulations were performed on a computer with Intel core i7-9700K and 32 GB memory using MATLAB 2019 b as the simulation tool.

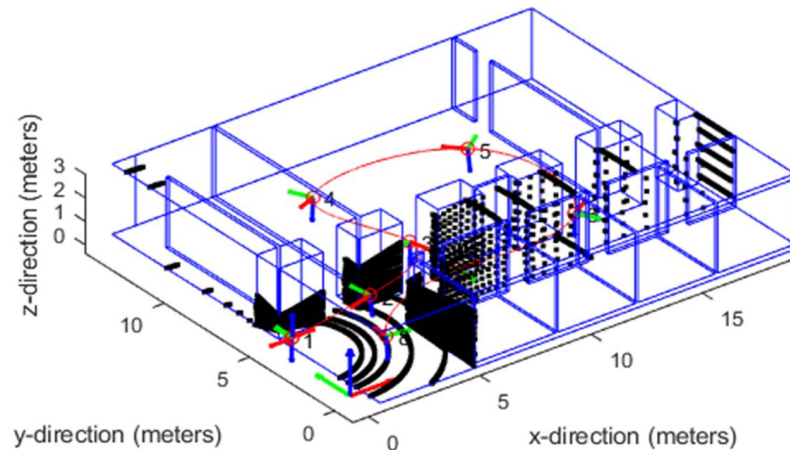


Figure 6. Schematic diagram of the trajectory of IMU and LiDAR.

Table 1. Motion trajectory generation control point parameters.

Num	$x/m$	$y/m$	$z/m$	$roll/^\circ$	$pitch/^\circ$	$yaw/^\circ$
0	0.305	3.810	0.610	0	−180	0
1	3.810	3.810	1.219	0	−188	8
2	7.010	5.669	1.524	0	−174	95
3	7.224	11.582	0.610	0	−176	25
4	13.472	10.668	0.914	0	−185	−55
5	13.259	4.145	1.219	0	−180	−150
6	7.772	3.810	0.914	0	−180	−180
7	2.438	1.067	1.219	0	−188	−100

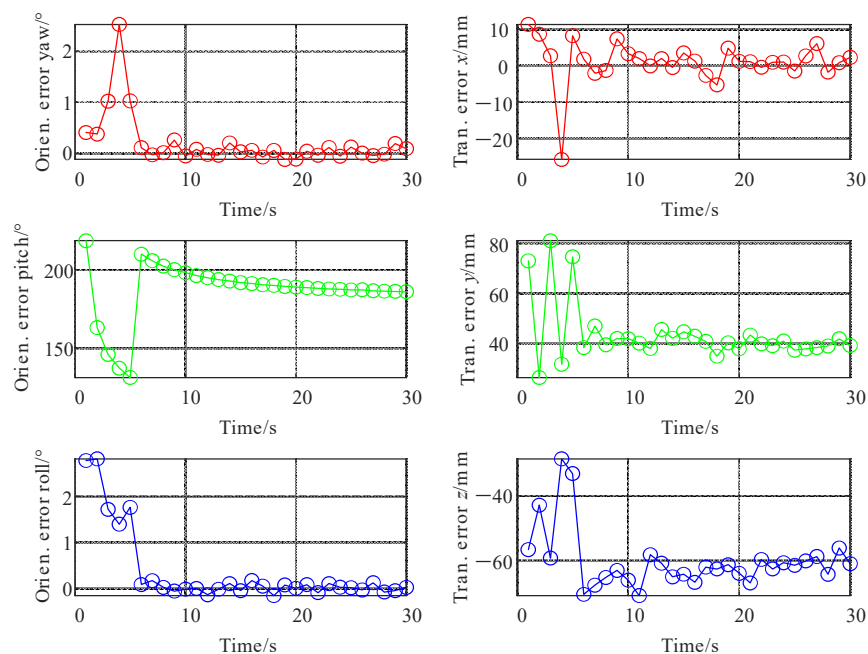
Under the above simulation parameter settings [13], Kalibr and the algorithm proposed in this paper were used to solve the spatial unified parameters. The Kalibr algorithm is an offline solution after importing the LiDAR data; statistics of the calibration error of its spatial parameters are shown in Figure 7, and the calibration error of time deviation is shown in Table 2.

Table 2. IMU and lidar time offset calibration results.

Num.	5 ms	10 ms	15 ms	20 ms	30 ms
Kalibr	4.87	9.84	15.22	19.84	29.88
Linear *	5.19	10.21	14.81	20.18	30.15
Proposed	5.34	10.19	14.83	19.85	29.91

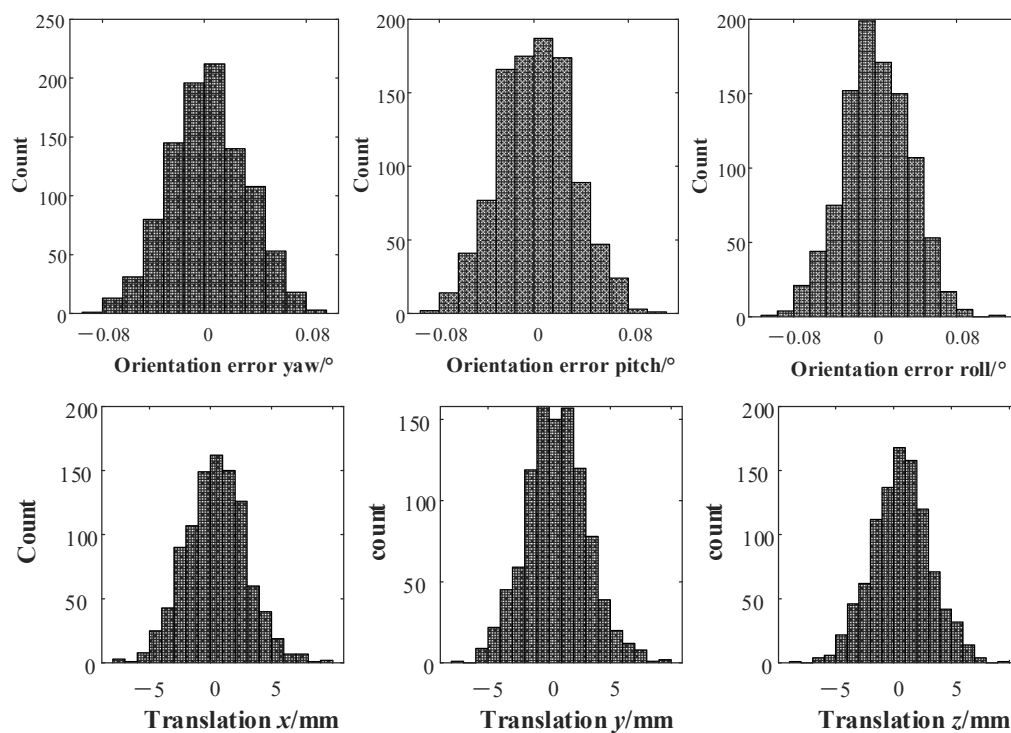
\* The linear method refers to the time deviation initial value solution algorithm proposed in this paper.

Figure 6 shows that the translation calibration error of the algorithm proposed in this paper fluctuated greatly within the initial 5 s. This was mainly because the addition time interval of the initial calibration parameters is set to 5 s. During this period, the prior calibration factor and LiDAR odometry factors dominated; however, with the increase in time, new factors were continuously added, the error continued to decrease, and it basically converged after 15 s, which shows that the algorithm proposed in this paper can effectively realize the spatial parameters calibration between IMU and LiDAR.



**Figure 7.** External parameters' calibration error of IMU and LiDAR varies with time.

Table 2 shows that the time deviation calibration error was significantly reduced before and after the nonlinear optimization. The optimized error is comparable to the Kalibr calibration result, and the final error was less than 0.1 ms. When the time deviation was larger, the calibration effect is better than that of Kalibr. This shows that the algorithm proposed in this paper can effectively realize the time deviation calibration between sensors. In addition, to verify the statistical characteristics of the algorithm proposed in this paper, 1000 simulation calibration experiments were performed, and the root mean square (RMS) error of each experiment was counted. The results are shown in Figure 8.

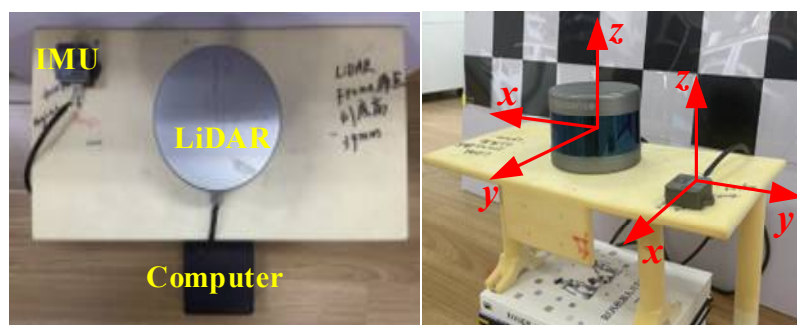


**Figure 8.** Statistical histogram of calibration errors of IMU and LiDAR external parameters.

According to Figure 8, under the conditions that the sensor motion was smooth and the motion excitation was sufficient, the translation calibration error of the algorithm proposed in this paper was 5 mm ( $3\delta$ ), the rotation calibration error was  $0.08^\circ$  ( $3\delta$ ), and the statistical distribution of the calibration error is basically conforming to the normal distribution. This shows that the algorithm proposed in this paper can achieve accurate calibration of the external parameters of IMU and LiDAR in a statistical sense.

## 5. Experiment

To verify the algorithm proposed in this paper, an actual performance test experiment was carried out. As shown in Figure 9, in the actual multisensor external parameter calibration test, the RS-LiDAR-16 (RoboSense, Shenzhen, China) lidar was used, which is a 16-line mechanical rotating radar, and the measurement error was less than 2 cm; the inertial measurement unit adopted HandsFree HFI-A9 (HandsFree Robot, Shenzhen, China), the static angle accuracy was  $0.3^\circ$ , the dynamic angle accuracy was  $1.5^\circ$ , the acceleration range was  $\pm 8$  g, the two were fixed on the same rigid body, and the data processing computer was an Intel NUC11TNK i7 (Ubuntu 20.04, Intel, Santa Clara, CA, United States) with 16 GB memory. In the actual test, the handheld sensor unit moved in 3D space to fully stimulate the acceleration and angular velocity components on each coordinate axis. At the same time, the algorithm proposed in this paper was used to solve the external parameters and time deviation between LiDAR and IMU, and it was compared with the current state of the art. Lastly, the calibrated localization and navigation unit was installed on an unmanned vehicle for localization and mapping experiments.



**Figure 9.** Physical map of IMU and LiDAR assembly.

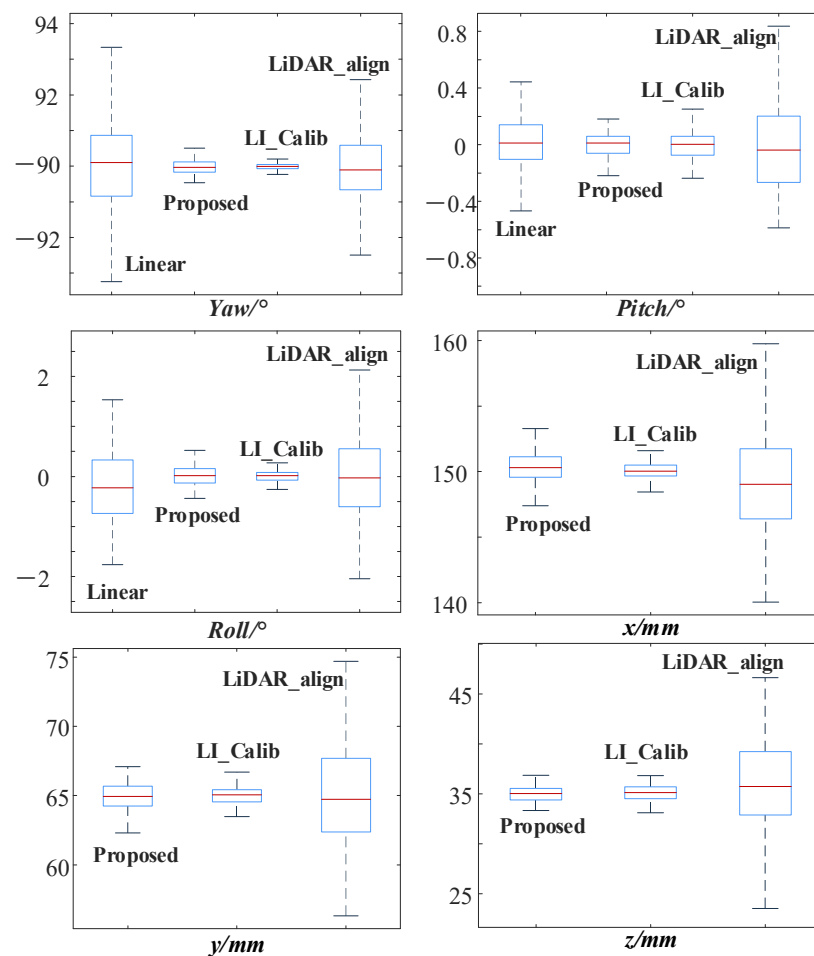
### 5.1. Sensor Spatiotemporal Unified Parameter Calibration Experiment

According to the data collected with the above-mentioned localization and navigation sensor unit, calibration tools `lidar_align` and `calib_lidar_imu`, and calibration algorithms LI-Calib and the linear and nonlinear optimization calibration algorithm proposed in this paper were used to calibrate the external parameters and time deviation. The RMS error statistics are shown in Table 3 and Figure 10, where the reference value of spatial calibration is the CAD reference value, and the reference value of time deviation is the average value of multiple calibrations as the reference value. This was segmented and then calibrated using the above algorithm for each data segment, 10 sets of calibration data were obtained, and statistics were performed.

**Table 3.** Time deviation calibration result (unit: ms).

Method	Mean	Median	Std. Dev.	RMS
Linear *	0.5993	0.6088	0.2340	0.6420
Proposed	0.2442	0.2484	0.1040	0.2647

\* The linear method refers to the time deviation initial value solution algorithm proposed in this paper.



**Figure 10.** Calibration results of external parameters of IMU and LiDAR.

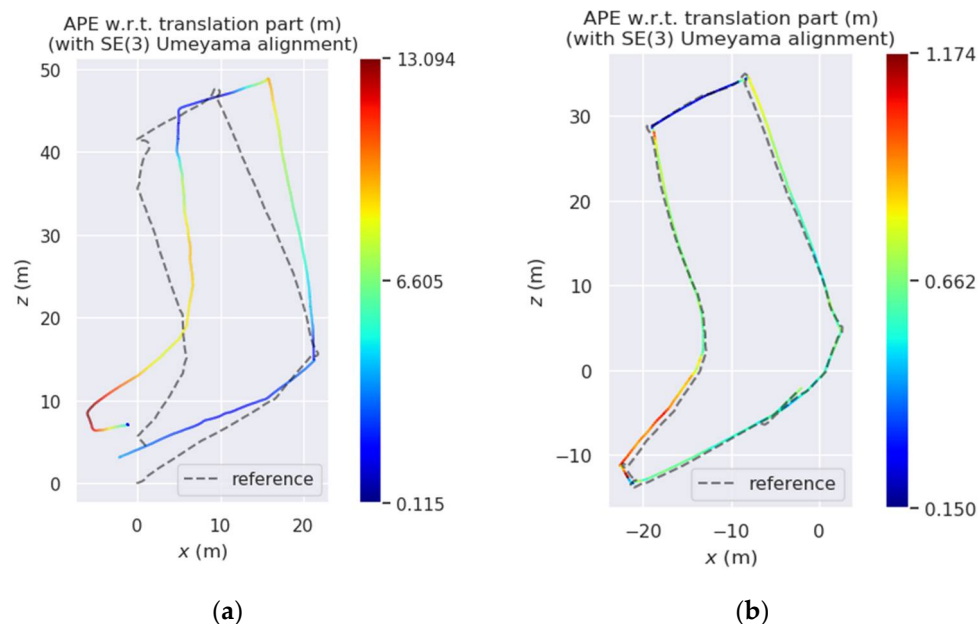
The statistics of the spatiotemporal calibration results shown in Figure 10 and Table 3 show the following. (1) The proposed algorithm was improved compared with other calibration algorithms, and the accuracy was comparable to that of the LI-Calib calibration algorithm. The actual calibration accuracy of the proposed algorithm could reach  $0.1^{\circ}$ , 8 mm, which could meet the high-precision requirements of the unmanned system for the localization and mapping. In addition, benefiting from the algorithm proposed in this paper, the degradation of the localization and determined by considering the localization performance of the odometer during the initial calibration, and the factor graph optimization at the back end enabled the calibration algorithm to adaptively select the localization trajectory data that met the requirements of the odometer localization performance to solve the calibration parameters. (2) The error of the linear calibration algorithm was larger than that of the nonlinear optimization algorithm, and the calibration result distribution range was large compared with that of the optimization-based method. This may have been due to the continuous addition of new calibration constraints during the nonlinear optimization process, so that the calibration results converged to their true values. (3) According to the statistical results of time deviation calibration, both the nonlinear graph optimization algorithm proposed in this paper and the initial calibration algorithm could realize the calibration of time deviation. The square root error reached 0.26 ms, and the improvement in accuracy compared to the initial calibration may have been due to the back-end optimization processing the trajectory data for a longer time.

## 5.2. Sensor Fusion Localization and Mapping Test Experiment

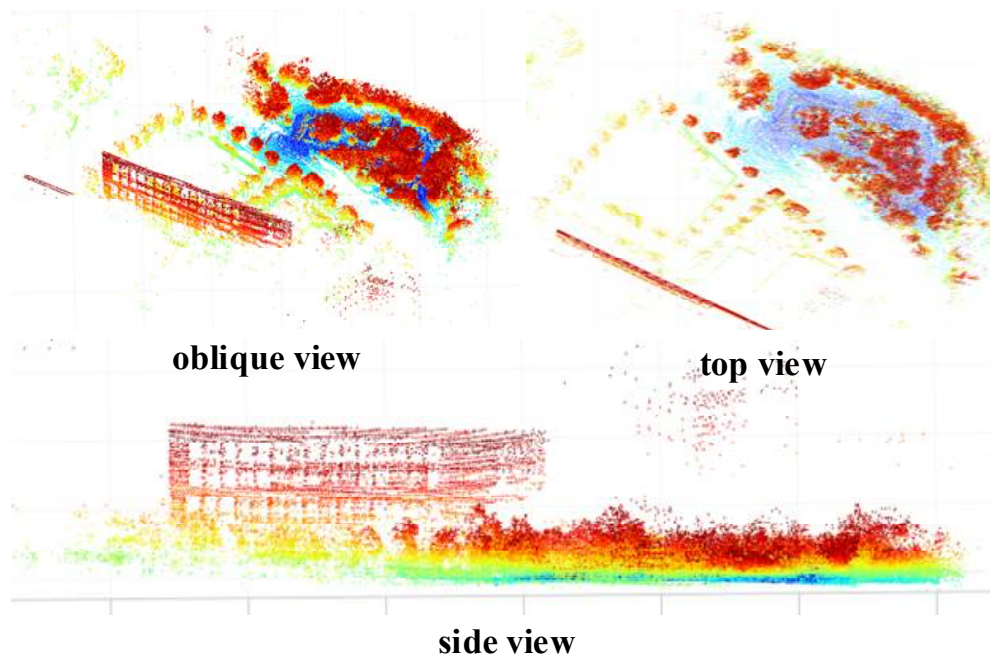
To verify the calibration results obtained by the algorithm in this paper, the calibration results obtained with the calibration algorithm in this paper were combined with those of



the LIO-SAM open-source algorithm and its recommended external parameter calibration algorithm to realize the odometry localization experimental environment mapping experiment of LiDAR and IMU fusion, respectively. EVO was used to evaluate the localization results as shown in Figures 11 and 12 [32].



**Figure 11.** IMU and lidar odometer localization results. (a) LiDAR\_IMU\_calib; (b) proposed.



**Figure 12.** LiDAR point cloud mapping results.

As shown in Figure 11, the localization results show that the localization accuracy was significantly improved before and after adding the spatiotemporal calibration results proposed in this paper, especially at the corners. The reasons may be as follows: (1) With the more data collected, there was an increasing number of constraints involved in calibration and motion state estimation, so that the accuracy of optimization and calibration parameters was continuously improved; (2) the calibration results and the high-frequency measurement information of the IMU could be fully used to correct the distortion of the LiDAR point

cloud frame, so that the point cloud frame motion estimation was more accurate, thereby reducing distortion during violent movements such as cornering. As shown in Figure 12, from the top and side views of the environment modeling results, the point cloud map reproduced the actual scene well in both the vertical and the horizontal directions because the sensor used by the algorithm proposed in this paper is used in the environment modeling. The calibration parameters between IMU and LiDAR data were fused to achieve more accurate motion estimation, so the keyframe point cloud map stitching based on motion estimation was also more accurate.

## 6. Conclusions

Aiming at the problem of unified spatiotemporal parameter calibration between LiDAR and IMU, a spatiotemporal unified parameter calibration algorithm between sensors based on the coarse and fine combination was proposed. The similarity was measured; the goal was to minimize the similarity distance, and nonlinear optimization was performed to solve the initial calibration results. At the same time, it was fully considered that the sensor parameters are likely to change continuously during the operation of the actual unmanned system, so the calibration parameters were introduced into the factor graph of motion estimation to estimate the calibration parameters in real time. The simulation and actual test results show that the two-step calibration strategy in this paper can effectively solve the calibration parameters; because of the factor graph optimization method, it can be flexibly extended to more sensor parameter calibration and fusion localization. Furthermore, compared with the state-of-the-art LiDAR and IMU calibration method LI\_Calib, the spatiotemporal online calibration method proposed in this paper can effectively eliminate odometer faults or drift, and obtain the calibration of time deviation while obtaining the calibration results with approximate accuracy.

The algorithm proposed in this paper needs to fully excite each coordinate axis of the sensor during the calibration process, so it is not suitable for unmanned systems that move on flat ground. In the follow-up, the multisensor online calibration problem on flat ground will be studied.

**Author Contributions:** Conceptualization, Y.L. and X.X.; methodology, Z.M.; software, S.Y.; validation, Y.L., and X.X.; formal analysis, Y.L.; investigation, Y.L.; resources, Y.L.; data curation, S.Y.; writing—original draft preparation, X.X.; writing—review and editing, Y.L.; visualization, S.Y.; supervision, Y.L.; project administration, Z.M.; funding acquisition, Z.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Shanghai Agriculture Applied Technology Development Program under Grant No. 2022-02-08-00-12-F01164 and China Postdoctoral Science Foundation under Grant 2021M702078.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Parekh, D.; Poddar, N.; Rajpurkar, A.; Chahal, M.; Kumar, N.; Joshi, G.P.; Cho, W. A Review on Autonomous Vehicles: Progress, Methods and Challenges. *Electronics* **2022**, *11*, 2162. [\[CrossRef\]](#)
2. Guanbei, W.; Guirong, Z. LIDAR/IMU Calibration Based on Ego-Motion Estimation. In Proceedings of the 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 18–20 December 2020.
3. 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 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
4. Wisth, D.; Camurri, M.; Das, S.; Fallon, M. Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1004–1011. [\[CrossRef\]](#)
5. Lobo, J.; Dias, J.M. Relative pose calibration between visual and inertial sensors. *Int. J. Robot. Res.* **2007**, *26*, 561–575. [\[CrossRef\]](#)

6. Alves, J.; Lobo, J.; Dias, J. Camera-inertial sensor modelling and alignment for visual navigation. *Mach. Intell. Robot. Control* **2003**, *5*, 103–112.
7. Kang, J.; Doh, N.L. Full-DOF calibration of a rotating 2-D LIDAR with a simple plane measurement. *IEEE Trans. Robot.* **2016**, *32*, 1245–1263. [CrossRef]
8. Pereira, M.; Santos, V.; Dias, P. Automatic Calibration of Multiple LIDAR Sensors Using a Moving Sphere as Target. In *Robot 2015: Second Iberian Robotics Conference*; Springer: Berlin/Heidelberg, Germany, 2016.
9. Liu, W.; Li, Z.; Malekian, R.; Sotelo, M.A.; Ma, Z.; Li, W. A novel multifeature based on-site calibration method for LiDAR-IMU system. *IEEE Trans. Ind. Electron.* **2019**, *67*, 9851–9861. [CrossRef]
10. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
11. Fleps, M.; Mair, E.; Ruepp, O.; Suppa, M.; Burschka, D. Optimization Based IMU Camera Calibration. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011.
12. Le Gentil, C.; Vidal-Calleja, T.; Huang, S. 3d Lidar-IMU Calibration Based on Upsampled Preintegrated Measurements for Motion Distortion Correction. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018.
13. Furgale, P.; Rehder, J.; Siegwart, R. Unified Temporal and Spatial Calibration for Multi-Sensor Systems. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013.
14. Rehder, J.; Beardsley, P.; Siegwart, R.; Furgale, P. Spatio-Temporal Laser to Visual/Inertial Calibration with Applications to Hand-Held, Large Scale Scanning. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014.
15. Lv, J.; Xu, J.; Hu, K.; Liu, Y.; Zuo, X. Targetless Calibration of Lidar-Imu System Based on Continuous-Time Batch Estimation. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
16. Van Nam, D.; Kim, G.-W. Online Self-Calibration of Multiple 2D LiDARs Using Line Features with Fuzzy Adaptive Covariance. *IEEE Sens. J.* **2021**, *21*, 13714–13726.
17. Ao, Y.; Chen, L.; Tschopp, F.; Breyer, M.; Siegwart, R.; Cramariuc, A. Unified Data Collection for Visual-Inertial Calibration via Deep Reinforcement Learning. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022.
18. Mair, E.; Fleps, M.; Suppa, M.; Burschka, D. Spatio-Temporal Initialization for IMU to Camera Registration. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics, Karon Beach, Thailand, 7–11 December 2011.
19. Kelly, J.; Sukhatme, G.S. A General Framework for Temporal Calibration of Multiple Proprioceptive and Exteroceptive Sensors. In *Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2014.
20. Li, M.; Mourikis, A.I. 3-D Motion Estimation and Online Temporal Calibration for Camera-IMU Systems. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.
21. Li, M.; Mourikis, A.I. Online temporal calibration for camera-IMU systems: Theory and algorithms. *Int. J. Robot. Res.* **2014**, *33*, 947–964. [CrossRef]
22. Qin, T.; Shen, S. Online Temporal Calibration for Monocular Visual-Inertial Systems. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
23. Liu, Y.; Meng, Z. Online temporal calibration based on modified projection model for visual-inertial odometry. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 5197–5207. [CrossRef]
24. Trawny, N.; Roumeliotis, S. *Indirect Kalman Filter for 3D Attitude Estimation*; Technical Report Number 2005-002; Department of Computer Science and Technology, University of Minnesota: Minneapolis, MN, USA, March 2005.
25. Muhammad, N.; Lacroix, S. Calibration of a Rotating Multi-Beam Lidar. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010.
26. Sommer, C.; Usenko, V.; Schubert, D.; Demmel, N.; Cremers, D. Efficient Derivative Computation for Cumulative B-Splines on Lie Groups. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
27. Kim, M.-J.; Kim, M.-S.; Shin, S.Y. A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, 15 September 1995.
28. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Trans. Robo.* **2016**, *33*, 1–21. [CrossRef]
29. Maas, H.-G.; Vosselman, G. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 153–163. [CrossRef]
30. Jutzi, B.; Gross, H. Normalization of LiDAR intensity data based on range and surface incidence angle. *Environ. Sci.* **2009**, *38*, 213–218.
31. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-Time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July 2014.
32. Grupp, M. evo: Python Package for the Evaluation of Odometry and Slam. 2017. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 21 June 2019).