



Article OMC-SLIO: Online Multiple Calibrations Spinning LiDAR Inertial Odometry

Shuang Wang ¹, Hua Zhang ¹ and Guijin Wang ^{2,3,*}

- ¹ Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, Southwest University of Science and Technology, Mianyang 621002, China
- ² Shanghai AI Laboratory, Shanghai 200232, China
- ³ Department of Electrical Engineering, Tsinghua University, Beijing 100089, China
- * Correspondence: wangguijin@mail.tsinghua.edu.cn

Abstract: Light detection and ranging (LiDAR) is often combined with an inertial measurement unit (IMU) to get the LiDAR inertial odometry (LIO) for robot localization and mapping. In order to apply LIO efficiently and non-specialistically, self-calibration LIO is a hot research topic in the related community. Spinning LiDAR (SLiDAR), which uses an additional rotating mechanism to spin a common LiDAR and scan the surrounding environment, achieves a large field of view (FoV) with low cost. Unlike common LiDAR, in addition to the calibration between the IMU and the LiDAR, the self-calibration odometer for SLiDAR must also consider the mechanism calibration between the rotating mechanism and the LiDAR. However, existing self-calibration LIO methods require the LiDAR to be rigidly attached to the IMU and do not take the mechanism calibration into account, which cannot be applied to the SLiDAR. In this paper, we propose firstly a novel self-calibration odometry scheme for SLiDAR, named the online multiple calibration inertial odometer (OMC-SLIO) method, which allows online estimation of multiple extrinsic parameters among the LiDAR, rotating mechanism and IMU, as well as the odometer state. Specially, considering that the rotating and static parts of the motor encoder inside the SLiDAR are rigidly connected to the LiDAR and IMU respectively, we formulate the calibration within the SLiDAR as two separate sets of calibrations: the mechanism calibration between the LiDAR and the rotating part of the motor encoder and the sensor calibration between the static part of the motor encoder and the IMU. Based on such a SLiDAR calibration formulation, we can construct a well-defined kinematic model from the LiDAR to the IMU with the angular information from the motor encoder. Based on the kinematic model, a two-stage motion compensation method is presented to eliminate the point cloud distortion resulting from LiDAR spinning and platform motion. Furthermore, the mechanism and sensor calibration as well as the odometer state are wrapped in a measurement model and estimated via an error-state iterative extended Kalman filter (ESIEKF). Experimental results show that our OMC-SLIO is effective and attains excellent performance.

Keywords: spinning LiDAR; mechanism calibration; sensor calibration; LiDAR inertial odometry; error-state iterative extended Kalman filter

1. Introduction

Given the high reliability and accuracy of light detection and ranging (LiDAR) sensors, LiDAR odometry (LO) and LiDAR simultaneous localization and mapping (SLAM) methods have played an important role in areas such as robotics, autonomous driving, and surveying [1–3]. However, the small field of view (FoV) and low vertical resolution of common LiDAR make it unsuitable for specific scenarios, such as unmanned aerial vehicles (UAVs) in indoor scenarios [4,5], autonomous ground vehicles (AGVs) in confined spaces [6,7] or uneven terrain [8,9]. Spinning LiDAR (SLiDAR), which uses an additional rotating mechanism to spin a common LiDAR and scan the surrounding environment,



Citation: Wang, S.; Zhang, H.; Wang, G. OMC-SLIO: Online Multiple Calibrations Spinning LiDAR Inertial Odometry. *Sensors* 2023, 23, 248. https://doi.org/10.3390/s23010248

Academic Editors: Henrik Hesse, Chee Kiat Seow, Yanliang Zhang, Torr Polakow and Kai Wen

Received: 31 October 2022 Revised: 19 December 2022 Accepted: 22 December 2022 Published: 26 December 2022



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/). achieves a large FoV at a low cost [10,11]. SLiDARs are widely used in the sensing part of various autonomous robots. For instance, a quadcopter, in which a 2D LiDAR is rotated by the rotor downdraft, can produce a point cloud map [12]. By continuously nodding [11] or spinning a 2D LiDAR [9], the mobile robots can model the surrounding or rough terrain. A 2D turntable equipped with 3D LIDAR, binocular vision and an infrared camera can obtain detailed unstructured terrain information for a hexapod wheel-legged robot [8].

By incorporating an inertial measurement unit (IMU), LiDAR inertial odometry (LIO) can present a better performance for localization and mapping in motion [13]. To make the implementation of LIO efficient and non-professional, self-calibration LIO methods have become a hot research topic in the related community [14–17]. Unlike the self-calibration odometer for common LiDAR that only considers the calibration between the IMU and the LIDAR, the self-calibration odometer for SLiDAR has to additionally consider the calibration between the rotating mechanism and the LIDAR due to the manufacturing and installation deviation, rotational wear and the nature of thermal expansion and contraction of the rotating mechanism [18]. Therefore, existing self-calibration LIO methods, whether filter-based [14,15] or optimization-based [16,17], do not take the mechanism calibration into account and require the LiDAR to be rigidly attached to the IMU, which cannot be implemented for SLiDAR. In addition, existing mechanism calibration methods, regarding the calibration between the rotating mechanism and the LIDAR, are offline and have very complex implementation principles that hinder their integration to self-calibration LIO [19–21].

To address the above issues, we propose firstly a novel self-calibration odometry scheme for SLiDAR, named the online multiple calibration spinning LiDAR inertial odometer (OMC-SLIO) method, which allows online estimation of multiple extrinsic parameters among the LiDAR, rotating mechanism and IMU, as well as the odometer state. Considering the rotating and static parts of the motor encoder inside the SLiDAR are rigidly connected to the LiDAR and IMU, respectively, we formulate the self-calibration within the SLiDAR as two independent sets of calibrations: the mechanism calibration between the LiDAR and the rotating part of the motor encoder and the sensor calibration between the static part of the motor encoder and the IMU. Thus, we construct a well-defined kinematic model from the LiDAR to the IMU with the angular information of the motor encoder. Based on the constructed kinematic model, a two-stage motion compensation method is presented to remove point cloud rotation and motion distortion simultaneously. Finally, we wrap the mechanism calibration, sensor calibration and the odometer state in a measurement model and perform online estimation based on the ESIEKF framework. Experimental results show that our OMC-SLIO is effective and attains excellent performance.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 formulates the problem. Section 4 introduces the proposed OMC-SLIO method in detail. In Section 5, experimental results are presented. We analyze the reason in Section 6. Finally, Section 7 summarizes the work of this paper.

2. Related Works

There are plenty of works regarding diverse calibrations among IMUs, cameras and LiDAR, and here we focus on the extrinsic calibration between LiDAR and the IMU and the mechanism calibration of SLiDAR.

2.1. Extrinsic Calibration between LiDAR and IMU

There are some tailored offline calibration methods for the LiDAR-IMU system. Ref. [22] uses pre-integration over upsampled IMU measurements derived from a Gaussian process (GP) to remove motion distortion and then combines the factors of IMU preintegration and LiDAR point-to-plane distances to calibrate extrinsic parameters. A fusion method, in which the ICP and ISPKF are respectively used to determine the unknown transformation and estimate the time delay between the LiDAR and IMU, is presented in [23]. Without any artificial targets or specific facilities, Ref. [24] proposes a multi-feature-based field calibration method for LiDAR-IMU systems that combines point/sphere, line/cylinder and planar features to reduce labor intensity. To associate laser points with stable environmental objects, Ref. [25] adopts a continuous-time IMU trajectory, modeled using GP regression, and segments a point map as structured planes to calibrate LiDAR and an IMU with on-manifold batch optimization. Ref. [26] adopts the B-spline over IMU measurements to formulate a continuous-time trajectory for fusing high-rate scanning points as [25], and uses the point-to-surfel constraint to calibrate parameters. Based on an EKF framework, Ref. [27] utilizes a discrete-time IMU state propagation model to compensate for motion distortion and proposes a motion-based constraint to refine the estimated states. Based on [26,27], Ref. [28] combines Hausdorff distance between the local trajectories and hand-eye calibration to solve the initial spatiotemporal relationship. Then, the IMU pre-integration and the point, line and plane features of the point cloud are wrapped into the objective function. Ref. [29] proposes an informative path planner to find the admissible path for ensuring accurate calibration. However, these offline calibration methods require that the IMU is rigidly attached to LiDAR and cannot be met by SLiDAR.

Regarding online extrinsic calibration in LIO, related works are around the tightlycoupled scheme based on a filter or a nonlinear optimization. Both LINS [14] and FAST-LIO [15] design an error-state iterated Kalman filter (ESIKF) to fuse the IMU and LiDAR. LINS estimates the relative pose between two consecutive local frames for updating the global pose estimate. FAST-LIO nevertheless adopts a scan-to-submap registration with an efficient Kalman gain computation. Besides filter-based methods, nonlinear optimization has been prevailing recently due to its better accuracy. LIOM [16] tightly couples LiDAR and IMU by jointly minimizing their costs in a local fixed-size window and adds an extra rotation constraint to further refine estimation. LIO-SAM [17] also formulates LIO as a factor graph which is akin to LIOM but adopts a keyframe strategy for performing real-time estimation. Although these methods can estimate the extrinsic parameter between the LiDAR and IMU, they also require that the IMU is rigidly attached to LiDAR and does not take mechanism calibration into account.

2.2. Mechanism Calibration of SLiDAR

As for the mechanism calibration of SLiDAR, some traditional calibration methods require ad-hoc tools [19,30,31] or regular calibrated scenes [18,32]. There has also been a lot of effort made to relieve calibration in targetless scenes [20,21,33–37].

For calibration in a more general scene, a point-point constraint scheme is utilized [33–35]. Based on the random sample consensus (RANSAC) method, Refs. [18,32] extract planes from the whole scene points to construct point-plane constraints for more accurate calibration. Some methods break the calibrated scene into small regions for extracting trivial planes. Refs. [36,37] search for their neighbor points within a suitable radius with the down-sampling scene point cloud. Refs. [20,21] divide the scene into small grids. In each small region, Ref. [20] adopts principal component analysis (PCA) to choose the planes for building point-plane constraints. With the estimated approximated probabilities, Ref. [21] pre-selects the valuable grids to extract planes based on RANSAC. The above methods for SLiDAR mechanism calibration are offline, and their complex principle cannot be directly applied for real-time self-calibration SLIO. In addition, they do not resolve the sensor calibration.

Different from the work in Sections 2.1 and 2.2, our method can aggregate both the mechanism and sensor extrinsic parameters as well as the odometry state into the ESIEKF framework to realize the online multiple extrinsic calibrations and SLiDAR inertial odometry.

3. Problem Formulation

Considering that the rotation and static parts of the encoder are rigidly attached to the LiDAR and IMU, respectively, we can formulate the multiple calibrations inside the SLiDAR platform as two independent sets of calibrations; one is between the LiDAR and the rotating part of the encoder (it exactly corresponds to the rotating mechanism and is therefore named mechanism calibration), and the other is between the static part of encoder and the IMU (named sensor calibration). To illustrate the formulation of multiple calibrations, the internal coordinate relation for SLiDAR is shown in Figure 1.



Figure 1. Internal coordinate relation for SLiDAR.

The subscripts of *L*, *S*, *B* and *I* respectively denote the LiDAR, the rotation part of the encoder, the static part of the encoder and the IMU. The corresponding coordination systems are denoted as \mathcal{F}_L , \mathcal{F}_S , \mathcal{F}_B and \mathcal{F}_I . In addition, the global frame is denoted as \mathcal{F}_G . A scanning LiDAR point in \mathcal{F}_L , \mathcal{F}_S , \mathcal{F}_B , \mathcal{F}_I and \mathcal{F}_G is respectively denoted as Lp , Sp , Bp , Ip and Gp .

Obviously, \mathcal{F}_B and \mathcal{F}_I are rigidly attached to the static part of the SLiDAR platform, and \mathcal{F}_L and \mathcal{F}_S are rigidly attached to the rotation part of the SLiDAR platform. The X coordinates of \mathcal{F}_S and \mathcal{F}_B are collinear, \mathcal{F}_S and \mathcal{F}_B will be exactly coincident, while the spinning angle φ of the encoder is 0 rad. The relative pose ${}^{I}T_B$ between \mathcal{F}_B and \mathcal{F}_I is the sensor calibration denoting the relative pose from the encoder to the IMU, and the relative pose ${}^{S}T_L$ between \mathcal{F}_L and \mathcal{F}_S is the mechanism calibration denoting the relative pose from LiDAR to the mounting point of the rotation mechanism. As a result, ${}^{I}T_B$ and ${}^{S}T_L$ are the interested multiple extrinsic parameters that we need to be estimated online in LIO.

In addition, as our SLiDAR system is unidirectional, mechanism calibration parameters along the spinning direction cannot be estimated due to unobservability [20]. Furthermore, the range bias of LiDAR might be several centimeters, which is much larger than the mechanism bias, so the translation parts of mechanism extrinsic parameters cannot be estimated in a non-specific scene [9]. Given the above reasons, we simplify mechanism calibration as only two rotational parameters ${}^{S}\omega_{L}(\omega_{y}, \omega_{z})$ in practice, in which both the rotational parameter in X coordination and all the translational parameters are set as zero. It should be noted that we still use ${}^{S}T_{L}$ to act as mechanism calibration later for clarity.

4. Method

4.1. Overall Pipeline

For SLiDAR, the overlap between the consecutive LiDAR scanning frame is minor due to the rapid LiDAR rotation; we therefore adopt a scan-to-submap match approach to ensure enough associations. Meanwhile, built upon an efficient ESIEKF framework [38], we aggregate both mechanism and sensor calibrations as well as odometry poses into one measurement model to pursue real-time performance. The overall pipeline of the proposed OMC-SLIO is shown in Figure 2, which mainly contains four parts: preprocessing, global transformation, ESIEKF and global mapping. The two-stage motion compensation acts as part of the global transformation.



Figure 2. The overall pipeline of proposed OMC-SLIO.

In preprocessing, the planar points are extracted from LiDAR scanning points, the spinning angles are obtained from the encoder, and the IMU states are propagated upon every IMU measurement. In global transformation, extracted planar points are projected to global space via successive transformations consisting of mechanism calibration, motion compensation, sensor calibration and IMU global poses. Specifically, motion compensation covers two kinds of distortions from LiDAR spinning and platform moving. The global planar points are used to associate the global planes in the submap of global mapping. Then, the residuals derived from point-to-plane constraints are passed into ESIEKF to update states. If converged, global mapping and odometry output will be updated, otherwise, global transformation and feature association will be executed again with updated states.

In the following, we will introduce more details on preprocessing, global transformation and ESIEKF; readers can refer to related works for details of global mapping [39].

4.2. Preprocessing

The frequencies of the data received from the LiDAR, encoder and IMU are different; for instance, the LiDAR is 10Hz and both the encoder and IMU are 200 Hz in our SLiDAR platform. Thus, before preprocessing, the raw encoder spinning angle, IMU angular rate and IMU acceleration will be temporarily stored in the respective buffers, until receiving one frame of the LiDAR scanning points is completed.

For the raw LiDAR scanning points, the planar points will first be extracted as in LOAM [40], and the extracted planar points will then be projected to the global space. Both the raw spinning angle and IMU data are first utilized in the two-stage motion compensation as introduced in Section 4.3.1. In addition, each angular rate and acceleration of IMU data in the buffer will be propagated to predict the ego-motion states; the detail of state propagation will be introduced in Section 4.4.2.

Note that as the timestamp of the latest rotation angle may be earlier than the timestamp at the end of the current LiDAR scan, this will result in some backward points not being able to be interpolated to correspond to the rotation angle. Therefore, we utilize the accumulated spinning angle measurements to estimate the current spinning rate in the preprocessing. Denote ω_{φ}^{N-1} and ω_{φ}^{N} as the last and current spinning rate, respectively,

$$\omega_{\varphi}^{N} = \frac{\omega_{\varphi}^{N-1} S_{t}^{N-1} + \varphi_{N}}{S_{t}^{N-1} + \Delta\varphi}, \ S_{t}^{N-1} = \sum_{i=1}^{i=N-1} \varphi^{i},$$
(1)

where $\Delta \varphi$ is the measurement period, S_t^{N-1} is the sum of the last N-1 measured spinning angles, and φ_N is the latest spinning angle. Based on the estimated spinning rate and the time difference, each point in the tail of LiDAR scanning can be integrated into an accurate spinning angle.

4.3. Global Transformation

For associating planar features in the submap, we need to project the raw point ${}^{L}p$ in the LiDAR frame into the global point ${}^{G}p$ with global transformation. In addition, due to the LiDAR's continuous spinning and the platform's moving, all the points received in a scanning period are not in the identity coordination, and thus, we need to transform all the points to the end-time of scanning via motion compensation. According to the coordinate relation of SLiDAR as shown in Figure 1 and considering the motion compensation, the global transformation of SLiDAR contains five consecutive transformations: extrinsic mechanism parameter, LiDAR spinning compensation, platform moving compensation, extrinsic sensor parameter and global IMU pose.

Denote L_{k-1} as the end of the last LiDAR scanning and L_k as the end of current LiDAR scanning. Assume ${}^{S_i}p_i$ is the point in the frame \mathcal{F}_S , which is projected from the original ${}^{L_i}p_i$ in \mathcal{F}_L with the extrinsic mechanism parameter ${}^{S}T_L$,

$$^{S_i}\boldsymbol{p}_i = {}^{S}\boldsymbol{T}_L{}^{L_i}\boldsymbol{p}_i. \tag{2}$$

Furthermore, denote $S_k p_i$ as the point of end time L_k in the frame of the rotation part of the encoder \mathcal{F}_S and $T_X(\varphi_s)$ as LiDAR spinning distortion compensation with the corresponding angle φ_s (derived in Section 4.3.1.1),

$$S_k \boldsymbol{p}_i = \boldsymbol{T}_X(\varphi_s)^{S_i} \boldsymbol{p}_i \tag{3}$$

Then, denote ${}^{B_i}p_i$ as the point in the frame of the static part of the encoder \mathcal{F}_B and $T_X(\varphi_a)$ as the transformation that projects ${}^{S_k}p_i$ into the frame \mathcal{F}_B with φ_a angle (derived in Section Removing LiDAR Spinning Distortion of 4.3.1),

$$B_i \boldsymbol{p}_i = \boldsymbol{T}_X(\varphi_a)^{S_k} \boldsymbol{p}_i$$
 (4)

Furthermore, assume that ${}^{B_k}p_i$ is the point of end time L_k in the frame \mathcal{F}_B and kT_i is the distortion compensation of platform moving,

B

$$\mathcal{P}_k \boldsymbol{p}_i = {}^k \boldsymbol{T}_i {}^{\boldsymbol{B}_i} \boldsymbol{p}_i. \tag{5}$$

Finally, denote ${}^{G}\boldsymbol{R}_{I}$ and ${}^{G}\boldsymbol{p}_{I}$ as the attitude and position of the IMU in the global frame respectively, the corresponding global point ${}^{G_{i}}\boldsymbol{p}_{i}$ is

$${}^{G_i}\boldsymbol{p}_i = {}^{G}\boldsymbol{R}_I{}^{I}\boldsymbol{T}_B{}^{B_k}\boldsymbol{p}_i + {}^{G}\boldsymbol{p}_I.$$
(6)

In the following, we will derive the details of the two-stage motion compensation, LiDAR spinning compensation and platform moving compensation, respectively.

4.3.1. The Two-Stage Motion Compensation

Different from the common LiDAR odometry, which only resolves the moving distortion, SLiDAR will incur the extra LiDAR spinning distortion. Based on the coordinate relation of SLiDAR in Figure 1, the bottom-up process of the two-stage motion compensation, which removes the LiDAR spinning distortion and platform moving distortion in turn, is shown in Figure 3.



Figure 3. The two-stage motion compensation. The green and yellow dots represent the sequence of LiDAR scan points before LiDAR spinning and platform moving compensation respectively. The purple dots represent the measured spinning angle sequence. The brown dots represent the sequence of measured IMU data.

4.3.1.1. Removing LiDAR Spinning Distortion

First, we interpolate the measured spinning angles to get the corresponding spinning angle of the LiDAR point, then rotate the point to the frame \mathcal{F}_B with the relative angle difference.

Assume the sample period of the spinning angle is $\Delta \gamma$, the end time of current scanning is ρ_k , the sample time for ${}^{S_i}p_i$ is ρ_i , and the spinning rate ω_{φ} is constant during a sample period $\Delta \gamma$. In Figure 3, $\rho_i \in [\gamma_{h-1}, \gamma_h]$, in which γ_{h-1} and γ_h are the adjacent time of spinning angle measurement. Denote the measured spinning angles at the time γ_{h-1} and γ_h are φ_{h-1} and φ_h , respectively. With the linear interpolation, the spinning angle at the time ρ_i is

$$\varphi_i = \frac{\gamma_h - \rho_i}{\Delta \gamma} \varphi_{h-1} + \frac{\rho_i - \gamma_{h-1}}{\Delta \gamma} \varphi_h \tag{7}$$

If ρ_i is larger than the latest time γ_h , we can utilize the estimated spinning rate ω_{φ} in time γ_h to predict the spinning angle corresponding to ρ_i ,

$$\varphi_i = \varphi_{h-1} + (\rho_i - \gamma_{h-1})\omega_{\varphi} \tag{8}$$

Since LiDAR spinning is only along the X-axis, so

$$T_X(\varphi_s) = T_X(\varphi_i - \varphi_k) = [\mathbf{R}_X(\varphi_i - \varphi_k), \mathbf{0}]$$
(9)

Similarly, we can get the spinning angle φ_k corresponding to ρ_k . Denote the measured spinning angle corresponding to the frame \mathcal{F}_B is φ_0 , we can derive

$$\boldsymbol{T}_{\boldsymbol{X}}(\varphi_a) = \boldsymbol{T}_{\boldsymbol{X}}(\varphi_k - \varphi_0) = [\boldsymbol{R}_{\boldsymbol{X}}(\varphi_k - \varphi_0), \, \boldsymbol{0}] \tag{10}$$

Notably, $R_X(\varphi)$ in both formulations (9) and (10) is defined as:

$$\boldsymbol{R}_{X}(\boldsymbol{\varphi}) = \begin{bmatrix} 1 & 0 & 0\\ 0 & \cos \varphi & -\sin \varphi\\ 0 & \sin \varphi & \cos \varphi \end{bmatrix}$$
(11)

From formulations (3), (4), (9) and (10), we can get

$${}^{B_i}\boldsymbol{p}_i = \boldsymbol{T}_X(\varphi_a)\boldsymbol{T}_X(\varphi_s)^{S_i}\boldsymbol{p}_i = \boldsymbol{R}_X(\varphi_k - \varphi_0)\boldsymbol{R}_X(\varphi_i - \varphi_k)^{S_i}\boldsymbol{p}_i = \boldsymbol{R}_X(\varphi_i - \varphi_0)^{S_i}\boldsymbol{p}_i$$
(12)

which means that we can group $T_X(\varphi_s)$ and $T_X(\varphi_a)$ as encapsulated LiDAR spinning distortion compensation.

4.3.1.2. Removing Platform Moving Distortion

Regarding platform moving compensation, the back-propagation of IMU measurements is employed to get the related pose for each point, and the point is then transformed to the scanning end-time with its corresponding related pose [15]. Different from the existing LIO methods which regard the point projected by motion compensation as the raw point for the measurement model, since the moving distortion transformation ${}^{k}T_{i}$ is the middle parameter in the global transformation for SLiDAR, we need to record the corresponding ${}^{k}T_{i}$ for raw point ${}^{L_{i}}p_{i}$ as part of the measurement model.

Performing the forward propagation on IMU inputs (in Section 4.4.2), we can get a predicted IMU pose chain until the time at the LiDAR scanning end L_k , in which we denote ${}_{I}^{G} \hat{x}_{I_k}$ as the predicted state. Assume the time ρ_i of point ${}^{B_i} p_i$ located in IMU time interval, that is $\rho_i \in [t_{j-1}, t_j]$, as shown in Figure 3. By querying the predicted IMU pose chain, we can get the IMU pose ${}_{I}^{G} \hat{x}_{I_j}$ at time t_j . Then, performing the back-propagation from the initial IMU state ${}_{I}^{G} \hat{x}_{I_j}$, we can derive the corresponding IMU state ${}_{I}^{G} \check{x}_{I_i}$ for ${}^{B_i} p_i$ with the time interval $t_j - \rho_i$. As for the implementation detail of back-propagation, the interested reader can refer to [15].

We denote the back-propagated IMU poses at the time ρ_i as $({}^{G}\check{R}_{I_i}, {}^{G}\check{p}_{I_i}) \in {}^{G}_{I}\check{x}_{I_i}$, the front-propagated IMU poses at the end of L_k as $({}^{G}\widehat{R}_{I_k}, {}^{G}\widehat{p}_{I_k}) \in {}^{G}_{I}\widehat{x}_{I_k}$ and the predicted sensor calibration as ${}^{I}\widehat{T}_B({}^{I}\widehat{R}_B, {}^{I}\widehat{t}_B)$, respectively. As a result, projecting ${}^{B_i}p_i$ to ${}^{B_k}p_i$ as

$${}^{B_{k}}\boldsymbol{p}_{i} = {}^{I}\widehat{\boldsymbol{R}}_{B}^{T} \Big({}^{G}\widehat{\boldsymbol{R}}_{Ik}^{T} \Big({}^{G}\check{\boldsymbol{R}}_{Ii} \Big({}^{I}\widehat{\boldsymbol{R}}_{B}{}^{B_{i}}\boldsymbol{p}_{i} + {}^{I}\widehat{\boldsymbol{t}}_{B}\Big) + {}^{G}\check{\boldsymbol{p}}_{Ii} - {}^{G}\widehat{\boldsymbol{p}}_{Ik}\Big) - {}^{I}\widehat{\boldsymbol{t}}_{B}\Big)$$
(13)

we can derive moving distortion compensation ${}^{k}T_{i}({}^{k}R_{i}, {}^{k}t_{i})$ as

$${}^{k}\boldsymbol{R}_{i} = {}^{I}\widehat{\boldsymbol{R}}_{B}^{TG}\widehat{\boldsymbol{R}}_{Ik}^{TG}\widehat{\boldsymbol{K}}_{I_{i}}^{TG}\widehat{\boldsymbol{R}}_{B}, \qquad (14)$$

$${}^{k}\boldsymbol{t}_{i} = {}^{I}\widehat{\boldsymbol{R}}_{B}^{T} \Big({}^{G}\widehat{\boldsymbol{R}}_{Ik}^{T}{}^{G}\check{\boldsymbol{R}}_{I_{i}} - \boldsymbol{I}\Big){}^{I}\widehat{\boldsymbol{t}}_{B} + {}^{I}\widehat{\boldsymbol{R}}_{B}^{TG}\widehat{\boldsymbol{R}}_{Ik}^{T} \Big({}^{G}\check{\boldsymbol{p}}_{I_{i}} - {}^{G}\widehat{\boldsymbol{p}}_{Ik}\Big).$$
(15)

For general IMUs, the propagated values ${}^{G}\check{p}_{I_{i}}$ and ${}^{G}\hat{p}_{I_{k}}$ are not reliable, and the related term ${}^{G}\check{p}_{I_{i}} - {}^{G}\hat{p}_{I_{k}}$ in formulation (15) is also usually trivial, so we ignore this term in the initial motion compensation. In the end, estimated odometry results will be used to compensate for distortion again with formulations (14) and (15) while finishing optimization. For pursuing real-time performance, we do not update formulation (14) and (15) with recorrect states during iterative optimization.

4.4. ESIEKF

With the ESIEKF optimization framework [38], we aggregate mechanism calibration, sensor calibration and IMU states into one measurement model to build our OMC-SLIO. Specifically, we will introduce the main contents of ESIEKF implementation on OMC-SLIO, which includes the state transition model, forward propagation and ESIEKF update.

4.4.1. State Transition Model

We take the IMU frame \mathcal{F}_I as the body frame and the first body frame of \mathcal{F}_I as the global frame \mathcal{F}_G . The true state is defined as

$$\triangleq \begin{bmatrix} G \mathbf{x}_I & I \mathbf{R}_B & I \mathbf{t}_B & S \boldsymbol{\omega}_L \end{bmatrix}^T$$
(16)

in which ${}^{I}\mathbf{R}_{B}$ and ${}^{I}\mathbf{t}_{B}$ are respectively the rotation and linear parts of sensor calibration, ${}^{S}\boldsymbol{\omega}_{L} = \begin{bmatrix} S \\ L \end{pmatrix} {}^{T}\boldsymbol{\omega}_{y} & S \\ L \end{pmatrix} {}^{T}$ is the interested rotational parameter vector of mechanism calibration as discussed in Section 3, and ${}^{G}_{I}\mathbf{x}_{I}$ denotes the IMU state. ${}^{G}_{I}\mathbf{x}_{I}$ is defined as

$${}^{G}_{I}\boldsymbol{x}_{I} \triangleq \begin{bmatrix} {}^{G}\boldsymbol{R}_{I} & {}^{G}\boldsymbol{p}_{I} & {}^{G}\boldsymbol{v}_{I} & \boldsymbol{b}_{\omega} & \boldsymbol{b}_{a} & {}^{G}\boldsymbol{g} \end{bmatrix}^{T}$$
(17)

where ${}^{G}R_{I}$, ${}^{G}p_{I}$, ${}^{G}v_{I}$ and ${}^{G}g$ denote the IMU attitude, position, velocity and gravity vector in the global frame \mathcal{F}_{G} , respectively, \boldsymbol{b}_{ω} is the IMU gyroscope bias, and \boldsymbol{b}_{a} is the IMU accelerometer bias. The continuous time kinematic model for state \boldsymbol{x} is:

$$\overset{G}{\mathbf{R}}_{I} = {}^{G}\mathbf{R}_{I} \lfloor \boldsymbol{\omega}_{m} - \boldsymbol{b}_{\omega} - \boldsymbol{n}_{\omega} \rfloor_{x}, \ \overset{G}{\mathbf{p}}_{I} = {}^{G}\boldsymbol{v}_{I}, \ \overset{G}{\mathbf{v}}_{I} = {}^{G}\mathbf{R}_{I}(\boldsymbol{a}_{m} - \boldsymbol{b}_{a} - \boldsymbol{n}_{a}) + {}^{G}\boldsymbol{g}$$

$$\overset{b}{\boldsymbol{b}}_{\omega} = \boldsymbol{n}_{\boldsymbol{b}\omega}, \ \overset{b}{\boldsymbol{b}}_{a} = \boldsymbol{n}_{\boldsymbol{b}a}, \ \overset{G}{\boldsymbol{g}} = \boldsymbol{0}, \ \overset{I}{\mathbf{R}}_{B} = \boldsymbol{0}, \ \overset{I}{\boldsymbol{t}}_{B} = \boldsymbol{0}, \ \overset{S}{\boldsymbol{\omega}}_{L} = \boldsymbol{0}$$

$$(18)$$

where ω_m and a_m are the raw IMU measurements, n_{ω} and n_a are the measurement white noise of ω_m and a_m , and both IMU bias b_{ω} and b_a are modeled as random walk and corrupted by Gaussian noise $n_{b\omega}$ and n_{ba} . Note that here we only consider two rotation parameters, ${}_{L}^{S}\omega_y$ and ${}_{L}^{S}\omega_z$, of mechanism calibration ${}^{S}T_L$. Denote ${}^{S}R_L$ and ${}^{S}t_L$ as the rotation and translation parts of ${}^{S}T_L$ respectively, then

$$\mathbf{R}_{L} = \mathbf{R}_{z} \begin{pmatrix} s \\ L \end{pmatrix} \mathbf{W}_{z} \\ \mathbf{R}_{z} \begin{pmatrix} s \\ L \end{pmatrix} \mathbf{W}_{z} \end{pmatrix} = \begin{bmatrix} cos_{L}^{S} \omega_{z} & sin_{L}^{S} \omega_{z} & 0 \\ -sin_{L}^{S} \omega_{z} & cos_{L}^{S} \omega_{z} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{R}_{y} \begin{pmatrix} s \\ L \end{pmatrix} \mathbf{W}_{y} = \begin{bmatrix} cos_{L}^{S} \omega_{y} & 0 & -sin_{L}^{S} \omega_{y} \\ 0 & 1 & 0 \\ sin_{L}^{S} \omega_{y} & 0 & cos_{L}^{S} \omega_{y} \end{bmatrix}$$
(19)

With the zero-order hold discretization, the discrete state transition model at the *i*-th IMU measurement can be expressed as [38]:

$$\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus \left(\Delta t \mathbf{f}(\mathbf{x}_i, \ \mathbf{u}_i, \ \mathbf{w}_i) \right) \tag{20}$$

where \boxplus is encapsulation operation "boxplus" as defined in [41], Δt is the IMU sample period, and the discrete kinematics function **f**, input *u*, and process noise *w* are respectively defined as:

$$\boldsymbol{u} \triangleq \begin{bmatrix} \boldsymbol{\omega}_m & \boldsymbol{a}_m \end{bmatrix} \tag{21}$$

$$\boldsymbol{w} \triangleq \begin{bmatrix} \boldsymbol{n}_{\omega} & \boldsymbol{n}_{a} & \boldsymbol{n}_{b\omega} & \boldsymbol{n}_{ba} \end{bmatrix}.$$
(22)

$$\mathbf{f}(\mathbf{x}_{i}, \mathbf{u}_{i}, \mathbf{w}_{i}) = \begin{bmatrix} \boldsymbol{\omega}_{m_{i}} - \boldsymbol{b}_{\omega_{i}} - \boldsymbol{n}_{\omega_{i}} \\ {}^{G} \boldsymbol{v}_{I_{i}} + \frac{1}{2} \begin{pmatrix} {}^{G} \boldsymbol{R}_{I_{i}} (\boldsymbol{a}_{m_{i}} - \boldsymbol{b}_{a_{i}} - \boldsymbol{n}_{a_{i}}) + {}^{G} \boldsymbol{g}_{i} \\ {}^{G} \boldsymbol{R}_{I_{i}} (\boldsymbol{a}_{m_{i}} - \boldsymbol{b}_{a_{i}} - \boldsymbol{n}_{a_{i}}) + {}^{G} \boldsymbol{g}_{i} \\ \boldsymbol{n}_{b\omega_{i}} \\ \boldsymbol{n}_{ba_{i}} \\ \boldsymbol{0}_{11 \times 1} \end{bmatrix}.$$
(23)

4.4.2. Forward Propagation

Assume the optimal estimated state of the last LiDAR scanning L_{k-1} at the end time is \overline{x}_{k-1} with covariance \overline{P}_{k-1} , which represents the covariance of the error state defined below:

$$\widetilde{\boldsymbol{x}}_{k-1} \triangleq \boldsymbol{x}_{i+1} - \overline{\boldsymbol{x}}_{k-1} = \begin{bmatrix} \delta^{G} \boldsymbol{\theta}_{I} & {}^{G} \widetilde{\boldsymbol{p}}_{I} & {}^{G} \widetilde{\boldsymbol{v}}_{I} & \widetilde{\boldsymbol{b}}_{\omega} & \widetilde{\boldsymbol{b}}_{a} & {}^{G} \widetilde{\boldsymbol{g}} & \delta^{I} \boldsymbol{\theta}_{B} & {}^{I} \widetilde{\boldsymbol{t}}_{B} & {}^{S} \widetilde{\boldsymbol{\omega}}_{L} \end{bmatrix}^{T}$$
(24)

where $\delta^G \theta_I = Log({}^{G}\overline{R}_{I}^{TG}R_{I})$ is the IMU attitude error, $\delta^I \theta_B = Log({}^{I}\overline{R}_{B}^{TI}R_{B})$ is the error of the rotation part of sensor calibration, and the rests are the standard errors like the form of $\tilde{x} = x - \bar{x}$. Here, $\delta\theta$ vector presents the small deviation between the true and the estimated rotations in the tangent space; it has a minimal representation of three degrees of freedom.

Upon each IMU input u_i , forward propagation is performed to predict the state \hat{x}_{i+1} by setting the process noise w_i to zero, and the covariance \hat{P}_{i+1} for $\tilde{x}_{i+1} \triangleq x_{i+1} \boxminus \hat{x}_{i+1} (\boxminus \text{ is also encapsulation operation "boxminus" as defined in [41]) is updated,$

$$\widehat{x}_{i+1} = \widehat{x}_i \boxplus (\Delta t \mathbf{f}(\widehat{x}_i, u_i, \mathbf{0})), \ \widehat{x}_0 = \overline{x}_{k-1},$$
(25)

$$\widehat{P}_{i+1} = F_{\widetilde{x}} \widehat{P}_i F_{\widetilde{x}}^T + F_w Q F_{w'}^T, \ \widehat{P}_0 = \overline{P}_{k-1},$$
(26)

$$F_{\widetilde{x}} = \left. \frac{\partial \widetilde{x}_{i+1}}{\partial \widetilde{x}_i} \right|_{\widetilde{x}_i = \mathbf{0}, w_i = \mathbf{0}}, F_w = \left. \frac{\partial \widetilde{x}_{i+1}}{\partial w_i} \right|_{\widetilde{x}_i = \mathbf{0}, w_i = \mathbf{0}}$$
(27)

where Q is the covariance of w, and the detailed derivation of $F_{\tilde{x}}$ and F_w can be found in [38]. Note that the forward propagation is from time t_{k-1} (the end time of the last LiDAR scanning L_{k-1}) to t_k (the end time of the current LiDAR scanning L_k). Denote the propagated state and covariance at t_k as \hat{x}_k and \hat{P}_k , and the error state $\tilde{x}_k = x_k \Box \hat{x}_k$ complies with the prior distribution,

$$\widetilde{\mathbf{x}}_k \sim \mathcal{N}\left(\mathbf{0}, \, \widehat{\mathbf{P}}_k\right)$$
(28)

4.4.3. ESIEKF Update

Denote the current iteration of IEKF as v, and the corresponding estimated and error states are \hat{x}_k^v and \tilde{x}_k^v , respectively. In addition, $\hat{x}_k^v = \hat{x}_k$ (if v = 0) and $x_k = \hat{x}_k^v \boxplus \tilde{x}_k^v$. Define the true planar feature point as

$${}^{L}\boldsymbol{p}_{i}^{gt} = {}^{L}\boldsymbol{p}_{i} - {}^{L}\boldsymbol{n}_{i}$$
⁽²⁹⁾

where ${}^{L}n_{i}$ is the measurement noise of the LiDAR scanning point. Since ${}^{L}p_{i}^{gt}$ is a planar point, its neighbors in the global map should be on a local plane. By projecting ${}^{L}p_{i}^{gt}$ to the global frame with the true state x_{k} , the measurement model can be derived as

$$\boldsymbol{h}_{i}(\boldsymbol{x}_{k},{}^{L}\boldsymbol{n}_{i}) = 0 = G\boldsymbol{n}_{i}^{T} \left(\left({}^{G}\boldsymbol{R}_{I} \left({}^{I}\boldsymbol{R}_{B} \left({}^{k}\boldsymbol{R}_{i}\boldsymbol{R}_{X}(\varphi_{i}-\varphi_{0})^{S}\boldsymbol{R}_{L} \left({}^{L}\boldsymbol{p}_{i}-{}^{L}\boldsymbol{n}_{i} \right) + {}^{k}\boldsymbol{t}_{i} \right) + {}^{I}\boldsymbol{t}_{B} \right) + {}^{G}\boldsymbol{p}_{I} \right) - {}^{G}\boldsymbol{c}_{i} \right)$$

$$(30)$$

where ${}^{G}R_{I} \in x_{k}$, ${}^{I}R_{B} \in x_{k}$, ${}^{S}R_{L} \in x_{k}$, ${}^{I}t_{B} \in x_{k}$, and ${}^{G}p_{I} \in x_{k}$, ${}^{G}n_{i}$ and ${}^{G}c_{i}$ are the normal vector and center point of the local plane fitted in the global map.

Ignoring the measurement noise ${}^{L}n_{i}$, we can project extracted planar feature point ${}^{L}p_{i}$ to the global frame with estimated state \hat{x}_{k}^{v} ,

$${}^{G}\boldsymbol{p}_{i}^{v} = {}^{G}\widehat{\boldsymbol{R}}_{I_{k}}^{v} \left({}^{I}\widehat{\boldsymbol{R}}_{B}^{v} \left({}^{k}\boldsymbol{R}_{i}\boldsymbol{R}_{X}(\varphi_{i}-\varphi_{0})^{S}\widehat{\boldsymbol{R}}_{L}^{vL}\boldsymbol{p}_{i} + {}^{k}\boldsymbol{t}_{i} \right) + {}^{I}\widehat{\boldsymbol{t}}_{B}^{v} \right) + {}^{G}\widehat{\boldsymbol{p}}_{I_{k}}^{v}$$
(31)

where ${}^{G}\widehat{\mathbf{R}}_{I_{k}}^{v} \in \widehat{\mathbf{x}}_{k}^{v}$, ${}^{I}\widehat{\mathbf{R}}_{B}^{v} \in \widehat{\mathbf{x}}_{k}^{v}$, ${}^{S}\widehat{\mathbf{R}}_{L}^{v} \in \widehat{\mathbf{x}}_{k}^{v}$, ${}^{I}\widehat{\mathbf{t}}_{B}^{v} \in \widehat{\mathbf{x}}_{k}^{v}$ and ${}^{G}\widehat{\mathbf{p}}_{I_{k}}^{v} \in \widehat{\mathbf{x}}_{k}^{v}$. Similarly, since ${}^{L}\mathbf{p}_{i}$ is a planar point, its neighbors in the global map should be on a local plane, so we can define

$$\boldsymbol{h}_{i}(\widehat{\boldsymbol{x}}_{k}^{\upsilon},\boldsymbol{0}) = {}^{\boldsymbol{G}}\boldsymbol{n}_{i}^{\boldsymbol{T}} \Big({}^{\boldsymbol{G}}\boldsymbol{p}_{i}^{\upsilon} - {}^{\boldsymbol{G}}\boldsymbol{c}_{i} \Big) = \boldsymbol{0}$$
(32)

where ${}^{G}n_{i}$ and ${}^{G}c_{i}$ are normal vector and center point of the local plane fitted by the neighboring points of ${}^{G}p_{i}^{v}$ in the global map. As a result, we can define the residual as

$$\boldsymbol{r}_{i}^{\upsilon} \triangleq \boldsymbol{h}_{i}\left(\boldsymbol{x}_{k},{}^{L}\boldsymbol{n}_{i}\right) \boxminus \boldsymbol{h}_{i}(\widehat{\boldsymbol{x}}_{k}^{\upsilon},\boldsymbol{0}) = \boldsymbol{h}_{i}\left(\widehat{\boldsymbol{x}}_{k}^{\upsilon} \boxplus \widetilde{\boldsymbol{x}}_{k}^{\upsilon},{}^{L}\boldsymbol{n}_{i}\right) \boxminus \boldsymbol{h}_{i}(\widehat{\boldsymbol{x}}_{k}^{\upsilon},\boldsymbol{0}) \approx \boldsymbol{H}_{i}^{\upsilon}\widetilde{\boldsymbol{x}}_{k}^{\upsilon} + \boldsymbol{D}_{i}^{\upsilon L}\boldsymbol{n}_{i},$$
(33)

$$\boldsymbol{H}_{i}^{\upsilon} = \left. \frac{\delta \boldsymbol{h}_{i}(\widehat{\boldsymbol{x}}_{k}^{\upsilon} \boxplus \widetilde{\boldsymbol{x}}_{k}^{\upsilon}, \boldsymbol{0})}{\delta \widetilde{\boldsymbol{x}}_{k}^{\upsilon}} \right|_{\widehat{\boldsymbol{x}}_{k}^{\upsilon} = \boldsymbol{0}}, \boldsymbol{D}_{i}^{\upsilon} = \left. \frac{\delta \boldsymbol{h}_{i}(\widehat{\boldsymbol{x}}_{k}^{\upsilon}, {}^{L}\boldsymbol{n}_{i})}{\delta^{L}\boldsymbol{n}_{i}} \right|_{\boldsymbol{L}_{\boldsymbol{n}_{i}} = \boldsymbol{0}},$$
(34)

where the detailed derivation for H_i^v and D_i^v can be referred to in [38], and the formulation (33) defines a posteriori distribution for \tilde{x}_k ,

$$\left(\boldsymbol{D}_{i}^{\nu L}\boldsymbol{n}_{i}\right)\left|\widetilde{\boldsymbol{x}}_{k}^{\nu}=\boldsymbol{r}_{i}^{\nu}-\boldsymbol{H}_{i}^{\nu}\widetilde{\boldsymbol{x}}_{k}^{\nu}\sim\mathcal{N}\left(\boldsymbol{0},\overline{\boldsymbol{R}}_{i}\right),\ \overline{\boldsymbol{R}}_{i}=\boldsymbol{D}_{i}^{\nu}\boldsymbol{R}_{i}\boldsymbol{D}_{i}^{\nu T}.$$
(35)

After every iteration, the prior distribution of \tilde{x}_k from the forward propagation has been evolved as:

$$\widetilde{\mathbf{x}}_{k} = \mathbf{x}_{\kappa} \boxminus \widehat{\mathbf{x}}_{\kappa} = (\widehat{\mathbf{x}}_{k}^{\upsilon} \boxplus \widetilde{\mathbf{x}}_{k}^{\upsilon}) \boxminus \widehat{\mathbf{x}}_{\kappa} \approx \widehat{\mathbf{x}}_{k}^{\upsilon} \boxminus \widehat{\mathbf{x}}_{\kappa} + \mathbf{J}^{\upsilon} \widetilde{\mathbf{x}}_{k}^{\upsilon}$$
(36)

$$J^{\nu} = \left(\frac{\partial((\widehat{\mathbf{x}}_{k}^{\nu} \boxplus \widetilde{\mathbf{x}}_{k}^{\nu}) \boxminus \widehat{\mathbf{x}}_{\kappa})}{\partial \widetilde{\mathbf{x}}_{k}^{\nu}}\right)_{\widetilde{\mathbf{x}}_{k}^{\nu} = \mathbf{0}}$$
(37)

where the detailed derivation for J^{v} can be found in [38], and $\hat{x}_{k}^{v} = \hat{x}_{\kappa}$, $J^{v} = I$ while v = 0. From formulation (36), we can find the evolutive prior distribution for \tilde{x}_{k}^{v} is,

$$\widetilde{\mathbf{x}}_{k}^{v} \sim \mathcal{N}\Big(-\mathbf{J}^{v-1}(\widehat{\mathbf{x}}_{k}^{v} \boxminus \widehat{\mathbf{x}}_{\kappa}), \ \mathbf{J}^{v-1}\widehat{\mathbf{P}}_{k}\mathbf{J}^{v-T}\Big).$$
(38)

Furthermore, combine the prior in formulation (38) and the posteriori distribution in formulation (35), which leads to the maximum a-posteriori estimate (MAP) for \tilde{x}_{k}^{v} ,

$$\min_{\widetilde{\mathbf{x}}_{k}^{\upsilon}} \left(|| \widehat{\mathbf{x}}_{k}^{\upsilon} \boxminus \widehat{\mathbf{x}}_{\kappa} + \mathbf{J}^{\upsilon} \widetilde{\mathbf{x}}_{k}^{\upsilon} ||_{\widehat{\mathbf{P}}_{k}^{-1/2}}^{2} + \sum || \mathbf{r}_{i}^{\upsilon} - \mathbf{H}_{i}^{\upsilon} \widetilde{\mathbf{x}}_{k}^{\upsilon} ||_{\overline{\mathbf{R}}_{i}^{-1/2}}^{2} \right),$$
(39)

where $||x||_{\Sigma}^2 = x^T \Sigma^T \Sigma x$. Substituting the linearization of the priori in the above equation and optimizing the resultant quadratic cost leads to the standard iterated Kalman filter,

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^{T} \left(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^{T} + \overline{\boldsymbol{R}} \right)^{-1}, \tag{40}$$

$$\widehat{\mathbf{x}}_{k}^{\nu+1} = \widehat{\mathbf{x}}_{k}^{\nu} \boxplus \left(\mathbf{K}\mathbf{r}^{\nu} + (\mathbf{I} - \mathbf{K}\mathbf{H})(\mathbf{J}^{\nu})^{-1}(\widehat{\mathbf{x}}_{k}^{\nu} \boxminus \widehat{\mathbf{x}}_{k}) \right),$$
(41)

where $H = [H_1^{vT}, \dots, H_m^{vT}]^T$, $\overline{R} = diag(\overline{R}_1, \dots, \overline{R}_m)$, $P = (J^v)^{-1}\widehat{P}_k(J^v)^{-T}$, and $r^v = [r_1^{vT}, \dots, r_m^{vT}]^T$. The updated estimate \widehat{x}_k^{v+1} is then used to compute the residual again and the process is repeated until convergence. After convergence, the optimal state estimation and covariance is

$$\overline{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^{\nu+1}, \ \overline{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}.$$
(42)

5. Experiment Results

In order to verify the proposed OMC-SLIO, we designed a SLiDAR experimental platform, as shown in Figure 4, in which the hardware system can be divided into two parts: the rotating part and the static part. Specifically, the rotating part consists of Velodyne VLP-16 LiDAR, the rotating mechanism and the rotating part of a DJI GM6020 motor (including the rotating part of the embedded encoder); the static part consists of the static part of a DJI GM6020 motor (including the static part of the static part of the embedded encoder); a XSENS MTi-630 IMU, a Piesia WK310CA computer (Intel Core i78565u CPU and 16 GB RAM) and a metal square base. Note that the Velodyne VLP-16 LiDAR has been mounted on the rotating mechanism as accurately as possible to ensure that all external parameters of the mechanism such as ${}_{I}^{S}\omega_{y}$ and ${}_{I}^{S}\omega_{z}$ are approaching zero.



Figure 4. 3D SLiDAR experimental platform.

We chose the state-of-the-art (SOTA) self-calibration methods, FAST-LIO [15] and LIO-SAM [17], and the loosely coupled method, LOAM [40], as comparison methods. Since the relative pose between LiDAR and an IMU is non-constant and severe point cloud distortion results from LiDAR spinning, these comparison methods would lead to failure

when they are directly applied for SLiDAR. Therefore, based on our constructed kinematic model and the two-step point cloud distortion compensation method, we transformed the original LiDAR scanning points to the space of the static part of the encoder (i.e., \mathcal{F}_B); then, the newly formed LiDAR scanning points were fed to these comparison methods. To distinguish the original forms of these comparison methods, we term these modified FAST-LIO, LIO-SAM and LOAM for SLiDAR as FAST-SLIO, SLIO-SAM and SLOAM, respectively.

Furthermore, to validate the correctness of our formulated SLiDAR calibration, we adopted two specific offline calibration methods to estimate the mechanism external parameters [27] and sensor external parameters [21], respectively. Then, the online calibration parameters in OMC-SLIO were replaced with the corresponding external parameters from the above offline calibration. Again, we refer to such pre-calibrated OMC-SLIO as the Calib-SLIO.

We conducted two sets of comparison experiments. One set of comparison experiments was conducted in a lab room with a Nokov Motion Capture System (NMCS) (shown in Figure 5), where the positioning results of the motion capture system were used as ground truth. Another set of comparison experiments was conducted in a larger underground parking to reveal the effect of external parameter self-calibration on localization and mapping.



Figure 5. Lab room with a Nokov Motion Capture System.

5.1. Lab Room

In the lab room test, the positioning result from the NMCS is regarded as the ground truth. The absolute translation errors (ATE) presented by MAE and RMSE and the average processing time (APT) for each LiDAR scanning (10 Hz) are listed in Table 1. The mapping results of the lab room using the comparison methods are shown in Figure 6. Note that we only present the results of comparative methods that worked successfully.

In Table 1, it can be found that our OMC-SLIO, Calib-SLIO and FAST-SLIO all achieve centimeter-level localization accuracy, while both SLOAM and SLIO-SAM fail to function successfully. Since fast LiDAR spinning presents minor overlap between successive frame scanning, the failure of SLOAM and SLIO-SAM stems from their basic matching based on successive frames. Further, our OMC-SLIO, pre-calibrated Calib-SLIO and FAST-SLIO pre-processed using our proposed method all achieve reliable localization and mapping, indicating the effectiveness of our proposed SLiDAR calibration formulation and the two-step motion compensation.

Method	X-Y-Z	Calib-SLIO	SLOAM	SLIO-SAM	FAST-SLIO	OMC-SLIO
MAE (cm)	Х	2.41	Failed	Failed	4.05	3.58
	Y	2.92			4.03	3.69
	Z	4.87			8.59	5.19
RMSE (cm)	Х	4.59	Failed	Failed	5.24	5.93
	Y	4.23			5.26	5.91
	Z	5.13			9.28	5.53
APT (ms)	-	34.43	Failed	Failed	34.72	35.49

Table 1. The ATE and APT of the methods in the lab room.

Ground truth: From Nokov Motion Capture System.



(a) The whole mapping result of Calib-SLIO

(b) The local mapping result of Calib-SLIO



(c) The local mapping result of FAST-SLIO



(d) The local mapping result of OMC-SLIO

Figure 6. The mapping results of the lab room with Calib-SLIO, FAST-SLIO and OMC-SLIO. Note that the green and red dots show the strong and weak laser beam reflection intensities, respectively.

Specifically, our OMC-SLIO shows a significant improvement in Z-directional localization, with a 40% improvement in accuracy compared to FAST-SLIO. Furthermore, the Z-direction localization of all methods exhibited the largest MAE and RMSE values, owing to the weaker laser beam reflection intensity from the smooth ground (as shown in Figure 6, the green and red points indicate strong and weak laser beam reflections, respectively). For the APT metric, the average runtime of our OMC-SLIO, Calib-SLIO and FAST-SLIO is around 35 ms. This APT result is reliable because our OMC-SLIO, Calib-SLIO, and FAST-SLIO are all able to achieve similar convergence in the lab room. Consistent with quantitative results, our OMC-SLIO shows sharper mapping details than FAST-SLIO and presents comparable mapping performance with Calib-SLIO, as shown in Figure 6.

5.2. Underground Parking

In the large underground parking, we cannot use the motion capture system and cannot receive GPS positioning signals. Since Calib-SLIO got the best positioning accuracy in the lab room experiments, the positioning results of Calib-SLIO were adopted as the ground truth in the underground parking test. The absolute translation errors (ATE)

presented via MAE and RMSE and the average processing time (APT) for each LiDAR scanning (10 Hz) are presented in Table 2. The underground parking mapping results for the comparison method are shown in Figure 7. Again, we only present the results for the methods that worked successfully.

Method	X-Y-Z	SLOAM	SLIO-SAM	FAST-SLIO	OMC-SLIO
MAE (cm)	Х	Failed	Failed	10.80	5.47
	Y			15.86	7.04
	Z			25.29	15.19
RMSE (cm)	Х	Failed	Failed	12.28	6.48
	Y			16.64	8.98
	Z			30.19	20.36
Time (ms)	-	Failed	Failed	67.50	49.32

Table 2. The ATE and APT of the methods in underground parking.

Ground truth: from Calib-SLIO.



(a) The whole mapping result of Calib-SLIO



(c) The whole mapping result of FAST-SLIO



(e) The whole mapping result of OMC-SLIO



(b) The local mapping result of Calib-SLIO



(d) The local mapping result of FAST-SLIO



(f) The local mapping result of OMC-SLIO

Figure 7. The mapping results of underground parking with Calib-SLIO, FAST-SLIO and OMC-SLIO methods. Note that the green and red dots show the strong and weak laser beam reflection intensities, respectively.

There is no doubt that SLOAM and SLIO-SAM still fail when applied to SLiDAR. Specifically, compared with FAST-SLIO, our OMC-SLIO has an average improvement of

50% in positioning accuracy in the X and Y directions and 35% in the Z direction, as shown in Table 2. Similarly, since the dark-colored floor and roof can absorb most of the laser beam energy (as shown in Figure 7, green and red dots indicate strong and weak laser beam reflections, respectively), both MAE and RMSE values in the Z direction in Table 2 are worse than those in the X or Y directions. Meanwhile, OMC-SLIO also has better real-time performance than FAST-SLIO due to better estimation convergence. Since the estimation bias of the rotation part of the extrinsic calibration will lead to more severe point cloud bias in larger scenes, the FAST-SLIO experiments in the underground parking exhibit more prominent localization and mapping errors than the experiments in the lab room. Consistent with quantitative results, OMC-SLIO exhibits clearer mapping details, while FAST-SLIO presents a distinctly blurred mapping, as shown on the right in Figure 7.

6. Discussion

In this section, we discuss why our OMC-SLIO can show better performance for SLiDAR compared to the comparative self-calibration odometer method, as shown in the previous experimental section.

The best positioning accuracy using Calib-SLIO verified the practicability of the calibration formulation within the SLiDAR. To some extent, our OMC-SLIO can be considered as an online version of Calib-SLIO. Our OMC-SLIO online calculates both mechanism and sensor parameters within SLiDAR, thus achieving excellent positioning and mapping accuracy. In the set of underground parking experiments, we used the offline estimated extrinsic parameters of Calib-SLIO as the ground truth. Then, all the extrinsic parameters estimated online using FAST-SLIO and OMC-SLIO were averaged and taken as the final extrinsic parameter results. Table 3 shows the errors in the online extrinsic parameter estimation of FAST-SLIO and OMC-SLIO.

Calibration		Μ	AE	RMSE	
		FAST-SLIO	OMC-SLIO	FAST-SLIO	OMC-SLIO
Mechanism	ry (deg)	-	0.056	-	0.058
	rz (deg)	-	0.213	-	0.222
Sensor	rx (deg)	0.269	0.151	0.303	0.172
	ry (deg)	0.652	0.334	0.815	0.488
	rz (deg)	0.611	0.454	0.650	0.491
	tx (cm)	2.3	0.7	2.6	0.7
	ty (cm)	1.8	1.3	1.9	1.4
	tz (cm)	10.4	0.9	11.6	1.1

Table 3. The error of online extrinsic parameter estimation of FAST-SLIO and OMC-SLIO.

Note that FAST-SLIO estimates only the extrinsic parameters of the sensor, while our OMC-SLIO estimates both the extrinsic parameters of the mechanism and the sensor. As shown in Table 3, compared with FAST-SLIO, OMC-SLIO reduces the online calibration errors by 43%, 44%, 26%, 72%, 27% and 91% on average for the 6 sensor extrinsic parameters, respectively. Meanwhile, the average error of the two additional online mechanism calibration parameters of OMC-SLIO is only 0.057° and 0.218°. FAST-SLIO does not consider the correlation of the extrinsic parameters inside the SLiDAR, while our OMC-SLIO establishes the kinematic model relationship between multiple sensors inside the SLiDAR and, on the other hand, constructs a unified observation model with multiple extrinsic parameters and the odometry states. Therefore, OMC-SLIO can be more reliable for SLiDAR to achieve better localization and mapping performance.

7. Conclusions

In this paper, we propose a novel self-calibration odometry scheme for SLiDAR, named as online multiple calibration spinning LiDAR inertial odometer (OMC-SLIO),

which allows online estimation of multiple extrinsic parameters among the LiDAR, rotating mechanism and IMU, as well as the odometer state. To the best of our knowledge, this is the first multi-extrinsic self-calibration LIO method for SLiDAR. By analyzing the relationships of the internal sensors, we formulate the calibration inside the SLiDAR as mechanism calibration and sensor calibration and build the kinematic model with the encoder information. Further, based on the kinematic model, we present a two-stage motion compensation to eliminate the point cloud distortion caused by LiDAR rotation and platform motion. Finally, the two sets of extrinsic parameters and odometer state are wrapped in a measurement model and estimated with the error-state iterative extended Kalman filter (ESIEKF). We design a 3D SLiDAR platform and conduct experiments in a lab room and underground parking. Experimental results show that our OMC-SLIO is effective and attains excellent performance. Further, the extrinsic parameters comparison test verifies that our OMC-SLIO can better estimate the multiple external parameters of SLiDAR and shows why our OMC-SLIO can achieve higher accuracy in localization and mapping. To maintain the consistence of the localization and mapping of our proposed OMC-SLIO, we will introduce the pose graph optimization resulted from the multiple frames match or the loop closure in future work.

Author Contributions: Conceptualization, S.W., G.W. and H.Z.; methodology and validation, S.W.; writing—original draft preparation, S.W.; writing—review and editing, G.W. and H.Z.; supervision, G.W. and H.Z.; project administration, H.Z.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Science and Technology Plan Project of the Sichuan Province of China (Grant No. 2021YFG0376, No. 2022YFSY0011 and No. 2022JDRC0073).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study is not publicly available at this time but may be obtained from the authors upon reasonable request.

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

References

- Palieri, M.; Morrell, B.; Thakur, A.; Ebadi, K.; Nash, J.; Chatterjee, A.; Kanellakis, C.; Carlone, L.; Guaragnella, C.; Agha-Mohammadi, A.A. LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time. *IEEE RA-L.* 2021, *6*, 421–428. [CrossRef]
- Dubé, R.; Cramariuc, A.; Dugas, D.; Sommer, H.; Dymczyk, M.; Nieto, J.; Siegwart, R.; Cadena, C. SegMap: Segment-based mapping and localization using data-driven descriptors. *Int. J. Robot. Res.* 2020, *39*, 339–355. [CrossRef]
- Zhen, W.; Hu, Y.; Liu, J.; Scherer, S. A Joint Optimization Approach of LiDAR-Camera Fusion for Accurate Dense 3-D Reconstructions. *IEEE RA-L.* 2019, 4, 3585–3592. [CrossRef]
- Zhen, W.; Zeng, S.; Soberer, S. Robust localization and localizability estimation with a rotating laser scanner. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Marina Bay Sands, Singapore, 29 May–3 June 2017; pp. 6240–6245. [CrossRef]
- Karimi, M.; Oelsch, M.; Stengel, O.; Babaians, E.; Steinbach, E. LoLa-SLAM: Low-Latency LiDAR SLAM Using Continuous Scan Slicing. *IEEE RA-L.* 2021, 6, 2248–2255. [CrossRef]
- Ebadi, K.; Chang, Y.; Palieri, M.; Stephens, A.; Hatteland, A.; Heiden, E.; Thakur, A.; Funabiki, N.; Morrell, B.; Wood, S.; et al. LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 80–86. [CrossRef]
- Han, W.; Chen, W.; Lihua, X. Lightweight 3-D Localization and Mapping for Solid-State LiDAR. *IEEE RA-L.* 2021, 6, 1801–1807. [CrossRef]
- Zhihua, C.; Jiehao, L.; Shoukun, W.; Junzheng, W.; Liling, M. Flexible gait transition for six wheel-legged robot with unstructured terrains. *Robot. Auton. Syst.* 2022, 150, 103989. [CrossRef]
- 9. Droeschel, D.; Schwarz, M.; Behnke, S. Continuous Mapping and Localization for Autonomous Navigation in Rough Terrain Using a 3D Laser Scanner. *Robot. Auton. Syst.* 2017, *88*, 104–115. [CrossRef]
- 10. Sheehan, M.; Harrison, A.; Newman, P. Automatic Self-calibration of a Full Field-of-View 3D n-Laser Scanner. *Int. Symp. Exp. Robot.* **2014**, *79*, 165–178. [CrossRef]

- 11. Anindya, H.; Lindsay, K.; Leena, V. Coordinated Nodding of a Two-Dimensional Lidar for Dense Three-Dimensional Range Measurements. *IEEE RA-L.* 2018, *3*, 4108–4115. [CrossRef]
- 12. Lukas, K.; Robert, Z.; Michael, B. Continuous-Time Three-Dimensional Mapping for Micro Aerial Vehicles with a Passively Actuated Rotating Laser Scanner. J. Field Robot. 2016, 33, 103–132. [CrossRef]
- 13. Soloviev, A.; Bates, D.; Graas, F. Tight Coupling of Laser Scanner and Inertial Measurements for a Fully Autonomous Relative Navigation Solution. *Navigation* **2014**, *54*, 189–205. [CrossRef]
- Lv, J.; Hu, K.; Xu, J.; Liu, Y.; Ma, X.; Zuo, X. CLINS: Continuous-Time Trajectory Estimation for LiDAR-Inertial System. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech, 27 September–1 October 2021; pp. 6657–6663. [CrossRef]
- 15. Wei, X.; Fu, Z. FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter. *IEEE RA-L.* **2021**, *6*, 3317–3324. [CrossRef]
- Haoyang, Y.; Yuying, C.; Ming, L. Tightly Coupled 3D Lidar Inertial Odometry and Mapping. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3144–3150. [CrossRef]
- 17. 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; pp. 5135–5142. [CrossRef]
- Morales, J.; Martínez, J.L.; Mandow, A.; Reina, A.J.; Pequeño-Boter, A.; García-Cerezo, A. Boresight Calibration of Construction Misalignments for 3D Scanners Built with a 2D Laser Rangefinder Rotating on Its Optical Center. *Sensors.* 2014, 14, 20025–20040. [CrossRef]
- Khurana, A.; Nagla, K. An Improved Method for Extrinsic Calibration of Tilting 2D LRF. J. Intell. Rob. Syst. 2020, 99, 693–712. [CrossRef]
- Jaehyeon, K.; Nakj-Lett, D. Full-DOF Calibration of a Rotating 2-D LIDAR With a Simple Plane Measurement. *IEEE Trans. Robot.* 2016, 32, 1245–1263. [CrossRef]
- Shuang, W.; Hua, Z.; Guijin, W.; Ran, L.; Jianwen, H.; Bo, C. FGRSC: Improved Calibration for Spinning LiDAR in Unprepared Scenes. *IEEE Sens. J.* 2022, 22, 14250–14262. [CrossRef]
- 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; pp. 2149–2155. [CrossRef]
- Liu, W. LiDAR-IMU Time Delay Calibration Based on Iterative Closest Point and Iterated Sigma Point Kalman Filter. Sensors 2017, 17, 539. [CrossRef]
- Liu, W.; Li, Z.; Malekian, R.; Sotelo, M.; Ma, Z.; Li, W. A Novel Multifeature Based On-Site Calibration Method for LiDAR-IMU System. *IEEE Trans. Ind. Electron.* 2020, 67, 9851–9861. [CrossRef]
- Shuaixin, L.; Li, W.; Jiuren, L.; Bin, T.; Long, C.; Guangyun, L. 3D LiDAR/IMU Calibration Based on Continuous-Time Trajectory Estimation in Structured Environments. *IEEE Access.* 2021, *9*, 138803–138816. [CrossRef]
- Jiajun, L.; Jinhong, X.; Kewei, H.; Yong, L.; Xingxing, Z. Targetless Calibration of LiDAR-IMU System Based on Continuous-time Batch Estimation. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vesgas, NV, USA, 24 October 2020–24 January 2021; pp. 9968–9975. [CrossRef]
- Mishra, S.; Pandey, G.; Saripalli, S. Target-free Extrinsic Calibration of a 3D-Lidar and an IMU. In Proceedings of the 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 23–25 September 2021; pp. 1–7. [CrossRef]
- 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. [CrossRef]
- Usayiwevu, M.; Gentil, C.; Mehami, J.; Yoo, C.; Fitch, R.; Vidal-Calleja, T. Information Driven Self-Calibration for Lidar-Inertial Systems. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vesgas, NV, USA, 24 October 2020–24 January 2021; pp. 9961–9967. [CrossRef]
- Yuan, C.; Bi, S.; Cheng, J.; Yang, D.; Wang, W. Low-Cost Calibration of Matching Error between Lidar and Motor for a Rotating 2D Lidar. *Appl. Sci.* 2021, 11, 913. [CrossRef]
- Mario, C.; Alexander, F.; Stefan, S. Calibration of a Rotating or Revolving Platform with a LiDAR Sensor. *Appl. Sci.* 2019, *9*, 2238. [CrossRef]
- Martínez, J.L.; Morales, J.; Reina, A.J.; Mandow, A.; Pequeno-Boter, A.; García-Cerezo, A. Construction and Calibration of a Low-Cost 3D Laser Scanner with 360° Field of View for Mobile Robots. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 149–154. [CrossRef]
- Jan, O.; Lars, P.; Arne, R.; Rüdiger, D. Fast Calibration of Rotating and Swivelling 3-D Laser Scanners Exploiting Measurement Redundancies. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 3038–3044. [CrossRef]
- Sosuke, Y.; Hiroshi, H.; Shigeyuki, O.; Shan, J.; Yuichi, M. Calibration of a Rotating 2D LRF in Unprepared Scenes by Minimizing Redundant Measurement Errors. In Proceedings of the 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Munich, Germany, 3–7 July 2017; pp. 172–177. [CrossRef]
- 35. Mark, S.; Alastair, K.; Paul, N. Self-Calibration for a 3D Laser. Int. J. Rob. Res. 2012, 31, 675–687.

- 36. Hatem, A.; Brett, B. Automatic Calibration of Spinning Actuated Lidar Internal Parameters. J. Field Robot. 2015, 32, 723–747. [CrossRef]
- 37. Zeng, Y.; Yu, H.; Dai, H.; Song, S.; Lin, M.; Sun, B.; Jiang, W.; Meng, M.Q. An Improved Calibration Method for a Rotating 2D LIDAR System. *Sensors* **2018**, *18*, 497. [CrossRef]
- 38. Dongjiao, H.; Wei, X.; Fu, Z. Embedding manifold structures into Kalman filters. *arXiv preprint*. 2021. Available online: https://arxiv.org/abs/2102.03804v3 (accessed on 26 June 2021).
- Wei, X.; Yixi, C.; Dongjiao, H.; Jiarong, L.; Fu, Z. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE TRO*. 2022, 38, 2053–2073. [CrossRef]
- Ji, Z.; Sanjiv, S. LOAM: Lidar Odometry and Mapping in Real-time. In Proceedings of the 2014 International Conference on Robotics Science and Systems (RSS), Hall, Sheffield, UK, 1–4 September 2014.
- 41. Hertzberg, C.; Wagner, R.; Frese, U.; Schröder, L. Integrating Generic Sensor Fusion Algorithms with Sound State Representations through Encapsulation of Manifolds. *Inf. Fusion.* **2013**, *14*, 57–77. [CrossRef]

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