Observability-Aware Intrinsic and Extrinsic Calibration of LiDAR-IMU Systems

Jiajun Lv, Xingxing Zuo, Kewei Hu, Jinhong Xu, Guoquan Huang, *Senior Member, IEEE*, and Yong Liu

***Abstract*—Accurate and reliable sensor calibration is essential to fuse LiDAR and inertial measurements, which are usually available in robotic applications. In this article, we propose a novel LiDAR-IMU calibration method within the continuous-time batch-optimization framework, where the intrinsics of both sensors and the spatial-temporal extrinsics between sensors are calibrated without using calibration infrastructure, such as fiducial tags. Compared to discrete-time approaches, the continuous-time formulation has natural advantages for fusing high-rate measurements from LiDAR and IMU sensors. To improve efficiency and address degenerate motions, the following two observability-aware modules are leveraged: first, The information-theoretic data selection policy selects *only* the most informative segments for calibration during data collection, which significantly improves the calibration efficiency by processing only the selected informative segments. Second, the observability-aware state update mechanism in nonlinear least-squares optimization updates *only* the identifiable directions in the state space with truncated singular value decomposition, which enables accurate calibration results even under degenerate cases where informative data segments are not available. The proposed LiDAR-IMU calibration approach has been validated extensively in both simulated and real-world experiments with different robot platforms, demonstrating its high accuracy and repeatability in commonly-seen human-made environments.**

***Index Terms*—Continuous-time representation, information matrix, optimization, sensor calibration, system observability.**

Manuscript received March 12, 2022; accepted April 25, 2022. This article was recommended for publication by Associate Editor M. Fallon and Editor F. Chaumette upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20484 and NSFC 62088101 Autonomous Intelligent Unmanned Systems. The work of Huang was partially supported by the University of Delaware College of Engineering. *(Jiajun Lv and Xingxing Zuo contributed equally this work.) (Corresponding author: Yong Liu.)*

Jiajun Lv, Xingxing Zuo, Kewei Hu, and Jinhong Xu are with the Institute of Cyber-System and Control, Zhejiang University, Hangzhou 310027, China (e-mail: lvjiajun314@zju.edu.com; xingxingzuo@zju.edu.cn; 1243246055@qq.com; xujinhong@zju.edu.cn).

Guoquan Huang is with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: ghuang@udel.edu).

Yong Liu is with the Institute of Cyber-System and Control, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute of Zhejiang University, Huzhou 313000, China (e-mail: yongliu@iipc.zju.edu.cn).

We also open source our codebase to benefit the research community: https://github.com/APRIL-ZJU/OA-LICalib.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2022.3174476.

Digital Object Identifier 10.1109/TRO.2022.3174476

I. Introduction

THE LiDAR-IMU (LI) system has prevailed in recent years [1]–[7], as an increasing number of robotic applications require accurate and robust LI-navigation solutions. Accurate LI extrinsic calibration is a prerequisite of LI navigation. Manually measuring the spatial extrinsic parameters (relative translation and rotation) between the LiDAR and IMU might be inaccurate or impractical, demanding an easy-to-use, automatic calibration approach. Besides the extrinsic calibration, calibrating the intrinsics of both sensors is equally important. IMU intrinsics [8], [9] modeling scale imperfection and axis misalignment between the gyroscope and accelerometer, are recommended to calibrate offline [10]. Similarly, LiDAR intrinsics [11], [12], such as each beam's horizontal and vertical angles and range offset, have been shown to be better calibrated for accurate registration and mapping rather than using factory default. Therefore, in this article, we perform both intrinsic and extrinsic calibration for LI systems.

Many calibration methods rely on infrastructure (e.g., fiducial targets or turntables) to aid calibration [13]–[15], which, however, increases the technical barrier for the widespread deployment of LI systems. LI sensors used in practical applications may not be perfectly time synchronized, necessitating the temporal calibration to compute the time offset between the two sensors. While discrete-time representation of states is most commonly used in calibration, its discretization introduces approximation when fusing asynchronous measurements and incurs a higher computational cost when processing high-rate data. Building upon our prior work [16], in this work, we develop a continuous-time LI calibration method that computes both intrinsic and extrinsic (spatial and temporal) LI sensing parameters without using dedicated infrastructure.

It is well understood that (infrastructure-free) sensor calibration can be heavily affected by the environment and motion because of observability (e.g., see [4], [10], [17]). In order to make the calibration tool easy-to-use for an end user, it is desirable to enable our LI calibration method to automatically address observability issues. To this end, we first perform observability-aware automatic selection of informative data for offline calibration; that is, we select only the most informative trajectory segments (rather than an uninformative long-session trajectory with weakly-excited motions). This is primarily due to the fact that if a calibration-data-collection trajectory is not fully excited, some calibration parameters may become unobservable and thus, cannot be computed or result in inconsistent results [10], [17]. However, how to collect informative data for calibration is not trivial for nonexpert users. To alleviate this technology barrier, the proposed observability-aware data selection will automatically select informative trajectory segments, which in

turn significantly reduces the computational cost as less data is needed for accurate calibration. On the other hand, if degenerate or weakly-excited motions inevitably occur—for example, in the autonomous-driving case, straight-line translation without rotation, and single-axis rotation—we also proposed to perform observability-aware state update during optimization so that no spurious information would influx into unobservable parameters. As a result, the proposed observability-aware calibration method, termed as *OA-LICalib*, is able to perform efficient and robust LI calibration with challenging motions and scenes.

In particular, the main contributions of this work include the following.

1) The proposed OA-LICalib is among the first to perform full LI calibration that is able to calibrate all sensing parameters including the spatial (i.e., rigid-body transformation between the sensors) and temporal (i.e., time offset between the sensors) extrinsics, and the intrinsics of LiDAR and IMU.
2) We propose the information-theoretic metrics to select informative trajectory segments for the LI calibration, instead of naively processing all available data regardless its observability, which significantly reduces the computational cost.
3) We develop the observability-aware state update during optimization by updating only the identifiable directions of state space under degenerate motions (e.g., planar motion and constant velocity).
4) Extensive experiments in simulations and real world are conducted to examine the feasibility and accuracy of the proposed OA-LICalib method, whose codebase is also open sourced to broadly benefit the robotics community.

The remainder of this article is organized as follows: After reviewing the literature in Section II, we provide the background materials of continuous-time batch optimization and present the sensor models in Section III. The problem formulation is given in Section IV, while the proposed observability-aware calibration method is presented in Section V. Simulated and real-world experiments are carried out in Sections VI and VII, respectively. Finally, Section VIII concludes this article.

II. Related Work

Sensor calibration has a rich literature. Instead of providing a comprehensive literature review, in this section, we only review the closely related work of LI calibration.

A. Spatial-Temporal LI Calibration

In order to calibrate the rigidly-connected LiDAR and IMU, Geiger *et al.* [18] proposed a motion-based calibration approach that performs the extrinsic calibration by hand-eye calibration [19]. However, their approach expects each sensor's trajectory to be estimated independently and accurately, which is difficult for the consumer-grade IMU. Gentil *et al.* [20] formulated the LI calibration as a factor graph optimization problem. Gaussian progress (GP) regression [21] was adopted to up-sample IMU measurements for pose interpolation at every time instants of capturing LiDAR points. Although the IMU data is continuously modeled based on GP, only the states at some specific time instants are optimized in the discrete-time factor graph, which may reduce accuracy. Recently, Mishra *et al.* [22] presented an extended kalman filter (EKF)-based LI extrinsic calibration method integrated into the visual-inertial navigation framework OpenVINS [23].

Online extrinsic LI calibration approaches are also available. In [4] and [24], LiDAR, IMU, and camera are tightly fused with a lightweight EKF, where the extrinsic parameters between the sensors are estimated online. Ye *et al.* [3] presented a tightly coupled LI odometry with online extrinsic parameters estimation, and adopted a linear motion model to eliminate motion distortion during the sweep. Qiu *et al.* [25] developed an IMU-centric temporal offset and extrinsic rotation calibration method for the LI system based on the motion correlation. Recently, Xu *et al.* [26] proposed FAST-LIO2, which is a tightly coupled LiDAR-inertial odometry with online extrinsic calibration in an iterated-EKF framework. Note that online calibration often assumes reasonable initial values in order to guarantee convergence and may have extra unobservable directions because of including calibration parameters into the state vector [4], [27].

Continuous-time batch optimization with temporal basis functions is also widely studied in calibration problems. Furgale *et al.* [28] detailed the derivation and realization for a full SLAM problem based on B-spline basis functions and evaluated the proposed framework within a camera-IMU calibration problem, which was further extended to support both temporal and spatial calibration [29]. Rehder *et al.* [30] adopted a similar framework for calibrating the extrinsics between a camera-IMU system and a single-beam LiDAR in two steps: the camera-IMU system is first calibrated with a chessboard, and then the single-beam LiDAR is calibrated with respect to the camera-IMU system.

B. Intrinsic Calibration

Li *et al.* [8] proposed an online state estimation framework using a multistate-constraint Kalman filter, which simultaneously estimates extrinsic parameters of a visual-inertial system, time offset, and intrinsics of IMU and camera. Rehder *et al.* [13] extended the previous work [29] to support the calibration of extrinsics and intrinsics of multiple IMUs in the visual-inertial system. Yang *et al.* [10] investigated the necessity of online IMU intrinsic calibration in the visual-inertial system and performed observability analysis for different IMU intrinsic models.

As the closest to our work, Liu *et al.* [14] proposed to calibrate an LI system's spatial extrinsics and intrinsics by discrete-time bundle adjustment optimization. A specially designed target composed of a cone and a cylinder is required to determine the intrinsics of 3-D LiDAR and formulate the constraints involved with the rigid transformation between LiDAR and IMU. A high-precision 3-axis-adjustable turntable platform with accurate attitude readings is also necessary to calibrate the IMU intrinsics. A close follow-up work [15] simplifies the method [14] by removing the dedicated target requirement and providing the concept to calibrate with the multiple geometric features naturally existing in human-made environments. Point/sphere, line/cylinder, and plane features are extracted and leveraged for providing valid constraints. However, reliable and automatic geometric features extraction from real-world scenes can be error-prone and susceptible; the extra facility of a high-precision turntable platform remains necessary for IMU intrinsic calibration.

C. Observability Awareness in Calibration

It is known that we can assess the information content of trajectory segments for camera and IMU calibration [9]. As such,

Maye *et al.* [31] proposed information-theoretic metrics to select informative data and discard redundant data for calibration. A truncated QR decomposition of the Fisher information matrix is employed to update estimates in only observable directions. Zhang *et al.* [32] analyzed the geometric structures in optimized problem for point cloud registration and determined well-conditioned directions to solve them partially. Schneider *et al.* [9] developed an online self-calibration method in discrete-time batch optimization for a visual-inertial system with the extended capability of calibrating IMU intrinsics. Multiple types of information metrics, including the trace, the determinant, and the maximal eigen value of the covariance matrix, are investigated for informative segment selection. The selected informative segments are fully excited with general motion, making the calibrated parameters observable and solvable. Jiao *et al.* [33] designed a sliding window-based multi-LiDAR odometry system with the capability of online extrinsic calibration between multiple LiDARs, and the singular values of Hessian matrix are leveraged to examine the convergence of rotational extrinsic calibration.

D. Extension of Our Previous Conference Publication [16]

While this article is evolved from our prior work on continuous-time extrinsic LI calibration of [16], there are significant contributions differentiating this work from [16] and others. To the best of authors' knowledge, the proposed OA-LICalib is among the first to fully calibrate all the sensing parameters of the LI sensors, including not only the spatial extrinsics as in [16], but the temporal extrinsics (time offset) as well as the intrinsic parameters of the LiDAR and IMU. As it is often required to fully excite the sensor platform during data collection in order for calibration results to converge, which, however, might not be possible for some commonly-seen applications, such as self-driving cars [4], in this article, we have proposed two observability-aware strategies to address this issue to make the proposed OA-LICalib method more easy-to-use for nonexpert end users. Specifically, 1) the information-theoretic-based data selection policy selects the most informative trajectory segments for calibration, which eases the effort in calibration data collection and improves the calibration efficiency. 2) The observability-aware state update scheme only updates the states (including calibration parameters) lying along the observable directions while preventing an influx of spurious information in the unobservable directions.

III. Problem Formulation

We first present our notations used throughout this article. We denote the 6-DoF rigid transformation by ${}^B_A\mathbf{T} \in \mathrm{SE}(3) \subset R^{4\times4}$, which transforms the point ${}^Ap \in R^3$ in frame $\{A\}$ into frame $\{B\}$. ${}^B_A\mathbf{T} = \begin{bmatrix} {}^B_A\mathbf{R} & {}^Bp_A \\ \mathbf{0} & 1 \end{bmatrix}$ consists of rotational part ${}^B_A\mathbf{R} \in \mathrm{SO}(3)$ and translational part ${}^Bp_A \in R^3$. For simplicity, we omit the homogeneous conversion in the rigid transformation by ${}^Bp = {}^B_A\mathbf{T}{}^Ap$. ${}^B_A\bar{q}$ is the quaternion of rotation corresponding to the rotation matrix ${}^B_A\mathbf{R}$. In our formulation, IMU reference frame $\{I\}$ is assumed to be rigidly connected to the LiDAR frame $\{L\}$. The map frame $\{M\}$ of LiDAR point cloud is determined by the first LiDAR frame $\{L_0\}$ when the calibration starts. Similarly, the global frame $\{G\}$ is determined as the first IMU frame $\{I_0\}$. The spatial extrinsics, including the relative rotation and translation from the LiDAR frame to the IMU frame, are denoted by ${}^I_L\bar{q}$ and ${}^I\mathbf{p}_L$, respectively. The temporal extrinsic parameter modeling the time offset between LiDAR and IMU is t_c. The LiDAR scan labeled with timestamp τ_L corresponds to IMU time instant $t_I = \tau_L + t_c$. Fig. 1 visualizes the different frames of reference used in this article.

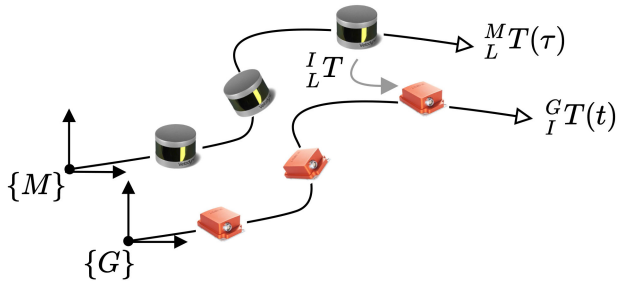


Fig. 1. Illustration of different frames of reference used in this article. The trajectory of LiDAR can be represented with the IMU trajectory, the 6-DoF rigid transformation ${}^I_L\mathbf{T}$, and the time offset t_c between LiDAR and IMU, by ${}^M_L\mathbf{T}(\tau) = ({}^G_I\mathbf{T}(\tau_0 + t_c){}^I_L\mathbf{T})^\top {}^G_I\mathbf{T}(\tau + t_c)\,{}^I_L\mathbf{T}$.

The extrinsic and intrinsic LI calibration considered in this article aims to estimate the following states:

$$\mathcal{X} = \{X_p, X_q, X_{Is}, X_I, X_L, {}^I_L\bar{q}, {}^I\mathbf{p}_L, t_c\} \tag{1}$$

which includes the control points of B-splines $\{X_p, X_q\}$ to represent the continuous-time trajectory, the IMU navigation states X_{Is}, the IMU intrinsics X_I, the LiDAR intrinsics X_L, and the LI spatiotemporal extrinsic parameters $\{{}^I_L\bar{q}, {}^I\mathbf{p}_L, t_c\}$. Based on the LI sensor data collected during the calibration process, we formulate the following nonlinear least-squares (NLS) problem:

$$\hat{\mathcal{X}} = \underset{\mathcal{X}}{\operatorname{argmin}}\, r, \quad r = r_I + r_L \tag{2}$$

where r_I is the IMU measurement residual cost function, which will be derived in detail in (12) and r_L is the LiDAR measurement residual cost function as derived in (22).

A. Continuous-Time Trajectory Representation

We employ B-spline to parameterize trajectory as it provides closed-form analytic derivatives [34], enabling the effortless fusion of high-frequency measurements for state estimation. B-spline also has the good property of being locally controllable, which means the update of a single control point only impacts certain consecutive segments of the spline [28]. This trait yields a sparse system with a limited number of control points. To parameterize the continuous-time 6-DoF trajectory, we leverage the split representation [35], which represents the 3-D translation and the 3-D rotation with uniform B-splines separately. Both the translation and the rotation splines are parameterized with cumulative form. To be specific, the translation $\mathbf{p}(t)$ of d degrees over time $t \in [t_i, t_{i+1})$ is controlled by the temporally uniformly distributed translational control points $\mathbf{p}_i, \mathbf{p}_{i+1}, \ldots, \mathbf{p}_{i+d}$, and the matrix format could be expressed

$$\mathbf{p}(t) = \mathbf{p}_i + \sum_{j=1}^{d} \mathbf{u}^\top \tilde{\mathbf{M}}^{(d+1)}_{(j)} \left(\mathbf{p}_{i+j} - \mathbf{p}_{i+j-1}\right) \tag{3}$$

where $\mathbf{u}^\top = \begin{bmatrix} 1 & u & \ldots & u^d \end{bmatrix}$ and $u = (t - t_i)/(t_{i+1} - t_i)$. $\tilde{\mathbf{M}}^{(4)}_{(j)}$ is the jth column of the cumulative spline matrix $\tilde{\mathbf{M}}^{(4)}$,

which only depends on the corresponding degree of uniform B-spline. In this article, cubic ($d = 3$) B-spline is employed, thus, the corresponding cumulative spline matrix is

$$\tilde{\mathbf{M}}^{(4)} = \frac{1}{6}\begin{bmatrix} 6 & 5 & 1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix}. \tag{4}$$

Besides, we adopt the cumulative representation of B-spline to parameterize the rotation on SO(3) like [35]–[37], and quaternions are served as the rotational control points of B-splines [38]

$$\bar{q}(t) = \bar{q}_i \otimes \prod_{j=1}^{d} \exp\left(\mathbf{u}^\top \tilde{\mathbf{M}}^{(4)}_{(j)} \log(\bar{q}_{i+j-1}^{-1} \otimes \bar{q}_{i+j})\right) \tag{5}$$

where $\otimes$ denotes quaternion multiplication. $\bar{q}_i$ denotes the rotational control point. $\exp(\cdot)$ is the operation that mapping Lie algebra elements to S^3, and $\log(\cdot)$ is its inverse operation [38].

In this calibration system, the continuous 6-DOF poses of IMU ${}^G_I\bar{q}(t), {}^G\mathbf{p}_I(t)$ are represented in the global frame $\{G\}$ by splines in the format of (3) and (5) with parameters of control points X_p and X_q for 3D translation and rotation, respectively. The derivatives of the splines with respect to time can be easily computed [37], which lead to linear accelerations ${}^I\mathbf{a}(t)$ and angular velocities ${}^I\omega(t)$ in the local IMU reference frame

$${}^I\mathbf{a}(t) = {}^G_I\mathbf{R}^\top(t)\left({}^G\ddot{\mathbf{p}}_I(t) - {}^G\mathbf{g}\right) \tag{6}$$

$${}^I\omega(t) = {}^G_I\mathbf{R}^\top(t){}^G_I\dot{\mathbf{R}}(t) \tag{7}$$

where ${}^G\mathbf{g} \in \mathbb{R}^3$ denotes the gravity vector in global frame. ${}^G\mathbf{g}$ has only two degrees of freedom since its norm is assumed to be a constant $\|{}^G\mathbf{g}\| \simeq 9.8$ in the system.

B. IMU Model

IMU comprises a 3-axis gyroscope and a 3-axis accelerometer. Inspired by [8], [10], [13], the IMU measurements are modeled as

$$\begin{aligned} {}^\omega\omega &= \mathbf{S}_\omega\mathbf{M}_\omega\,{}^\omega_I\mathbf{R}\,{}^I\omega(t) + \mathbf{A}_\omega\,{}^I\mathbf{a}(t) + \mathbf{b}_\omega + \mathbf{n}_\omega \\ &= h_{I_\omega}(t, X_q, {}^G\mathbf{g}, \mathbf{S}_\omega, \mathbf{M}_\omega, {}^\omega_I\mathbf{R}, \mathbf{A}_\omega, \mathbf{b}_\omega) \end{aligned} \tag{8}$$

$$\begin{aligned} a_a &= \mathbf{S}_a\mathbf{M}_a\,{}^a_I\mathbf{R}\,{}^I\mathbf{a}(t) + \mathbf{b}_a + \mathbf{n}_a \\ &= h_{I_a}(t, X_p, X_q, {}^G\mathbf{g}, \mathbf{S}_a, \mathbf{M}_a, {}^a_I\mathbf{R}, \mathbf{b}_a) \end{aligned} \tag{9}$$

where X_p and X_q are the translational and rotational control points of B-splines, respectively. $\mathbf{b}_\omega$ and $\mathbf{b}_a$ are the biases of gyroscope and accelerometer, which are assumed to be under Gaussian random walk. $\mathbf{n}_\omega$ and $\mathbf{n}_a$ are the zero-mean white Gaussian noise with covariance matrix Σ_ω and Σ_a, respectively, which can be obtained from IMU noise model without approximation or propagation. $\mathbf{S}_{\omega/a}$ is a diagonal matrix modeling the scale imperfection as

$$\mathbf{S}_\omega = \begin{bmatrix} S_{w1} & 0 & 0 \\ 0 & S_{w2} & 0 \\ 0 & 0 & S_{w3} \end{bmatrix}, \mathbf{S}_a = \begin{bmatrix} S_{a1} & 0 & 0 \\ 0 & S_{a2} & 0 \\ 0 & 0 & S_{a3} \end{bmatrix}. \tag{10}$$

$\mathbf{M}_{\omega/a}$ accounting for axis misalignment is an upper-triangular matrix with diagonal elements as identity

$$\mathbf{M}_\omega = \begin{bmatrix} 1 & M_{w1} & M_{w2} \\ 0 & 1 & M_{w3} \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{M}_a = \begin{bmatrix} 1 & M_{a1} & M_{a2} \\ 0 & 1 & M_{a3} \\ 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

A_ω is a full 3×3 matrix accounting for the acceleration dependence (g-sensitivity) of the measurements.

Since the gyroscope $\{\omega\}$ and accelerometer $\{a\}$ inside the IMU are individual sensors, there is a misalignment between them and the base IMU frame $\{I\}$. ${}^{\omega/a}_I\mathbf{R}$ compensates for the rotational misalignment, and the translational misalignment is omitted for calibration since it is close to zero for single-chip MEMS sensors [9], [10], [39]. We can only calibrate ${}^\omega_I\mathbf{R}$ or ${}^a_I\mathbf{R}$ since the base IMU frame $\{I\}$ is aligned with either $\{\omega\}$ or $\{a\}$, and calibrating both makes the system unobservable [10]. In this article, we choose to calibrate ${}^\omega_I\mathbf{R}$ only. In addition, we ignore the calibration of gravity sensitivity in analogy to [10].

Usually, IMU measurements are given at discrete-time instants. The measurements at timestamp t_k are denoted as ${}^{\omega_k}\omega_m, {}^{a_k}a_m$. Since the trajectory is formulated in continuous time, we can readily get ${}^I\mathbf{a}(t_k), {}^I\omega(t_k)$ from (6) and (7) by computing differentials of the trajectory. The IMU induces cost functions, r_I, related to estimated variables including the control points of the trajectory $\{X_p, X_q\}$, the IMU navigation states $X_{\text{Is}} = \{{}^G\mathbf{g}, \mathbf{b}_\omega, \mathbf{b}_a\}$, and the IMU intrinsics $X_I = \{\mathbf{S}_\omega, \mathbf{M}_\omega, {}^\omega_I\mathbf{R}, \mathbf{S}_a, \mathbf{M}_a\}$ are formulated as

$$r_I = r_\omega + r_a, \quad \text{where} \tag{12}$$

$$r_\omega = \sum_k \frac{1}{2}\|{}^{\omega_k}\omega_m - h_{I_\omega}(t_k, X_q, X_I, X_{Is})\|^2_{\Sigma_\omega} \tag{13}$$

$$r_a = \frac{1}{2}\|{}^{a_k}a_m - h_{I_a}(t_k, X_p, X_q, X_I, X_{Is})\|^2_{\Sigma_a} \tag{14}$$

$\|\mathbf{e}\|^2_\Sigma = \mathbf{e}^\top\Sigma^{-1}\mathbf{e}$ denotes the squared energy norm weighted by the inverse of the covariance matrix Σ, and r_ω, r_a are residuals of angular velocity and linear acceleration according to (8) and (9), respectively.

C. 3-D LiDAR Model

A mechanical 3-D LiDAR with multiple laser beams [40] measures the ranges given by individual laser heads pointing to different elevation angles. A LiDAR scan is captured when the rigidly connected laser heads rotate around a mechanical central axis. A laser head i with elevation angle ϕ_i at LiDAR time instant τ_k (with azimuth angle θ_{ik}) gets a range measurement ρ_{ik}, then the LiDAR point measurement transferred from the spherical coordinate system can be denoted by

$${}^{L_k}P_{ik} = \begin{bmatrix} {}^{L_k}x_{ik} \\ {}^{L_k}y_{ik} \\ {}^{L_k}z_{ik} \end{bmatrix} = \begin{bmatrix} \rho_{ik}\cos\phi_i\cos\theta_{ik} \\ \rho_{ik}\cos\phi_i\sin\theta_{ik} \\ \rho_{ik}\sin\phi_i \end{bmatrix}. \tag{15}$$

However, the abovementioned ideal model does not hold in practical manufacture since spatial offsets exist between laser heads, and nonnegligible errors exist in both the range measurement and the azimuth angle θ_{ik}. As shown in Fig. 2, the intrinsics of an individual laser head i consist of the following.

1) Elevation angle correction factor $\delta\phi_i$ and azimuth angle correction factor $\delta\theta_i$.

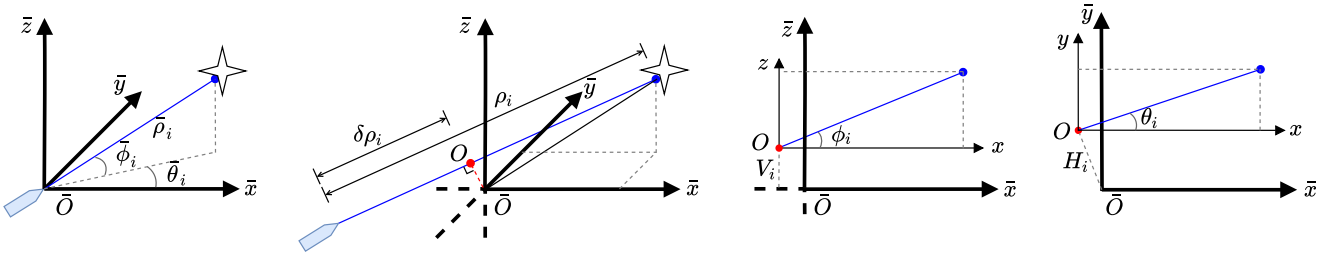


Fig. 2. Intrinsics of an individual laser comprising a multibeam 3-D LiDAR. The left figure depicts the ideal model, while the three figures on the right illustrate the intrinsics arising from correction factors and offsets.

2) Vertical spatial offset V_i and horizontal spatial offset H_i.
3) Scale factor of range measurement s_i and range measurement offset $\delta\rho_i$.

Given the intrinsics, the zero-mean Gaussian measurement noise $n_{\rho,ik}$ over the range measurement, and defining the following factors:

$$\begin{cases} \bar{\phi}_i = \phi_i + \delta\phi_i \\ \bar{\theta}_{ik} = \theta_{ik} + \delta\theta_i \\ \bar{\rho}_{ik} = s_i\rho_{ik} + \delta\rho_i + n_{\rho,ik} \end{cases}. \tag{16}$$

The position of LiDAR point measurement is given by

$${}^{L_k}P_{ik} = \begin{bmatrix} \bar{\rho}_{ik}\cos\bar{\phi}_i\cos\bar{\theta}_{ik} + H_i\sin\bar{\theta}_{ik} \\ \bar{\rho}_{ik}\cos\bar{\phi}_i\sin\bar{\theta}_{ik} + H_i\cos\bar{\theta}_{ik} \\ \bar{\rho}_{ik}\sin\bar{\phi}_i + V_i \end{bmatrix}. \tag{17}$$

The intrinsics of a 3-D LiDAR with l beams are denoted by

$$X_L = \{\delta\phi_i, \delta\theta_i, V_i, H_i, s_i, \delta\rho_i\}_{|i=0,1,\ldots,l-1}. \tag{18}$$

For a LiDAR point measurement ${}^{L_k}P_{ik}$ with an associated 3-D plane with closest point parameterization [41], ${}^{M}\pi_j = {}^{M}d_{\pi,j}\,{}^{M}n_{\pi,j}$, where ${}^{M}d_{\pi,j}$ and ${}^{M}n_{\pi,j}$ denote the distance of the plane to origin and unit normal vector expressed in map frame, respectively. The point to plane distance is given by

$$\begin{aligned} {}^{M}P_{ik} &= {}^{M}_{L_k}\mathbf{R}(\tau_k)\,{}^{L_k}P_{ik} + {}^{M}\mathbf{p}_{L_k}(\tau_k) \qquad (19) \\ z_{ijk} &= {}^{M}n_{\pi,j}^{\top}\,{}^{M}P_{ik} + {}^{M}d_{\pi,j} \\ &= h_L(\tau_k, X_p, X_q, X_L, {}^{I}_{L}\bar{q}, {}^{I}\mathbf{p}_L, t_c). \qquad (20) \end{aligned}$$

The distance measurement mentioned above is related to variables including the control points of trajectory $\{X_p, X_q\}$, LiDAR intrinsics X_L, and spatial-temporal parameters $\{{}^{I}_{L}\bar{q}, {}^{I}\mathbf{p}_L, t_c\}$. The 3-D plane is extracted from LiDAR point cloud map, detailed in Section IV-B, and assumed free of noise when computing z_{ijk}. Thus, $\Sigma_{z_{ijk}}$, the covariance of the distance measurement z_{ijk}, can be easily propagated from the Gaussian noise $n_{\rho,ik}$ by first-order approximation. The relative pose $\{{}^{M}_{L_k}\mathbf{R}(\tau_k), {}^{M}\mathbf{p}_{L_k}(\tau_k)\}$ in (20) is involved with the following equations from parameterized IMU continuous-time trajectory:

$${}^{M}_{L_k}\mathbf{T}(\tau_k) = \left({}^{G}_{I}\mathbf{T}(\tau_0 + t_c){}^{I}_{L}\mathbf{T}\right)^{\top}\,{}^{G}_{I}\mathbf{T}(\tau_k + t_c)\,{}^{I}_{L}\mathbf{T}. \tag{21}$$

It is easy to find that the time offset t_c between LiDAR and IMU is also involved in the process of fetching pose ${}^{M}_{L_k}\mathbf{T}(\tau_k)$ from continuous-time trajectory parameterized in the IMU time axis. In practice, the LiDAR point measurement at timestamp τ_k is denoted as $z_{m,ijk}$. By minimizing the point to plane distance, we have the following cost functions from LiDAR points associated with 3-D planes:

$$\begin{aligned} r_L = \frac{1}{2}\sum_i\sum_j\sum_k \| & z_{m,ijk} \\ & - h_L(\tau_k, X_p, X_q, X_L, {}^{I}_{L}\bar{q}, {}^{I}\mathbf{p}_L, t_c)\|^2_{\Sigma_{z_{m,ijk}}} \end{aligned} \tag{22}$$

where i, j, k denote the indexes of LiDAR beams, 3-D planes and points in a scanned ring by a laser beam, respectively.

IV. Continuous-Time Full LI Calibration

In this section, we present in detail the proposed continuous-time full LI calibration method. Before diving into details, we first provide an overview of the overall architecture, as shown in Fig. 3. Specifically, we first initialize the control points of rotation B-splines with the IMU angular velocities, and the extrinsic rotation ${}^{I}_{L}\bar{q}$ by aligning the IMU rotations with the LiDAR odometry (rotations) computed with NDT registration [42]. By reducing the rotational distortion effect in LiDAR scans, we recompute the LiDAR odometry to improve its accuracy. We then initialize the LiDAR surfels by fusing LiDAR scans with the computed LiDAR odometry and find the point-to-surfel correspondences. With the LI measurements, we perform batch optimization to estimate the considered states including B-splines control points and spatial-temporal extrinsics, whose optimal estimates are used to refine the surfels map and point-to-plane data association.

A. State Initialization

Initial guesses of the calibrated parameters are needed for the iterative NLSs optimization. For initialization, we set IMU navigation states as $\hat{X}_{\mathrm{Is}_0} = \{\|{}^{G}\mathbf{g}\|e_3, \mathbf{0}_3, \mathbf{0}_3\}$ and IMU intrinsics as $\hat{X}_{I_0} = \{I_3, I_3, I_3, I_3, I_3\}$, where $e_3 = [0, 0, 1]^{\top}$, I_3 and $\mathbf{0}_3$ represent the 3×3 identity matrix and zero matrix, respectively. As for the initial values of LiDAR intrinsics, $\hat{X}_{L_0}$ is initialized as $\{0, 0, 0, 0, 1, 0\}_i|_{i=1,\ldots,l}$. The time offset t_c between LiDAR and IMU is initialized as zero, which means the timestamp t_k of IMU is equal to the timestamp τ_k of LiDAR at the beginning.

The initialization of control points and extrinsic parameters are described in detail as follows. Prior to detailing the initialization process, we clarify the inputs for initialization including discrete poses at about 10 Hz from LiDAR odometry, angular velocities, and linear accelerates at typically 400 Hz from IMU. We adopt NDT registration to implement simple scan-to-map LiDAR odometry, sufficient for sequences with a few seconds,

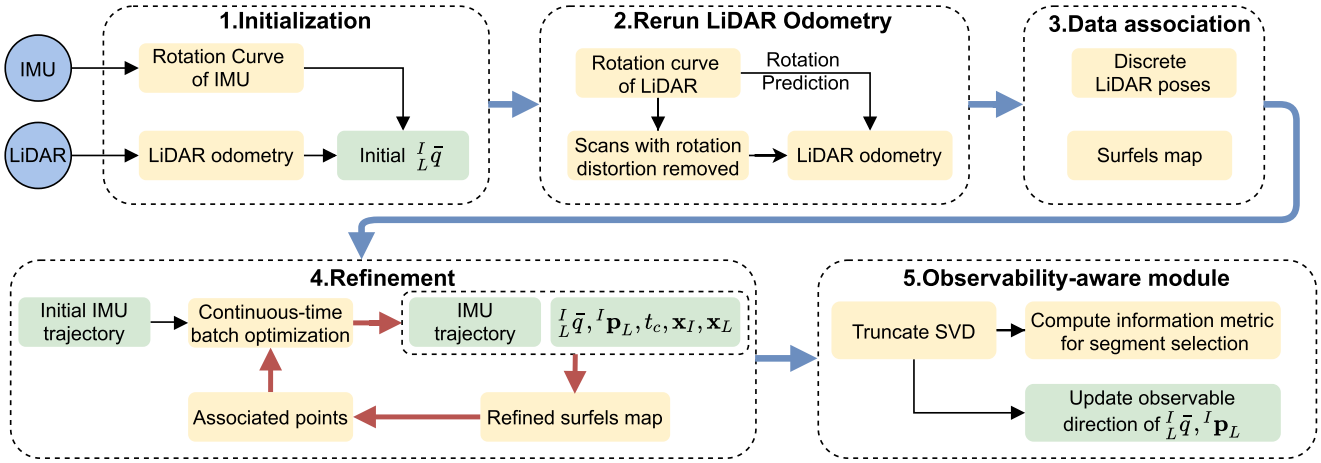


Fig. 3. Pipeline of proposed LI calibration method, which allows leveraging all the raw measurements from IMU and LiDAR sensor in a continuous-time batch optimization framework. Details are provided in Sections IV and V.

and build a global map consists of keyframed LiDAR scans determined according to the pose changes.

1) Initialize Control Points of Continuous-Time Trajectory: Before detailing the concrete initialization method, we emphasize that the spline trajectory is employed to represent the IMU trajectory rather than LiDAR's. If we model the continuous-time trajectory in LiDAR sensor, the predicted linear acceleration and angular velocity in IMU frame need to be transformed from LiDAR frame as

$$\begin{aligned} {}^{I}a(t) &= {}^{I}_{L}\mathbf{R}({}^{M}_{L}\mathbf{R}(t)^{\top}\left({}^{M}a(t) - {}^{M}g\right) \\ &\quad + \lfloor {}^{L}\dot{\boldsymbol{\omega}}(t)\rfloor {}^{I}p_L + \lfloor {}^{L}\boldsymbol{\omega}(t)\rfloor^2 {}^{I}p_L) \end{aligned} \tag{23}$$

$${}^{I}\boldsymbol{\omega}(t) = {}^{I}_{L}\mathbf{R}{}^{L}\boldsymbol{\omega}(t) \tag{24}$$

where $\lfloor\cdot\rfloor$ represents the skew-symmetric matrix of a 3-D vector. Note that angular accelerations ${}^{L}\dot{\boldsymbol{\omega}}(t)$ are required to be computed, which makes the system complicated. Therefore, we choose to parameterize the trajectory in IMU frame instead of LiDAR frame.

Given the raw angular velocity measurements from the IMU sensor, a series of rotational control points X_q of the rotational trajectory can be initialized by solving the following least-square problem:

$$\begin{aligned} \hat{X}_q &= \underset{X_q}{\operatorname{argmin}}\ r_\omega \\ &= \underset{X_q}{\operatorname{argmin}} \sum_k \frac{1}{2}\|{}^{\omega_k}\omega_m - h_{I_\omega}(t_k, X_q, X_I, X_{Is})\|^2_{\Sigma_\omega} \end{aligned} \tag{25}$$

where r_ω is defined in (13) and here we ignore the intrinsics of IMU and the angular velocity bias of IMU. It is important to note that ${}^{G}_{I}\bar{q}(t_0)$ is fixed to the identity quaternion during the optimization. In (25), we try to initialize the rotational trajectory by the raw IMU measurements rather than the integrated IMU poses, since the latter are inaccurate and always affected by drifting IMU biases and noises.

For the initialization of translational control points X_p, we first utilize the LiDAR odometry to get descrete-time LiDAR poses. Then, we formulate and solve the following linear problem:

$$\begin{aligned} \hat{X}_p = \underset{X_p}{\operatorname{argmin}} \sum_k \|{}^{G}_{M}\mathbf{T}\,{}^{M}\hat{\boldsymbol{p}}_{L_k} - {}^{G}_{I}\hat{\mathbf{R}}(t_k, \hat{X}_q)\,{}^{I}\boldsymbol{p}_L \\ - {}^{G}\boldsymbol{p}_I(t_k, X_p)\| \end{aligned} \tag{26}$$

where ${}^{M}\hat{\boldsymbol{p}}_{L_k}$ is the estimated position of scan at timestamp τ_k from the LiDAR odometry; ${}^{G}_{I}\hat{\mathbf{R}}(t_k, \hat{X}_q)$ controlled by $\hat{X}_q$ has been initialized in (25) and ${}^{G}\boldsymbol{p}_I(t_k, X_p)$ controlled by X_p is the translational trajectory to be initialized. Considering map frame and global frame are determined as the first measurements from LiDAR and IMU, respectively, ${}^{G}_{M}\mathbf{T}$ is equivalent to ${}^{I_0}_{L_0}\mathbf{T}$, that is ${}^{I}_{L}\mathbf{T}$. The initial values of ${}^{I}_{L}\mathbf{T}$ computed in Section IV-A2 are used here. This initialization method of control points is relatively coarse, but it can effectively improve the convergence speed of the optimization problem, compared to leave X_p uninitialized as zeros.

2) Extrinsic Initialization: Inspired by Yang and Shen [43], we initialize the extrinsic rotation by aligning two rotation sequences from LiDAR and IMU. With discrete LiDAR poses from LiDAR odometry, it is easy to get the relative rotation between two consecutive LiDAR scans, ${}^{L_k}_{L_{k+1}}\bar{q}$. Besides, the relative rotation between time interval $[t_k, t_{k+1}]$ in the IMU frame can also be obtained from the initialized rotational trajectory as ${}^{I_k}_{I_{k+1}}\bar{q} = {}^{G}_{I}\bar{q}^{-1}(t_k) \otimes {}^{G}_{I}\bar{q}(t_{k+1})$. The relative rotations at any k from two sensors should satisfy the following equation:

$${}^{I_k}_{I_{k+1}}\bar{q} \otimes {}^{I}_{L}\bar{q} = {}^{I}_{L}\bar{q} \otimes {}^{L_k}_{L_{k+1}}\bar{q}. \tag{27}$$

The abovementioned equation can be transferred into another equivalent representation [44]:

$$\begin{aligned} \mathcal{L}({}^{I_k}_{I_{k+1}}\bar{q})\,{}^{I}_{L}\bar{q} = \mathcal{R}({}^{L_k}_{L_{k+1}}\bar{q})\,{}^{I}_{L}\bar{q} \\ \Rightarrow \left(\mathcal{L}({}^{I_k}_{I_{k+1}}\bar{q}) - \mathcal{R}({}^{L_k}_{L_{k+1}}\bar{q})\right) {}^{I}_{L}\bar{q} = \mathbf{0}. \end{aligned} \tag{28}$$

For a unit quaternion $\bar{q} = \begin{bmatrix} q_v & q_w \end{bmatrix}^{\top}$ with imaginary part q_v and real part q_w, $\mathcal{L}(\bar{q})$ and $\mathcal{R}(\bar{q})$ are multiplication matrices defined

as follows:

$$\mathcal{L}(\bar{q}) = \begin{bmatrix} q_w \mathbf{I}_3 - \lfloor \mathbf{q}_v \rfloor & \mathbf{q}_v \\ -\mathbf{q}_v^T & q_w \end{bmatrix}$$

$$\mathcal{R}(\bar{q}) = \begin{bmatrix} q_w \mathbf{I}_3 + \lfloor \mathbf{q}_v \rfloor & \mathbf{q}_v \\ -\mathbf{q}_v^T & q_w \end{bmatrix}.$$

Equation (28) is in a form of ${}^{k}_{k+1}A^{I}_{L}\bar{q} = \mathbf{0}$. By stacking a series of relative rotation measurements ${}^{k}_{k+1}A$ row by row into a big matrix $\mathbf{A}$, we have $\mathbf{A}^{I}_{L}\bar{q} = \mathbf{0}$, then ${}^{I}_{L}\hat{\bar{q}}$ can be computed as the right unit singular vector corresponding to the smallest singular value of $\mathbf{A}$.

For the initialization of extrinsic translation ${}^{I}\boldsymbol{p}_{L}$, the initial values of ${}^{I}\boldsymbol{p}_{L}$ are simply set to zeros, and the proposed method is able to converge to reasonable values in practical experiments when enough motion excitation exists.

B. Data Association of LiDAR Measurements

With the estimated rotational extrinsics ${}^{I}_{L}\hat{\bar{q}}$, we can transfer the rotational continuous-time curve in IMU frame into LiDAR frame and remove the rotational motion distortion in LiDAR scans. Subsequently, we perform the NDT-based LiDAR odometry again with the initial rotational guess retrieved from the rotational continuous-time curve transformed. Since LiDAR scans are partially undistorted and better initial guesses are provided, NDT-based LiDAR odometry can generate more accurate estimated poses. With the odometry results, a LiDAR point cloud map can be constructed by assembling multiple LiDAR scans into an identical frame. We further uniformly discretize the LiDAR point cloud map into 3-D cells and compute the point distribution in every 3-D cell to identify surfels based on a plane-likeness coefficient [45]:

$$\mathcal{P} = 2\frac{\lambda_1 - \lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}. \tag{29}$$

In the abovementioned equation, $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are the point distribution eigen values in a 3-D cell. For the cell with planar point distribution, $\mathcal{P}$ will be close to 1. The cell with plane-likeness coefficient $\mathcal{P}$ larger than a threshold is determined as a surfel cell, and the points inside this cell are used to fitting a plane. The estimated plane is parameterized by the closest point as mentioned in Section III-C. Some LiDAR points are associated with surfels if the point-to-plane distance is less than a threshold, whereas the others are discarded. As mentioned in Section III-C, the plane parameters are assumed to be perfect and keep constant when we perform least-squares optimization, so are the surfels map and the data associations. This is in a fashion of Expectation–maximization algorithm. After the optimization, the surfels map, the data associations, and the plane parameters will be updated with updated estimated states, which will be detailed in Section IV-C. In addition, two practical tricks are engaged to repress the effect of outlier data associations on optimization: first, only the point and plane correspondence with a distance below 5 centimeters will be taken into the continuous-time batch optimization; second, we rely on the M-estimation [46] idea and apply the Huber robust kernel to deal with outliers.

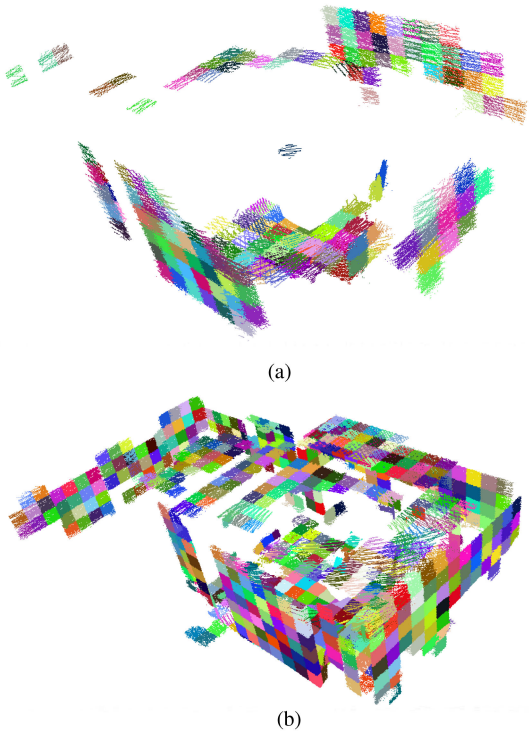


Fig. 4. Typical surfels map in an indoor scenario. The initial surfels map suffers from motion distortion with the initialized LiDAR poses. With the estimated states after just one iteration of batch optimization, the quality of refined surfels map becomes significantly better. Points in different surfels are colorized differently. (a) Surfels map in the first iteration. (b) Surfels map in the second iteration.

C. Refinement

After the continuous-time batch optimization, the estimated states including the extrinsics become more accurate. Thus, we leverage the current best estimates to remove the motion distortion in the raw LiDAR scans, rebuild the LiDAR surfels map, and update the point-to-surfel data associations. Note that results from NDT-based LiDAR odometry are only utilized at the very beginning for initializing the LiDAR poses and LiDAR map. The refinement step with iterative batch-optimization will generate more accurate estimations and reliable data association. The typical LiDAR surfels maps in the first and second iterations of the batch optimization are shown in Fig. 4. The quality of the map can be improved in a significant margin after one iteration in the batch optimization.

As for the intrinsic calibration, a 16-beam LiDAR contains 96 parameters (6 parameters per beam), and some parameters of the first laser are fixed as a reference during the calibration process. Only the distance offset and distance scale are estimated for the first laser. In addition, there are 15 parameters of IMU intrinsic parameters. Considering the large number of intrinsic parameters to be estimated, to prevent the system from falling into local optimum and reduce the complexity of the problem, we optimize the spatial-temporal parameters only and fix the intrinsics at the initial values first, and allow optimizing the intrinsic parameters a few iterations later when the estimates of the trajectory and extrinsic parameters converge. Furthermore, we correct the raw measurements of IMU and LiDAR based on the estimated intrinsic parameters after they have gradually converged, instead of using the poor intrinsic estimates to correct

the measurements at the beginning, which may hurt the calibration results. Throughout the experiments, we start to calibrate the intrinsic parameters after 2 iterations of batch optimization and correct raw measurements based on the estimated intrinsic parameters after 11 iterations.

V. Observability-Aware LI Calibration

In this section, we describe in detail the observability-aware strategies used in the proposed LI calibration in order to improve its robustness to degenerate motions and thus make it more easy-to-use for practitioners.

A. Information-Theoretic Data Selection

Note that some data segments collected without sufficient motion excitation or scene constraints may degrade the calibration [47], and blindly processing all the available data may result in unnecessarily high computational cost. As such, the proposed method seeks to automatically select the most informative data segments for calibration. In particular, to facilitate the ensuing derivations, assuming the white Gaussian noise and independent measurements, the NLS problem (2) is equivalent to the maximum likelihood estimation (MLE) problem and can be reformulated as follows (see [48], [49]):

$$\hat{X} = \underset{X}{\operatorname{argmin}} \sum_i \frac{1}{2} \|e_i(X)\|^2_{\Sigma_i} \tag{30}$$

where e_i denotes the residual induced from the ith measurement with covariance Σ_i. An iterative method such as Gaussian–Newton and Levenberg–Marquard is often used to solve this problem. That is, at each iteration, by linearizing the above-mentioned nonlinear system at the current estimate $\hat{X}$, $e_i(X) = e_i(\hat{X}) + J_i \delta X$, with the Jacobian matrix J_i, we have the following linear least-squares problem with respect to the update increment (or error state estimate) $\delta\hat{X}$:

$$\delta\hat{X} = \underset{\delta X}{\operatorname{argmin}} \sum_i \frac{1}{2} \|e_i(\hat{X}) + J_i \delta X\|^2_{\Sigma_i}. \tag{31}$$

The optimal solution is given by the following normal equation:

$$\underbrace{\left(\sum_i J_i^\top \Sigma_i^{-1} J_i\right)}_{\mathbf{A}} \delta X = \underbrace{-\sum_i J_i^\top \Sigma_i^{-1} e_i(\hat{X})}_{\mathbf{b}}. \tag{32}$$

Once $\delta\hat{X}$ is computed, we update the estimate as $\hat{X} \leftarrow \hat{X} \boxplus \delta X$, where $\boxplus$ denotes the addition operation on manifold.

Note that $A_{k\times k}$ in (32) is the Fisher information matrix of the MLE (or NLS) problem (30) and encapsulates all the information contained in the measurements. It is also important to note that A might be rank deficient, for example, due to the lack of motion excitation and/or environmental constraints (see [4]). It, thus, is critical to collect informative data ensuring full-rank A (or observability [50]). To this end, we first partition the entire collected data into multiple segments with constant-time length. We then perform singular value decomposition (SVD) of A for each segment

$$A = USU^\top \tag{33}$$

where $S = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$ is the diagonal matrix of singular values in decreasing order, and $U = \begin{bmatrix} u_1, u_2, \ldots, u_k \end{bmatrix}$ is an orthogonal matrix. As in [50], we employ the information metric (or observability index) of an individual segment based on the minimum singular value—that is, a larger one implies a more informative segment. With that, for a specific sequence of collected data, we first evaluate the information metric of each segment and then select the most informative ones for LI calibration.

B. Observability-Aware Update

While we have tried to collect the most informative data for calibration, it still might be the case that the selected data cannot guarantee observability, and thus, some states cannot be estimated. For example, when a ground vehicle moves on planar surfaces in an urban environment, the lack of pitch and roll rotation and translation on the vertical direction often leads to part of states unobservable. As shown in [4], one-axis rotation is a so-called degenerate motion causing the translation of extrinsics ${}^I P_L$ partially unobservable. If this unobservability issue is not properly addressed, the calibration performance would be largely degraded.

To make the proposed LI calibration robust to (weak) unobservability, we explicitly enforce observability constraints during each iteration; that is, only updating the states lying on the observable directions. To this end, we employ the truncated SVD (TSVD) [51], [52] to perform low-rank approximation of the information matrix, i.e., selecting the singular values $(\sigma_1, \sigma_2, \ldots, \sigma_l)$ over an information threshold ϵ that is a design choice. With this observability-assurance information matrix and SVD-based matrix pseudoinverse, we have the following solution [see (32)]:

$$\delta\hat{X} = \sum_{i=1}^{l} \frac{u_i^\top \mathbf{b} u_i}{\sigma_i}. \tag{34}$$

As a result, as compared to solving for $\delta\hat{X}$ by using all the singular values of the full information matrix—which may erroneously introduce observable directions due to numerical issues—we here explicitly avoid updating the unobservable (or weakly observable) parameters, thus, conservatively ensuring observability.

VI. Simulation Results

To validate the performance of the proposed continuous-time batch optimization-based LI calibration method, we conduct a series of simulations with ground truth of all estimated parameters to examine the accuracy of intrinsic and extrinsic calibration results, and also numerically study the sensitivity of the system to the time offset in simulations.

For the results presented in the following, we simulate an Xsens-100 IMU[1] and a 16-beam 3-D LiDAR VLP-16[2]. The synthetic noises in measurement are consistent with the real-world sensors. IMU measurements are reported at 400 Hz. LiDAR scans are received at 10 Hz and the laser rays cover 360° horizontally and ±15° vertically. The motion distortion effect is also simulated to match reality. The intrinsic parameters of IMU and LiDAR are randomly generated from normal distributions with standard deviations provided in Table I. The extrinsic parameters from LiDAR to IMU are set as $[0.3, 0.15, 0.05]^\top$ meter

[1][Online]. Available: https://www.xsens.com/products/mti-100-series/
[2][Online]. Available: https://velodyneLiDAR.com/vlp-16.html

TABLE I
STANDARD DEVIATIONS FOR GENERATING RANDOM LIDAR AND IMU INTRINSICS IN SIMULATION EXPERIMENTS

Parameter	Value	Parameter	Value
Elevation angle correction $\delta\phi$ (°)	0.1	Vertical spatial offset V (m)	0.005
Azimuth angle correction $\delta\theta$ (°)	0.05	Horizontal spatial offset H (m)	0.005
Range scale factors (%)	0.005	Range offset $\delta\rho$ (m)	0.01
Gyro. Nonlinearity (%)	0.01	Accel. Nonlinearity (%)	0.1
Gyro. Nonorthogonality (°)	0.05	Accel. Nonorthogonality (°)	0.05

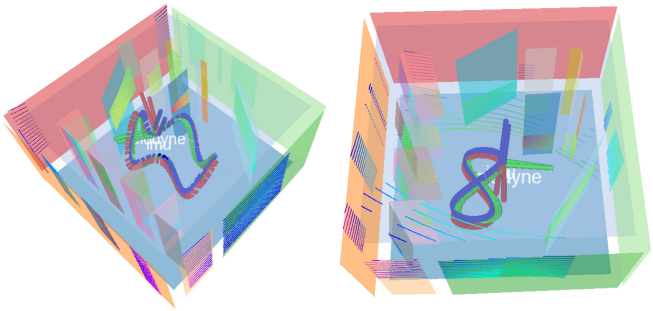


Fig. 5. Simulated environmental setup filled with planes in different colors. The colored points indicate a simulated 3-D LiDAR scan. Left: The sinusoidal trajectory for testing calibration in full excitation motion. Right: Figure-8-shape trajectory for testing calibration in degenerate motion.

in translation and $[1, 2, 5]^\top$ degree in rotation in all simulation experiments. Note that the extrinsic translation has a minimum component of 0.05 meters and a maximum component of 0.3 meters to verify the calibration accuracy of the proposed method for different magnitudes of the translation.

For the simulated trajectories, a sinusoidal trajectory and a figure-8-shape trajectory are introduced. The sinusoidal trajectory is formulated as

$$\begin{cases} p_x(t) = 2.0 \cdot \cos(\frac{\pi}{5} \cdot t) + 5 \\ p_y(t) = 1.5 \cdot \sin(\frac{\pi}{5} \cdot t) + 5 \\ p_z(t) = 0.8 \cdot \cos(\frac{4\pi}{5} \cdot t) + 5 \\ r_x(t) = 0.4 \cdot \cos(t) \\ r_y(t) = 0.6 \cdot \sin(t) \\ r_z(t) = 0.7 \cdot t \end{cases} \tag{35}$$

and the trajectory of figure-8-shape is as follows:

$$\begin{cases} p_x(t) = 2.0 \cdot \cos(\frac{\pi}{5} \cdot t) \\ p_y(t) = 1.5 \cdot \sin(\frac{\pi}{5} \cdot t) \cdot \cos(\frac{\pi}{5} \cdot t) + 5 \\ p_z(t) = 2 \\ r_x(t) = r_y(t) = 0 \\ r_z(t) = 0.4 \cdot \sin(t) \end{cases} \tag{36}$$

where $t \in [0, 10]$, $P(t) = [p_x(t) \;\; p_y(t) \;\; p_z(t)]^\top$ is the translational trajectory in meter and $r_{\text{euler}}(t) = [r_z(t) \;\; r_y(t) \;\; r_z(t)]^\top$ represents the euler angles in rad. The sinusoidal trajectory is with fully excited motions, while the figure-8-shape trajectory is a simulation of planar motion, where the robot moves on the xy plane, and orientations change along z-axis only. For both simulated trajectories, it is straightforward to derive the linear acceleration and angular velocity at any time in order to generate synthetic IMU readings. Note that in each experiment, we perform 10 Monte Carlo experiments with the same time span of 10 s.

The simulated environment is a room constructed with many planes and a volume of $12 \times 10 \times 10$ m^3, as shown in Fig. 5,

TABLE II
RESULTS OF TEN MONTE CARLO SIMULATIONS WITH THE SINUSOIDAL TRAJECTORY: MEAN AND STANDARD DEVIATION OF THE EXTRINSIC CALIBRATION RESULTS WITH AND WITHOUT CALIBRATING THE INTRINSICS OF LIDAR AND IMU

Parameter	Gt	W/ Calib. Intr.	W/O Calib. Intr.
p_x (cm)	30	**29.81** $\pm$ 0.54	35.15 $\pm$ 0.51
p_y (cm)	15	**14.86** $\pm$ 0.25	22.14 $\pm$ 0.73
p_z (cm)	5	**4.68** $\pm$ 0.28	1.67 $\pm$ 0.14
roll (deg)	1	**1.16** $\pm$ 0.02	1.97 $\pm$ 0.02
pitch (deg)	2	**1.91** $\pm$ 0.02	2.41 $\pm$ 0.03
yaw (deg)	5	**5.02** $\pm$ 0.05	5.44 $\pm$ 0.05

Note that the best results are marked in bold.

where the two types of trajectories are also plotted. The sinusoidal trajectory evaluates the intrinsic and extrinsic calibration in full motion excitation while the figure-8-shape trajectory for analyzing the extrinsic calibration performance in degenerate motion.

To evaluate the impact of LiDAR intrinsic calibration on the accuracy of map reconstruction, we introduce the mean map entropy (MME) metric [53], which evaluates the sharpness of a map. The entropy h for a map point $\mathbf{p}_k$ is calculated by

$$h(\mathbf{p}_k) = \frac{1}{2} \ln |2\pi e \Sigma(\mathbf{p}_k)| \tag{37}$$

where $\Sigma(\mathbf{p}_k)$ is the sample covariance of map points within a local radius r around $\mathbf{p}_k$. Local radius r is 0.3 m in our experimental evaluations. The MME $H(Q)$ is averaged over all map points Q

$$H(Q) = \frac{1}{n} \sum_{k=1}^{n} h(\mathbf{p}_k). \tag{38}$$

The smaller the MME value the sharper the map is (i.e., the better map reconstruction).

A. Calibration With Full Motion Excitation

1) Spatial-Temporal Calibration: The extrinsic calibration results with and without calibrating the intrinsic parameters of LiDAR and IMU in ten simulated sequences are shown in Table II. We can find that with the awareness of intrinsic parameters, the proposed method OA-LICalib achieves an accuracy of 0.40cm in translation and 0.18° in rotation, and the calibration errors extremely increase without intrinsic calibrations of LiDAR and IMU. It also confirms OA-LICalib's capability of calibrating extrinsic parameters with components at different magnitudes, e.g., translation varies from 5 to 30 cm, and rotation varies from 1° to 5° in this experiment. It is also interesting to note that the magnitudes of the rotational and translational trajectories in (35) affect the accuracy of the extrinsic calibration results. In the x-axis translation and z-axis rotation, OA-LICalib gives calibrated parameters with much lower relative error. This phenomenon agrees with the observability analysis in [54] that calibration trajectory with strong motion excitation, i.e., large accelerations and angular velocities, can lead to more accurate calibrations of extrinsic parameters than weal excitation.

Furthermore, the temporal calibration results in 7 trails are shown in Table III. Note that time offsets vary from 1 to 21 ms,

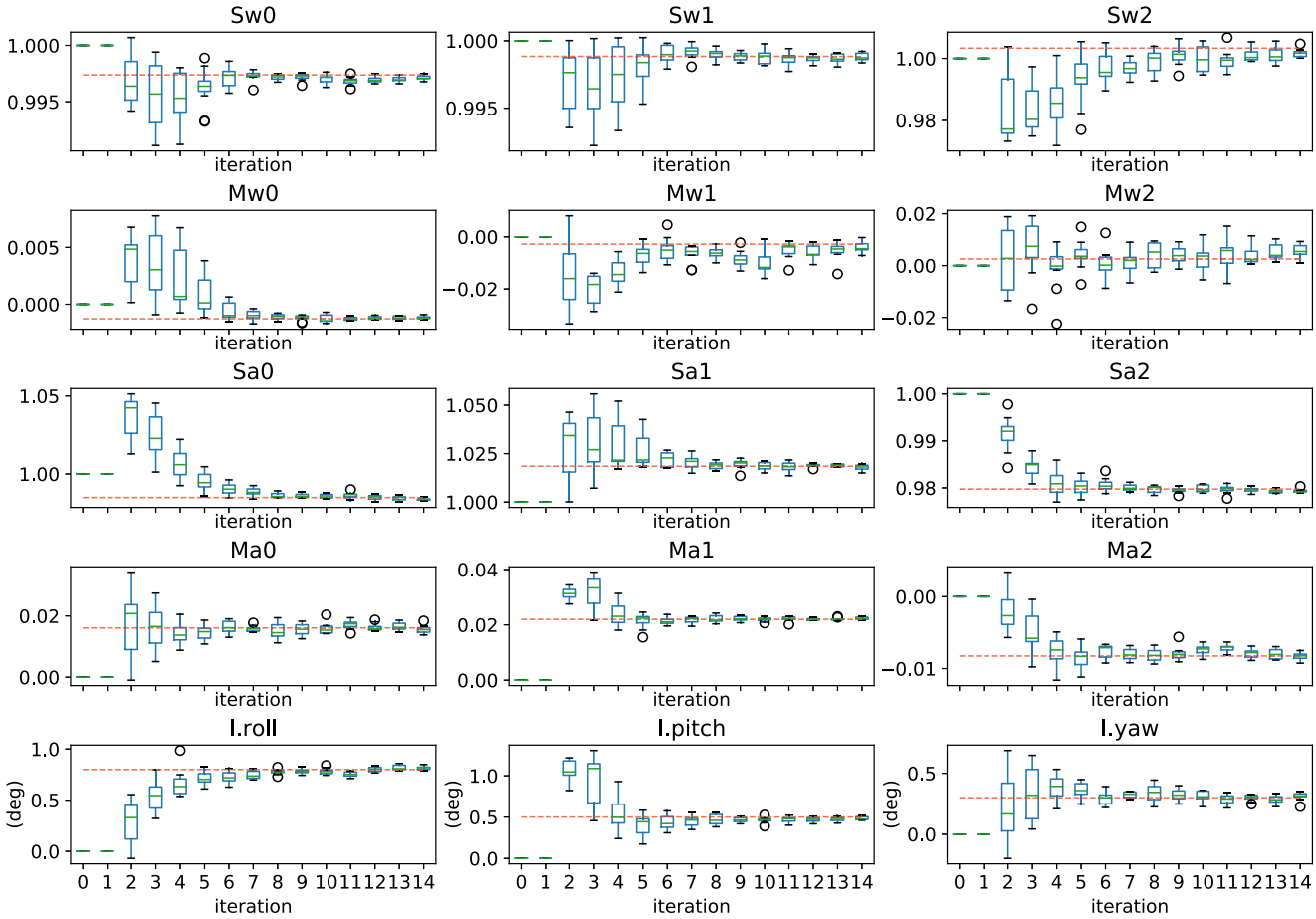


Fig. 6. Results of 10 Monte Carlo simulations with the sinusoidal trajectory: the IMU intrinsic parameters change with the increased number of iterations. The ground truth is denoted with the red dashed line. Each row from top to bottom represents scale and misalignment parameters of gyroscope, and scale and misalignment parameters of accelerometer, and euler angles of ${}^{\omega}_{I}R$ in (8), respectively.

TABLE III
RESULTS OF SIMULATIONS WITH THE SINUSOIDAL TRAJECTORY: TIME OFFSET CALIBRATIONS

	Time offset (ms)						
True value	1	2	3	5	8	12	21
Estimated	1.01	1.86	2.77	4.79	7.87	12.94	20.63
Error	0.01	−0.14	−0.23	−0.21	−0.13	−0.06	−0.37

are manually added to one of the simulated sequences and calibrated. For the precision of temporal calibration, the proposed method can achieve millisecond-level accuracy.

2) Intrinsic Calibration: Fig. 6 illustrates the convergence process of IMU intrinsic parameters along with iterations and visualizes the distribution of calibration results for 10 simulated trials. Note that I.roll, I.pitch, and I.yaw are the corresponding Euler angles of ${}^{\omega}_{I}R$ in (8). The intrinsic parameters remain unchanged during the first iteration and start to be optimized in the second iteration, and the results gradually stabilize after about eight iterations. Since the 11th iteration, the raw measurements are corrected based on the estimated intrinsic parameters (as mentioned in Section IV-C). We can see that the estimated intrinsic parameters finally converge to near the true value, and the final estimated intrinsic results are summarized in Table IV.

TABLE IV
RESULTS OF TEN MONTE CARLO SIMULATIONS WITH THE SINUSOIDAL TRAJECTORY: MEAN AND ERROR OF IMU INTRINSIC PARAMETERS AFTER 14 ITERATIONS

Parameter	Gt	Estimated	Error
Sw0	0.99738	0.99714±0.00020	−0.00024
Sw1	0.99885	0.99882±0.00031	−0.00003
Sw2	1.00329	1.00169±0.00132	−0.00160
Sa0	0.98487	0.98374±0.00071	−0.00113
Sa1	1.01851	1.01780±0.00164	−0.00071
Sa2	0.97975	0.97936±0.00037	−0.00039
Mw0	−0.00127	−0.00117±0.00015	0.00010
Mw1	−0.00279	−0.00388±0.00193	−0.00109
Mw2	0.00254	0.00577±0.00254	0.00323
Ma0	0.01607	0.01553±0.00129	−0.00054
Ma1	0.02197	0.02238±0.00049	0.00041
Ma2	−0.00825	−0.00825±0.00052	0.00001
I.roll (deg)	0.80000	0.81411±0.01924	0.01411
I.pitch (deg)	0.50000	0.48720±0.02066	−0.01280
I.yaw (deg)	0.30000	0.31130±0.03401	0.01130

The experiment indicates that the proposed method has high accuracy and repeatability in intrinsic calibration.

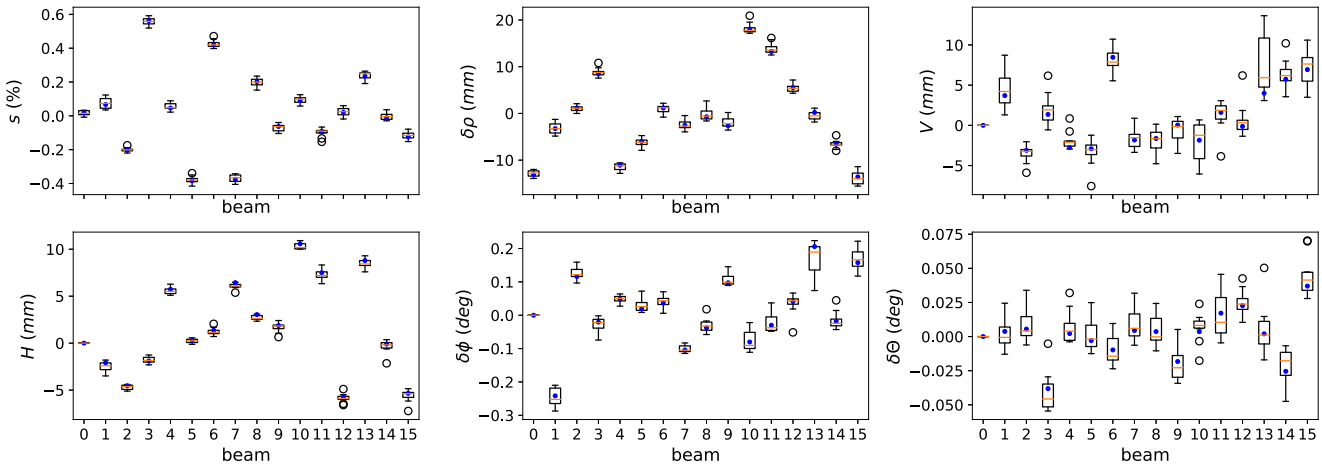


Fig. 7. Results of 10 Monte Carlo simulations with the sinusoidal trajectory: the intrinsic calibration results of a simulated 16-beam 3-D LiDAR. For the first laser, only the distance offset and distance scale are estimated while the other parameters are fixed as a reference during the calibration process. The blue dots are the ground truth and red lines are the mean of each parameter.

TABLE V
RESULTS OF TEN MONTE CARLO SIMULATIONS WITH THE SINUSOIDAL TRAJECTORY: THE AVERAGED ABSOLUTE ERRORS OF THE ESTIMATED LIDAR INTRINSIC PARAMETERS AFTER 14 ITERATIONS

Parameter	Error	Parameter	Error
s (%)	0.0058±0.0023	$\delta\rho$ (mm)	0.3048±0.2167
$\delta\phi$ (deg)	0.0075±0.0081	V (mm)	0.6383±0.7838
$\delta\theta$ (deg)	0.0029±0.0019	H (mm)	0.2111±0.1106

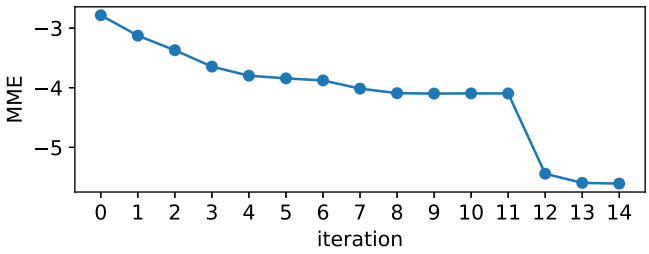


Fig. 8. MME evaluation of the LiDAR map quality along with the number of iterations in a typical simulated experiment. The evaluated point cloud map is initially assembled with the poses from NDT-based LiDAR odometry, then assembled with iteratively refined poses and intrinsics from the continuous-time batch optimization.

The final calibration results of the simulated 16-beam LiDAR intrinsic parameters in 10 trails are shown in Fig. 7. Besides, the averaged absolute errors of the estimated LiDAR intrinsic parameters in six types are summarized in Table V. The proposed method can identify six types of intrinsic parameters of multibeam LiDAR and estimate accurate results. We further evaluate the quality of LiDAR point cloud map, which is affected by both the estimated trajectory and the calibrated parameters. Fig. 8 shows the MME metric in a typical simulated experiment. The evaluated point clouds consisting of around $215k$ points are assembled by the poses from NDT-based LiDAR odometry first, and refined poses from the continuous-time batch optimization. The map becomes more accurate with increased iterations due to the batch optimization over the continuous-time trajectory and calibrated parameters. Notably, a significant improvement in map accuracy after the 11th iteration due to the correction of the raw LiDAR points based on the current best estimation of LiDAR intrinsics. It is obvious that intrinsic parameters calibration significantly affects both the LiDAR map quality and the extrinsic calibration. Therefore, we advocate calibrating the intrinsic parameters to enable highly accurate LIDAR localization and mapping.

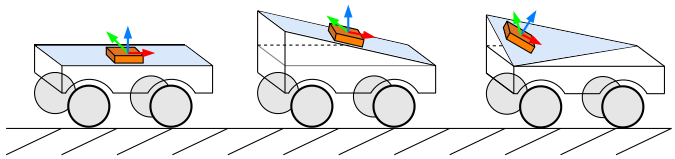


Fig. 9. Simulation to examine observability awareness: three mounting setups with different relative orientations between the robot and the LI sensor rig are tested (LiDAR and IMU are rigidly connected, while LiDAR is not shown in this figure). From left to right: case A, case B, case C. Among these three cases, the robots move in a shared trajectory of figure-8 shape as (36), while the unobservable extrinsic directions in local IMU frames are different.

B. Calibration Under Degenerate Motion

In the same simulation environment with figure-8-shape trajectory, shown in the right picture of Fig. 5, we investigate the extrinsic calibration accuracy of OA-LICalib in the degenerate case. With the trajectory in (36), where rotation happens along the z-axis only, the LI extrinsic translation along the rotation axis is unobservable according to the analysis in [4]. In order to examine whether the proposed OA-LICalib is able to automatically identify the unobservable directions, we also simulate three different mounting cases (A, B, C), as shown in Fig. 9, where the LI sensor rig is rigidly attached to the robot platform at three different relative orientations. Among the three cases, the robots move in the same trajectory [see (36)], thus, the unobservable directions of the extrinsic translation are the identical vertical axis in the robot coordinate frame. However, due to the different relative rotations between robot platform and the LI sensor rigs, the unobservable directions of the extrinsic translation, ${}^{I}\mathbf{p}_{L}$, are different when represented in local IMU frame.

TABLE VI
RESULTS OF SIMULATION WITH THE FIGURE-8 TRAJECTORY: THE GROUND TRUTH AND THE AUTOMATICALLY DETECTED UNOBSERVABLE EXTRINSIC DIRECTIONS UNDER THREE MOUNTING CASES ARE SHOWN IN FIG. 9

Case		Degenerate direction					
Case A	Truth	0.00000	0.00000	0.00000	0.00000	0.00000	1.00000
	Detected	–0.00001	0.00000	0.00005	0.00165	–0.00110	1.00000
Case B	Truth	0.00000	0.00000	0.00000	0.50000	0.00000	0.86603
	Detected	0.00000	0.00000	–0.00001	0.50117	–0.00023	0.86535
Case C	Truth	0.00000	0.00000	0.00000	0.50000	0.43301	0.75000
	Detected	0.00006	0.00004	0.00007	0.50070	0.43296	0.74957

The head part corresponds to extrinsic rotation and the tail part corresponds to extrinsic translation.

TABLE VII
RESULTS OF TEN MONTE CARLO SIMULATIONS WITH THE FIGURE-8 TRAJECTORY: MEAN OF THE ESTIMATED EXTRINSIC PARAMETERS IN THREE MOUNTING CASES WITH OR WITHOUT OBSERVABILITY-AWARE CALIBRATION

Seq.	Method	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			RMSE	
		p_x	p_y	p_z	roll	pitch	yaw	Pos (cm)	Rot (deg)
Groundtruth		30.00	15.00	5.00	1.00	2.00	5.00		
Case A	W/O Obser.	27.24	11.49	7.17	0.85	2.40	4.83	2.87	0.27
	W Obser.	30.97	12.09	8.05	0.95	2.44	4.89	**2.50**	**0.26**
Case B	W/O Obser.	26.50	11.53	6.88	0.76	2.27	4.88	3.04	0.22
	W Obser.	29.73	11.94	5.69	1.01	2.27	4.81	**1.82**	**0.19**
Case C	W/O Obser.	26.32	11.20	6.03	0.08	2.28	4.86	3.11	0.56
	W Obser.	29.67	10.42	5.56	0.18	2.28	4.88	**2.67**	**0.50**

The best results are marked in bold.

Table VI summarizes the unobservable direction of the three mounting cases in Fig. 9. For case A where the IMU z-axis is aligned with the vertical axis, the unobservable direction is purely in the z-axis of extrinsic translation, ${}^{I}\mathbf{p}_{L}$. For a typical trial, OA-LICalib identifies the singular value of the information matrix corresponding to extrinsics as $[41760,\ 22193,\ 3525,\ 491,\ 295,\ 0.000]^{\top}$, and it is obvious that there exists one unobservable direction. The eigen vectors corresponding to the minimum singular value (deemed as the detected unobservable direction) in the three cases are also summarized in Table VI. We can see that the detected unobservable directions are consistent with the truths. To verify the effectiveness of the observability scheme in Section V-B, we conduct the ablation experiments on extrinsic calibration by starting from perturbed initial extrinsic parameters (with perturbations of around 3 degrees and 3 centimeters on rotations and translations, respectively). Table VII shows the extrinsic calibration results ($\{{}^{I}_{L}\bar{q},\ {}^{I}P_{L}\}$) of 10 Monte-Carlo trials, and the root mean square error (RMSE) is calculated by comparing the mean calibration results to the ground truths. OA-LICalib with observability awareness keeps the unobservable directions of extrinsic translation around the initial values, and outperforms the method without observability awareness.

VII. REAL-WORLD EXPERIMENTS

In the real-world experiments, we validate the proposed method on multiple our own datasets as well as two public datasets. Table VIII summarizes the sequences in terms of collection configuration, duration, environment, motion type, etc. For the motion type, as follows.

1) Random Shaking: We hold the sensor by hand and shake the sensor suite aggressively both in rotations and translations;
2) Casual Waking: We hold the sensor and walk around;

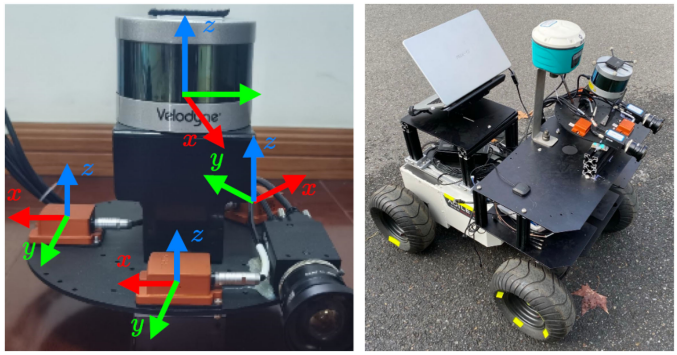


Fig. 10. Left: The self-assembled sensor rig used in real-world experiments. Right: The ground wheeled robot for collecting *SKG-01* sequence.

3) Planar Motion: The sensor rig is mounted on moving wheeled robots, see Fig. 10 (right).

The self-collected datasets are collected in both indoor and outdoor scenarios by the self-assembled sensor rig shown in the left of Fig. 10, which consists of 3 Xsens-100 IMUs sampled at 400 Hz and a Velodyne VLP-16 LiDAR sampled at 10 Hz. The *Casual-walk* sequence[3] is from LIO-SAM [5] and is collected by a Microstrain 3DM-GX5-25 IMU sampled at 500 Hz and a Velodyne VLP-16 LiDAR sampled at 10 Hz. The Hilti SLAM challenge dataset [55] is intended for assessing the accuracy of state estimation in challenging environments and it contains a number of visual, LiDAR, and inertial sensors. In the experiments, the data from Bosch BMI085 IMU sampled at 200 Hz and the Ouster OS0-64 LiDAR sampled at 10 Hz are utilized for evaluation.

The proposed method is extensively tested on the *Stairs* and *Lab* sequences to examine the accuracy and repeatability of calibrated results. Due to the absence of ground-truth extrinsic transformation between LiDAR and IMU in real-world experiments, the relative poses of three components IMU inferred from CAD assembly drawings are introduced as references. In addition, the proposed observability-aware calibration method incorporating the informative data selection and observability-aware state update mechanism is thoroughly evaluated. On the one hand, we examine the feasibility of the proposed method in degenerate motion cases over the *SKG-01* sequence and *Basement* sequence, which are collected on the vehicles under planar motion. On the other hand, we also analyze the proposed informative segment selection algorithm in the YQ-01, Casual-walk, IC_Office, Office_Mitte sequences, among which the former two sequences are outdoors while the latter two are indoors. Note that only one of the three IMUs (see Fig. 10), IMU1, is utilized to evaluate the SKG-01 and YQ-01 sequences.

We also compare the proposed method with the open-source LI calibration toolkit Imu_Lidar_Calibration [22] (abbreviated as ILC) and our previous method LI-Calib [16]. Notably, LI-Calib [16] does not have the observability-aware scheme described in Section V or the explicit initialization of control points (see Section IV-A1), compared to the proposed OA-LICalib. In addition, two online calibration methods, LIO-Mapping [3] (abbreviated as LIOM) and FAST-LIO2 [26], are also included for the comparisons. LIOM and FAST-LIO2 are LI odometry and simultaneously calibrate the extrinsic parameters between

[3][Online]. Available: http://www.udel.edu/009352

TABLE VIII
DETAILS OF SEQUENCES USED IN REAL-WORLD EXPERIMENTS

Dataset	Sequence	Config	Duration	Env.	Ave. Angular Velocity (rad/s)	Motion Type	Excitation Type
Self collection	*Vicon-(01,02,...,10)*	Hand-held	15	Indoor	0.983	Random shaking	Full
	Stairs-(01,02)	Hand-held	60	Indoor	1.118	Random shaking	Full
	Lab-(01,02)	Hand-held	60	Indoor	1.215	Random shaking	Full
	SKG-01	On-Vehicle	120	Outdoor	0.407	Planar motion	Degenerate
	YQ-01	Hand held	271	Outdoor	0.760	Casual walking and occasional random shaking	Occasionally full
LIO-SAM [5]	*Casual-walk*	Hand held	655	Outdoor	0.521	Casual walking and occasional random shaking	Occasionally full
Hilti SLAM challenge [55]	*IC_Office*	Hand held	200	Indoor	0.348	Casual walking	Occasionally full
	Office_Mitte	Hand held	264	Indoor	0.350	Casual walking	Occasionally full
	Basement[1]	On-vehicle	330	Indoor	0.116	Planar motion	Degenerate

LiDAR and IMU. In practice, online calibration methods estimate the whole trajectory of the entire calibration sequence, and the final converged values of the extrinsic parameters are deemed the calibration results, as are the ILC method. Since the ILC method particularly requires the sensor to keep static for seconds to initialize the system, we skip some data segments at the beginning in order to satisfy the static initialization requirements in some sequences. Notably, under general motion scenarios, the proposed OA-LICalib and our previous method LI-Calib are initialized from the identity extrinsic rotation matrix and zero extrinsic translations without bells and whistles. However, for the initial values of the extrinsics for these compared methods, including ILC, LIOM, and FAST-LIO2, the extrinsic rotation parameters are initialized by the reference parameters provided by the dataset, while the translation extrinsic parameters are initialized as zero. These compared methods fail to converge when starting from the identity extrinsic rotation matrix due to the lack of an extrinsic initialization module.

A. Trajectory Representation Capability

In the *Vicon* sequences, ground truth poses are provided by the Vicon system at 100 Hz, with a duration of about 15 s to evaluate the accuracy of estimated trajectories and the representation capability of the splines. Comparing the estimated trajectories with the motion-capture system's reported trajectories, the average absolute trajectory error (ATE) [56] overall ten sequences are 0.0183 m. Fig. 11 shows two typical estimated trajectories aligned with the ground truth and also illustrates fitting results. The estimated B-Spline trajectory fits well with the ground-truth trajectory and the acceleration measurements and angular velocity measurements, indicating the B-Spline's high representation capability.

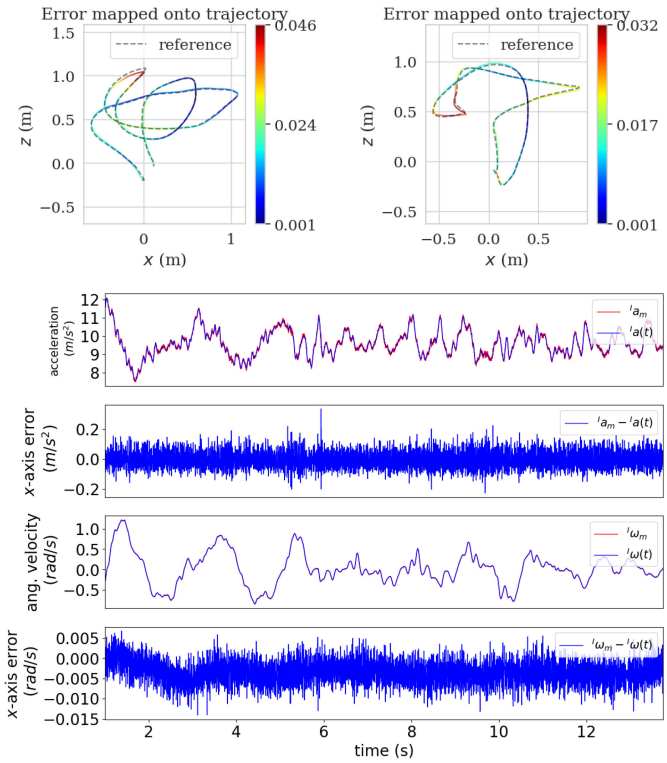


Fig. 11. Top: Estimated continuous-time trajectories aligned with the ground-truth. The color of the trajectory indicates ATE. Bottom: The corresponding fitting results of the top-left trajectory. Only the x-axis is plotted. The fitting errors almost have the same order of magnitude as the IMU measurement noises.

B. Calibration With Full Motion Excitation

We gathered four sequences of a 60-s duration with full excitation in structured environments, *Stairs-01 and Stairs-02* collected at the stairway, *Lab-01 and Lab-02* in the laboratory, verifying the calibration accuracy of the full motion excitation data. Each sequence is split into four segments of 15 s and calibrated independently for calibration test. The extrinsic calibration results of three IMUs with respect to the LiDAR by the proposed OA-LICalib are shown in Figs. 12 and 13, where the black rectangles indicate the CAD sketch of three IMUs. We can intuitively find that the translational calibration results are distributed within rectangles, with the distribution range in translation at around 2 cm. The distribution range of rotation

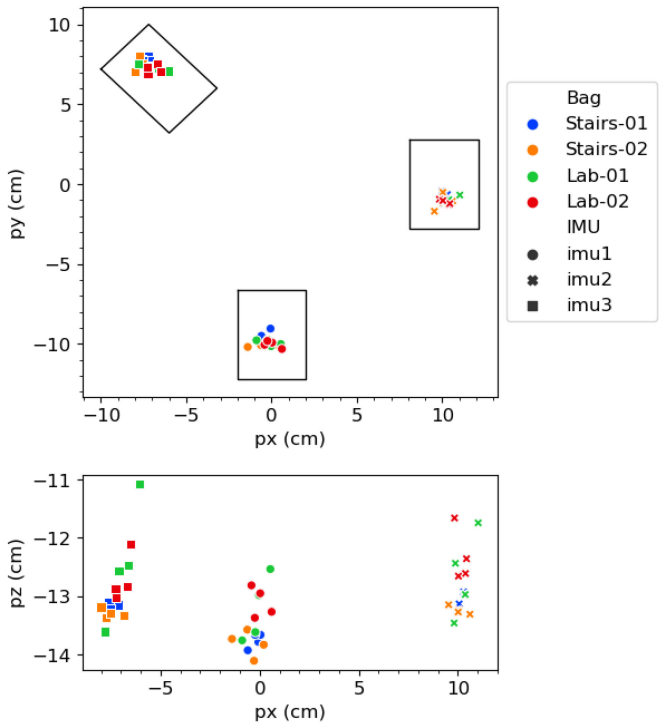


Fig. 12. Extrinisc translation calibration results from OA-LICalib over *Stairs* and *Lab* sequences. Top: Top view of the calibration results. The three black rectangular boxes represent the actual mounting positions of three IMUs, respectively. Bottom: Front view of the calibration results.

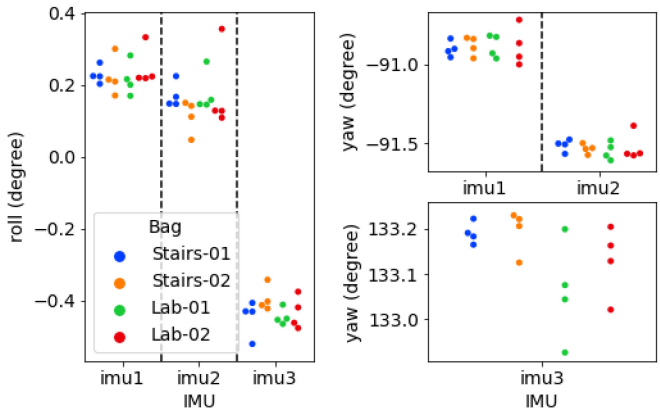


Fig. 13. Extrinsic rotation calibration results from OA-LICalib over *Stairs* and *Lab* sequences.

concentrates within 0.2° for each IMU. These results demonstrate the proposed method's high repeatability and reliability in practical deployments.

In addition, the quantitative comparison results are summarized in Table IX, where RMSE is calculated by comparing it to the CAD reference. Note that LIOM fails to converge on the *Stairs* sequences and ILC does not converge on the *Lab* sequences in our experiments. Failing to convergence means rotational RMSE is greater than 30°. The nonconverged results are excluded from the statistics in Table IX. Although the discrete-time methods, ILC, LIOM, and FAST-LIO are initialized from the reference extrinsic rotation, the continuous-time-based methods LI-Calib and OA-LICalib show apparent advantages over all of them, while starting from the identity extrinsic rotation matrix. In addition, compared to LI-Calib, OA-LICalib takes more effort on the initialization of control points (see Section IV-A1), which benefits the convergence and improves the accuracy of calibrations. It is worthwhile to mention that the calibration results of OA-LICalib are with high calibration accuracy yet low standard deviations, which demonstrate its great reliability. To give an impression of the efficiency, the proposed OA-LICalib's time consumption of main stages in two typical 15 s segment over the sequence *Lab-01* (under fully excited motion) and sequence *SKG-01* (under degenerate motion) are summarized in Table X. Basically, the proposed method takes about only one minute to generate the highly-accurate calibration results automatically on a 15 s calibration segment.

C. Calibration With Informative Segments

There is a relatively common situation in practice where some datasets provide a large number of various data sequences yet without accurate extrinsic parameters. There's no possibility of collecting new sequences with the sensors used in the existing datasets. Hence, if users want to get more accurate extrinsic parameters, they need to complete the calibration based on the existing data sequences. Special care should be taken when identifying informative sequences for accurate calibration.

Regarding the implementation details about multisegment calibration, we first perform single segment calibration and identify informative segments; afterward, with all the measurements from selected informative segments, we conduct a total continuous-time batch optimization to simultaneously estimate each segment's individual states (including the continuous-time trajectory, time offset t_c, IMU navigation states X_{Is}), and the shared spatial extrinsic parameters $\{{}^I_L\bar{q}, {}^IP_L\}$. We evaluate the proposed informativeness-aware segment selection algorithm in the self-collected sequence *YQ-01* and public-available sequence *Casual-walk and Office_Mitte*, *IC_Office* from the Hilti dataset. Like the *Casual-walk* sequence collection process, we walk in a circle profile with the occasional shake of the sensor to collect *YQ-01* sequence. Again, we split sequences into segments with a duration of 15-s for each segment.

1) YQ-01 *Sequence:* Fig. 14 details the shape of the trajectory and the corresponding angular velocity curves of the *YQ-01* sequence. The top figure in Fig. 14 illustrates each segment's information metrics and angular velocities. The segments with minimal singular value (see Section V-A) above 100 (black dash line) are selected as the informative segments and the relative location of these segments are shown in bright colors at the bottom of Fig. 14. It is interesting to see the amplitudes of angular velocity can indistinctly reflect the information metrics. The extrinsic calibration results with data of one segment (Segment-n) and multisegment (n Segments) are summarized in Table XI. Note that "n Segments" means we take the first n informative segments, e.g., "3 Segments" represents Segment-1, Segment-2, and Segment-3 are jointly utilized for calibration. The evaluation results show that the selected single-segment calibration is able to generate reasonable calibration results, which demonstrate the effectiveness of the informative segment selection. Besides, multisegment calibration is generally stable compared to single-segment calibration, and four informative segments are enough to get a reliable estimation of the extrinsics.

2) Hilti Sequences: Furthermore, we test the multisegment calibration on two handheld sequences of the Hilti dataset, the comparison results are reported in Table XII. Fig. 15 reports

TABLE IX
MEAN AND STANDARD DEVIATION OF THE CALIBRATION RESULTS FROM DIFFERENT METHODS ON *STAIRS* AND *LAB* SEQUENCES

Method	IMU	${}^{L}\mathbf{p}_{I}$ (cm)			${}^{L}_{I}\bar{q}$ in Euler (deg)			RMSE	
		p_x	p_y	p_z	roll	pitch	yaw	Pos (cm)	Rot (deg)
CAD Reference	IMU1	0	−10	−13	0	0	−90	–	–
	IMU2	10	0	−13	0	0	−90	–	–
	IMU3	−7	8	−13	0	0	135	–	–
ILC	IMU1	9.61±0.79	−13.00±4.78	−7.34±6.75	7.78±0.47	−4.95±5.83	−95.82±2.18	11.65	5.04
	IMU2	19.65±5.63	−1.69±3.36	−4.44±3.54	9.73±1.40	−3.48±3.35	−93.59±1.59	6.67	6.30
	IMU3	7.74±2.13	7.20±2.33	−10.95±**0.49**	−6.63±2.04	7.69±2.05	127.47±1.35	7.51	6.32
LIOM	IMU1	−0.74±3.77	−12.87±2.40	−2.91±4.31	9.64±1.19	0.09±0.25	−93.31±2.37	8.60	7.30
	IMU2	7.70±6.42	−5.85±2.63	−2.68±2.71	9.82±1.71	0.19±0.29	−94.06±2.50	6.98	6.13
	IMU3	−2.64±2.54	11.21±7.15	6.43±3.96	7.89±4.09	0.32±0.27	131.30±2.97	11.65	5.04
FAST-LIO2	IMU1	−1.56±1.28	−10.56±3.24	−12.94±2.78	−0.52±1.65	1.49±1.06	−91.00±0.92	0.96	1.08
	IMU2	6.76±1.89	−3.08±2.40	−12.39±1.96	−1.05±2.00	1.45±0.89	−91.34±1.32	2.61	1.29
	IMU3	−7.84±1.58	4.48±2.72	−10.41±0.85	−0.34±1.03	−0.7±2.13	132.96±0.78	2.57	1.26
LI-Calib	IMU1	−0.97±2.37	−9.80±1.49	−12.98±1.41	0.47±0.40	0.18±0.87	−90.90±0.29	0.57	0.60
	IMU2	9.64±2.09	−1.08±1.10	−12.47±1.25	0.36±0.39	0.42±0.83	−91.56±0.27	0.73	0.96
	IMU3	−6.73±2.03	7.09±1.33	−12.66±1.11	−0.56±0.60	0.89±0.81	133.14±0.31	0.58	1.23
OA-LICalib	IMU1	−0.22±**0.50**	−9.92±**0.31**	−13.48±**0.44**	0.23±**0.04**	0.21±**0.06**	−90.89±**0.07**	**0.31**	**0.54**
	IMU2	10.16±**0.36**	−1.00±**0.31**	−12.83±**0.54**	0.16±**0.07**	0.46±**0.17**	−91.53±**0.05**	**0.59**	**0.93**
	IMU3	−7.19±**0.53**	7.46±**0.34**	−12.89±0.61	−0.43±**0.04**	0.68±**0.05**	133.14±**0.09**	**0.33**	**1.17**

The units for translation and rotation are in "cm" and "degree," respectively. The nonconverged results of ILC and LIOM are excluded from the statistics.
The best results are marked in bold.

TABLE X
OA-LICALIB'S TIME CONSUMPTION OF MAIN STAGES

Step	Fully excited Seq. [0,15] of *Lab-01*			Degenerate Seq. [0,15] of *SKG-01*		
	Count	Processing /Step(s)	Total(s)	Count	Processing /Step(s)	Total(s)
Extri. Init. and Rerun Odom.	1	15.73	15.73	1	12.25	12.25
Data Association	1	1.80	1.80	1	3.31	3.31
Init. Traj. and First Opt.	1	4.47	4.47	1	6.54	6.54
Refinement	5	6.73	33.64	5	8.03	40.17
TSVD	0	-	-	5	0.03	0.13
Total	-	-	55.65	-	-	64.40

Two segments from *Lab-01* and *SKG-01* sequence are evaluated, respectively. The former segment is under fully excited motion, while the latter is under degeneration motion.

TABLE XI
ESTIMATED SPATIAL PARAMETERS $\{{}^{I}_{L}\bar{q}, {}^{I}P_{L}\}$ AND TIME OFFSET t_c USING ONE SEGMENT (SEGMENT-n) OR MULTISEGMENT (n SEGMENTS) OVER *YQ-01* SEQUENCE

Data	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			Time	RMSE	
	p_x	p_y	p_z	roll	pitch	yaw	Offset (ms)	Pos (cm)	Rot (deg)
CAD Reference	−10	0	13	180	−180	−90	–		
Segment-1	−11.0	0.3	14.6	179.75	−179.76	−89.25	7.4	1.10	0.48
Segment-2	−11.0	−0.2	14.0	179.77	−179.81	−89.23	7.6	0.82	**0.48**
Segment-3	−11.2	0.7	13.5	179.77	−179.8	−89.18	7.4	0.85	0.51
Segment-4	−11.0	0.0	14.7	179.73	−179.77	−89.2	7.8	1.14	0.51
Segment-5	−9.7	0.3	12.8	179.71	−179.81	−89.18	7.4	**0.27**	0.51
Segment-6	−10.6	−0.2	15.1	179.78	−179.82	−89.19	7.9	1.27	0.50
2 Segments	−11.3	0.2	14.4	179.76	−179.77	−89.25	–	1.11	**0.47**
3 Segments	−11.3	0.3	14.2	179.76	−179.77	−89.24	–	1.04	0.48
4 Segments	−11.2	0.2	14.3	179.75	−179.77	−89.23	–	1.03	0.49
5 Segments	−11.1	0.2	14.1	179.75	−179.79	−89.23	–	**0.91**	0.48
6 Segments	−11.1	0.2	14.3	179.76	−179.8	−89.23	–	0.99	0.48

The best results are marked in bold.

TABLE XII
COMPARISON RESULTS OF DIFFERENT METHODS ON HILTI DATASET

Method	Sequence	Data	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			Time offset (ms)	RMSE	
			p_x	p_y	p_z	roll	pitch	yaw		Pos (cm)	Rot (deg)
Reference			1.00	−0.66	9.47	0.02	−0.37	−0.14	∼1ms		
FAST-LIO2	*Office_Mitte*	All data	0.43	−1.02	6.58	0.05	−0.02	−0.35	–	1.71	0.24
	IC_Office	All data	0.69	−1.21	5.02	0.03	0.12	−0.59	–	2.59	0.39
ILC	*Office_Mitte*	All data	−0.02	−1.09	11.95	1.54	−0.50	0.12	−3.86	1.57	0.89
	IC_Office	All data	−0.24	3.65	10.95	−0.01	0.97	5.16	−11.90	2.73	3.15
OA-LICalib	*Office_Mitte*	Segment-1	0.20	−0.70	9.10	0.12	−0.07	−0.26	1.62	**0.51**	0.20
		Segment-2	0.71	0.86	5.57	0.25	−0.31	0.45	1.76	2.42	0.37
		Segment-3	−0.44	−0.66	6.96	0.04	−0.20	−0.08	1.65	1.67	**0.10**
		Segment-4	−2.64	−1.74	9.73	0.00	0.10	0.09	1.53	2.20	0.30
		2-Segment	−0.15	0.16	7.21	0.16	−0.14	−0.11	–	1.54	0.16
		3-Segment	−0.22	0.25	7.72	0.12	−0.15	−0.15	–	1.34	**0.14**
		4-Segment	−0.20	0.02	7.69	0.11	−0.13	−0.18	–	**1.30**	0.15
	IC_Office	Segment-1	1.16	−2.30	6.33	−0.05	−0.20	−0.43	1.55	2.05	0.20
		Segment-2	−1.49	0.58	8.81	−0.03	−0.42	−0.02	0.96	1.65	0.08
		Segment-3	0.77	2.17	5.67	0.10	−0.31	−0.09	1.30	2.74	**0.06**
		Segment-4	−0.17	0.17	7.34	−0.20	−0.25	−0.07	1.43	**1.48**	0.15
		2-Segment	0.48	−0.86	7.37	−0.03	−0.27	−0.45	–	**1.25**	0.19
		3-Segment	0.64	0.24	6.76	0.01	−0.27	−0.27	–	1.66	0.09
		4-Segment	0.53	0.30	6.91	−0.02	−0.27	−0.23	–	1.60	**0.08**

LIOM fails to give reasonable results, thus, not listed in the table.
The best results are marked in bold.

the information metrics of every individual segment, linear acceleration curves, and angular velocity curves in the *IC_Office* sequence, and we pick out the four most informative segments for multisegment calibration. The results in Table XII suggest multisegment calibration is more stable and more accurate than single-segment calibration. The proposed OA-LICalib with multisegment calibration also significantly outperforms ILC and FAST-LIO2.

Note that using minimal singular value to measure the informativeness of each individual segment within a single sequence is feasible, however, it is not meaningful to compare the minimal singular values of different sequences under significantly different scenarios. Minimal singular value is not a general information metric [9], and it is affected by the sensory observations, trajectory profile, surrounding environment, etc. This is reflected by the different amplitudes of minimal singular values over different sequences, as shown in Figs. 14 and 15. On the positive side, a minimal-singular-value-based information metric is sufficient for informative segments selection from a given sequence and leads to reasonable calibration results.

We further investigate the performance of noninformative segments and their effect on multisegment calibration results. Table XIII shows the calibration results using multiple segments in *IC-Office* sequences. The segments [60, 75] and [105, 120]

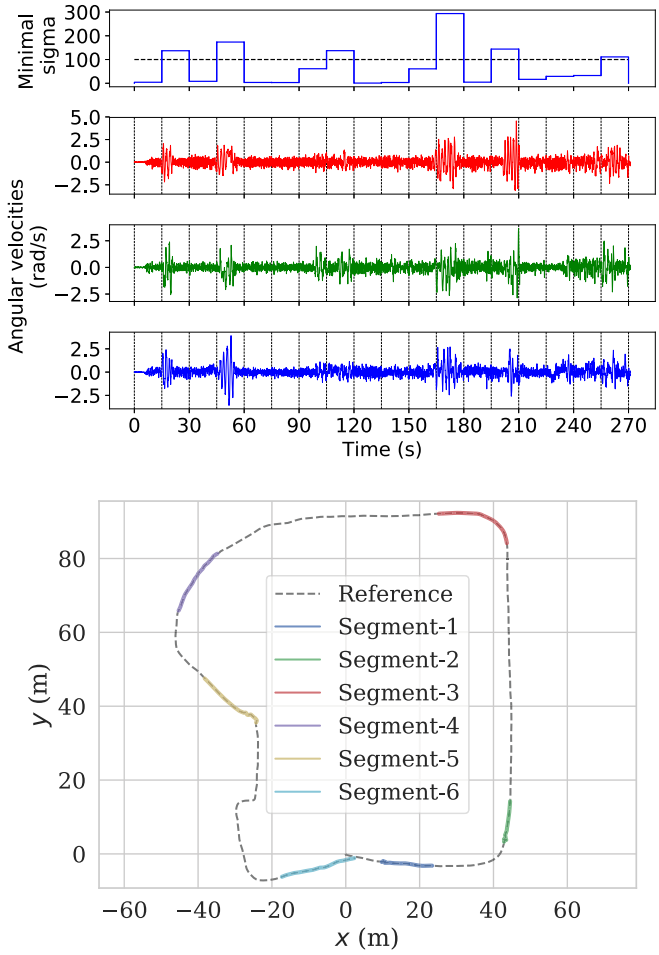


Fig. 14. Evaluation in *YQ-01* sequence. Top: The informativeness of every individual segment of *YQ-01* sequence, which has a total trajectory length of 363 meters; the corresponding three-axis angular velocity curves. Bottom: The whole trajectory is plotted in dashed line and the selected segments in bright colors. The trajectory direction is counterclockwise.

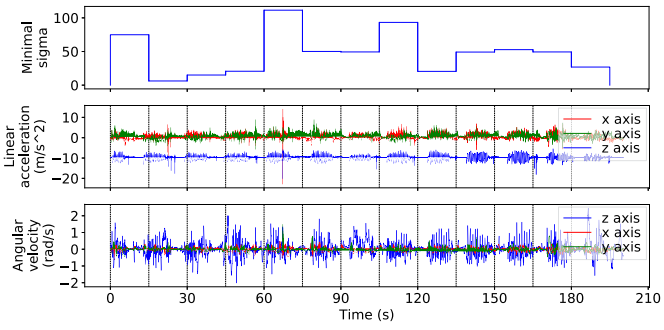


Fig. 15. Top: The informativeness of every individual segment of *IC-Office* sequence. Bottom: the corresponding linear acceleration curves and angular velocity curves.

are identified as informative segments, while the segment [15, 30] is a noninformative segment automatically by OA-LICalib, as shown in Fig. 15. The results in Table XIII suggest that the noninformative segment generates a much bigger calibration error. When the noninformative segments are included for multisegment calibration, the calibration results may be degraded compared to calibration using only informative segments. The

TABLE XIII
MULTIPLE-SEGMENT CALIBRATION USING NONINFORMATIVE SEGMENT IN *IC-OFFICE* SEQUENCES

Data	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			Time offset	RMSE	
	p_x	p_y	p_z	roll	pitch	yaw	(ms)	Pos (cm)	Rot (deg)
Reference	1.00	−0.66	9.47	0.02	−0.37	−0.14	∼1ms		
[60,75]	1.16	−2.30	6.33	−0.05	−0.20	−0.43	1.55	2.05	0.20
[105,120]	−1.49	0.58	8.81	−0.03	−0.42	−0.02	0.96	1.65	**0.08**
[15,30]	0.68	−2.84	1.47	0.09	−0.45	0.01	0.66	4.79	0.11
[60,75]+[105,120]	0.48	−0.86	7.37	−0.03	−0.27	−0.45	–	**1.25**	0.19
[60,75]+[15,30]	0.69	−2.75	3.91	−0.01	−0.33	−0.42	–	3.43	0.16

The segments [60,75] and [105,120] are identified as informative segments and segment [15,30] is identified as noninformative segment, as shown in Fig. 15.
The best results are marked in bold.

TABLE XIV
EVALUATION OF CALIBRATION RESULTS USING SINGLE OR MULTI-INFORMATIVE SEGMENTS OVER THE *CASUAL-WALK* SEQUENCE

Data	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			Time
	p_x	p_y	p_z	roll	pitch	yaw	Offset (ms)
Segment-1	−0.9	0.4	−4.7	178.88	179.83	178.52	8.3
Segment-2	−0.8	0.7	−2.9	178.92	179.81	178.53	8.1
Segment-3	−1.6	−0.1	−3.2	178.98	179.90	178.50	9.1
Segment-4	−0.2	0.0	−1.2	178.91	179.77	178.72	8.5
2 Segments	−1.0	0.5	−3.8	178.89	179.81	178.52	–
3 Segments	−1.3	0.4	−3.7	178.91	179.83	178.52	–
4 Segments	−1.1	0.7	−3.3	178.91	179.83	178.54	–

multisegment calibration result is more like a weighted average of the calibration results from their component segments, thus, the calibration result is more stable than single-segment calibration.

3) Casual-Walk *Sequence:* The *Casual-walk* sequence is one of the common cases where the dataset is publicly available but without providing carefully calibrated extrinsic parameters. The rough extrinsic parameters are $[0, 0, 0]^{\top}$ in translation and identity matrix in rotation. We try to conduct multisegment calibration on this sequence. This sequence was collected by randomly walking around the campus and occasionally shaking the sensor vigorously, meaning that the sequence is coupled with a number of segments suitable for calibration. We perform multisegment data calibration using only half of the sequence which is divided into 20-s segments, and the results are shown in Table XIV. Since there are no reliable reference values for the dataset's extrinsics or ground truth trajectory, we can only evaluate the quality of the LiDAR point cloud map, which implicitly reflects the calibration accuracy. The MMEs of the maps are -2.141, -2.163, and -2.185 for the rough extrinsics, the calibrated spatial extrinsics, and the calibrated spatial-temporal extrinsics, respectively. The map quality obtained from the calibrated spatial-temporal extrinsics is the best, which verifies the effectiveness and accuracy of the multisegment calibration.

D. Calibration in Degenerate Case

In practical applications, the on-vehicle situation is a common case with motion degeneration and there are high demands for accurate calibrations in this scenario. In this section, we also test the extrinsic calibration in on-vehicle conditions over the self-collected *SKG-01* sequence and the publicly-available *Basement* sequence. Regarding the initialization under degenerate cases, since the compared methods ILC, LIOM, and FAST-LIO2 fail to be initialized with the identity extrinsic rotation matrix, all the methods are initialized with the perturbed extrinsics, with

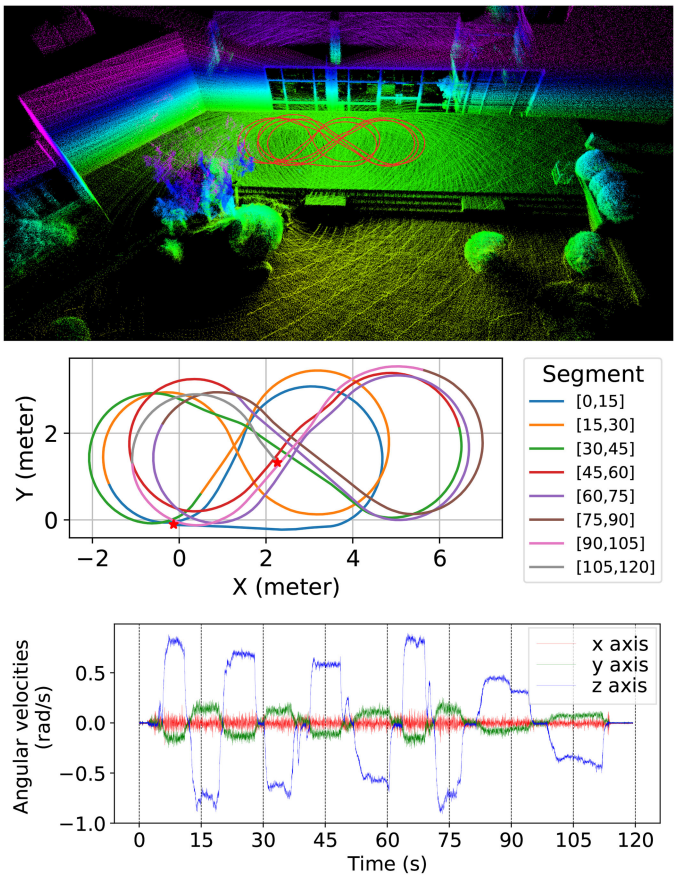


Fig. 16. Details about the *SKG-01* sequence. Top: The environment for collecting *SKG-01* sequence and the actual figure-8-shape trajectory. Middle: Concrete trajectories for each 15-s segment, the red stars indicate the start and end position of the trajectory, respectively. Bottom: Corresponding three-axis angular velocity curves.

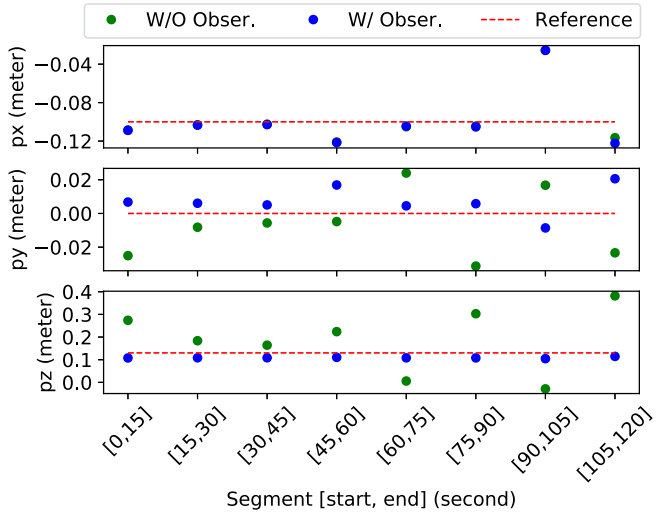


Fig. 17. Comparison of the estimated ${}^{I}P_{L}$ under degenerate case over *SKG-01* sequence: using OA-LICalib with and without observability-aware calibration. References are from the CAD sketch.

TABLE XV
SINGULAR VALUE OF EACH SEGMENT IN *SKG-01* SEQUENCE IDENTIFIED BY OA-LICALIB

Time	Singular values in a decreasing order					
[0,15]	5545	4659	3664	958	282	0.010
[15,30]	33 595	16 045	6844	930	724	0.016
[30,45]	27 126	10 997	7585	1915	678	0.016
[45,60]	47 269	25 342	5606	726	365	0.009
[60,75]	9332	9186	3634	929	452	0.008
[75,90]	60 975	15 907	5675	563	234	0.004
[90,105]	18 542	7932	760	290	125	0.006
[105,120]	20 997	10 650	3940	811	8	0.012

The secondary minimum singular value of the segments [90, 105] and [105, 120] are relatively small.

rotational perturbations of 2 degrees per axis and translational perturbations of 2 cm per axis. In general, all the methods start from the same initial extrinsic parameters for fair comparisons.

1) SKG-01 *Sequence:* To ensure the LiDAR sensor has sufficiently large visibility of the ground for localization, we mount the sensor suite at a slight incline downwards, as shown in Fig. 10. We collect the *SKG-01* sequence with a duration of 120 s, while the ground robot moves on a planar ground surface. This sequence is split into six 15-s segments, which are individually used for calibration. Fig. 16 illustrates the details of *SKG-01* sequence and Fig. 17 displays translation calibration results of OA-LICalib with and without observability awareness (see Section V-B). Table XV reports the singular value of each segment in *SKG-01* sequence identified by OA-LICalib. For the first 6 segments, the eigen vector corresponding to the minimum singular value is almost the same that is $[0.000,\ 0.000,\ 0.000,\ 0.007,\ -0.187,\ 0.982]$. Note that the head part of the eigen vector corresponds to extrinsic rotation and the tail part corresponds to extrinsic translation, which is determined by the stack order of parameters δX in (32). The identified degradation direction by the eigen vector is close to the z-axis, which is reasonable and coincides with the fact that a certain angular offset exists between the sensor frame and the ground plane. As shown in Fig. 17, translation results in the z-axis of OA-LICalib without observability awareness has a large deviation from the reference value. With the awareness of observability, OA-LICalib is able to keep the main degenerate axis, z-axis, as a prior and eliminate the coupled errors in the x-axis, resulting in more accurate calibrated extrinsic parameters. However, for segments at time intervals [90, 105] and [105,120], OA-LICalib estimates poor results which is most likely due to insufficient excitation. From the angular velocities curves of the *SKG-01* sequence shown in Fig. 16, we can find that the magnitudes of angular velocity are relatively small for these two segments, leading to a deterioration of the observability, which is also reflected by the singular values shown in Table XV.

2) Basement *Sequence:* We further test in the on-vehicle *Basement* sequence of the Hilti dataset. The beginning of the *Basement* is skipped due to less motion. Fig. 18 reports the calibration results. OA-LICalib with observability awareness automatically identifies the principal nonobservable direction as the z-axis of extrinsic translation and succeeds in keeping it to the initial value. Besides, we notice that the errors of final calibration results in the observable x-axis and y-axis of translations are larger than those on the *SKG-01* sequence. Because the Hilti dataset is not specifically collected for calibration, and the motion excitation in this sequence is insufficient, which is also

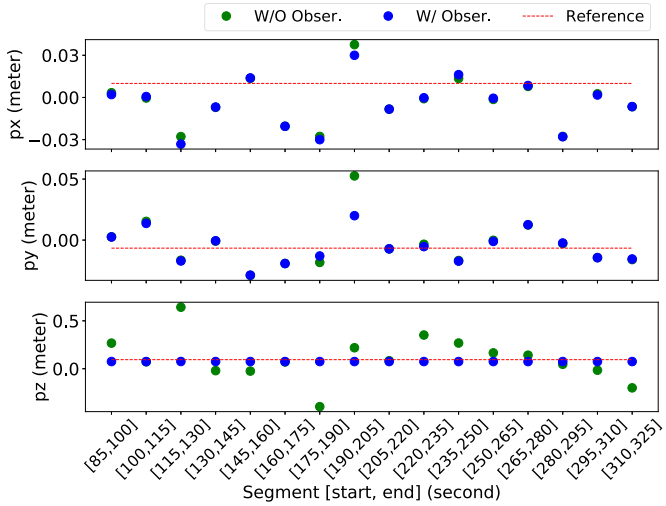


Fig. 18. Comparison of the estimated ${}^{I}P_{L}$ under degenerate case over *Basement* sequence: using OA-LICalib with and without observability awareness. The beginning of the sequence is skipped due to less motion.

TABLE XVI
COMPARISON OF THE CALIBRATION RESULTS UNDER DEGENERATE CASE OVER *SKG-01* AND *BASEMENT* SEQUENCES

Seq.	Method	Data	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			RMSE	
			p_x	p_y	p_z	roll	pitch	yaw	Pos (cm)	Rot (deg)
SKG-01	Reference		−10.00	0.00	13.00	0.00	0.00	90.00		
	FAST-LIO2	[0,120]	4.24	2.09	11.79	0.59	0.97	90.77	8.34	0.79
	ILC	[0,120]	−10.50	−0.60	15.35	−0.51	0.13	91.13	**1.43**	**0.72**
	W/O Obser.	[0,15]	−10.85	−2.50	27.39	−0.16	0.29	90.81	8.45	0.50
	W/O Obser.	[15,30]	−10.30	−0.82	18.39	−0.12	0.31	90.64	3.15	0.42
	W/O Obser.	[30,45]	−10.24	−0.56	16.41	−0.19	0.28	91.01	**2.00**	0.61
	W/O Obser.	[45,60]	−12.10	−0.48	22.39	−0.10	0.26	90.56	5.56	**0.36**
	W/ Obser.	[0,15]	−10.88	0.68	10.78	−0.16	0.29	90.80	1.43	0.50
	W/ Obser.	[15,30]	−10.34	0.61	10.84	−0.12	0.31	90.64	1.31	0.41
	W/ Obser.	[30,45]	−10.29	0.51	10.86	−0.19	0.28	91.01	**1.28**	0.61
	W/ Obser.	[45,60]	−12.15	1.69	11.04	−0.10	0.26	90.56	1.94	**0.36**

Seq.	Method	Data	${}^{I}\mathbf{p}_{L}$ (cm)			${}^{I}_{L}\bar{q}$ in Euler (deg)			RMSE	
			p_x	p_y	p_z	roll	pitch	yaw	Pos (cm)	Rot (deg)
Basement	Reference		1.00	−0.66	9.47	0.02	−0.37	−0.14		
	FAST-LIO2	[85,325]	0.96	1.53	7.33	1.22	0.91	1.51	**1.77**	**1.39**
	ILC	[85,325]	11.05	10.67	−46.97	4.42	5.38	9.01	33.74	6.74
	W/O Obser.	[85,100]	0.34	0.27	26.72	0.16	−0.34	0.27	9.98	0.25
	W/O Obser.	[100,115]	−0.04	1.53	7.28	0.31	−0.46	−0.29	**1.88**	0.19
	W/O Obser.	[115,130]	−2.78	−1.65	64.22	0.10	−0.41	1.25	31.69	0.80
	W/O Obser.	[130,145]	−0.70	−0.08	−1.99	0.19	−0.22	−0.12	6.70	**0.13**
	W/ Obser.	[85,100]	0.21	0.25	7.52	0.16	−0.34	0.26	**1.32**	0.24
	W/ Obser.	[100,115]	0.06	1.37	7.47	0.31	−0.45	−0.29	1.73	0.19
	W/ Obser.	[115,130]	−3.32	−1.71	7.56	0.10	−0.41	1.30	2.79	0.83
	W/ Obser.	[130,145]	−0.68	−0.06	7.51	0.19	−0.22	−0.12	1.53	**0.13**

The best results are marked in bold.

reflected by the small magnitude of averaged angular velocity, as shown in Table VIII.

Table XVI reports the calibration results for the different methods. Again, LIOM fails to give reasonable results thus are excluded from the table. The main degenerate direction of *SKG-01* and *Basement* sequences is the extrinsic translation along the z-axis. Without the observability-aware module, our method, FAST-LIO2, and ILC fail to give good calibrations along the z-axis of extrinsic translation. With observability awareness, OA-LICalib is able to generate accurate calibration results in on-vehicle conditions.

E. Remarks

It is clear from the previous simulation and experimental results that fully excited motions, whenever possible, are generally recommended for accurate LI calibration, which agrees with the calibration literature [10], [28], [54]. For example, the proposed OA-LICalib is able to provide accurate calibration results from any segment with fully excited motion in sequence *Lab* and *Stairs*, as shown in Figs. 12 and 13 and Table IX. Among the sequences with occasional fully excited motion, e.g., *YQ-01*, *IC_Office*, and *Office_Mitte* sequences, the automatically selected informative segments can also generate reasonable calibration results as summarized in Tables XI and XII. In contrast, the OA-LICalib has larger errors on segments under degenerate planar motions, such as [90,105] of the *SKG-01* sequence (see Fig. 17) and [115,130] of the *Basement* sequence (see Fig. 18), which are the segments with small angular velocities (see Fig. 16).

While the fully exited motions are recommended, they might not be feasible in practice and it, thus, becomes necessary to address degeneracy. The proposed OA-LICalib with observability-awareness is able to select informative segments based on the singular values of the information matrix and tackle the degeneracy via TSVD. Notably, the information metric of singular values is affected by the sensory observations, motion profiles and environments. Some other information metrics, for example, the trace, determinant, and maximal eigenvalue of the information matrix can also be used [9].

It should be noted that the proposed OA-LICalib relies on the point-to-surfel constraints, which well constrain the LI calibration problem in structured environments. This implies that planar patches (surfels) exist in the environments and can be identified. From our experience, no matter it is indoors or outdoors, the OA-LICalib usually works well when there are enough man-made structures (e.g., walls, buildings, staircases) within 50 m and the calibration data collection with fully excited motion lasts for 10–20 s. However, in some cluttered environments without clear planar regions or some scenarios with fairly sparse structures, the proposed approach would have degraded performance. For example, it may fail to provide reasonable results on some outdoor sequences of the Hilti dataset [55], in which the scenarios are nearly empty without clear human-made structures. To address this issue, resulting from insufficient point-to-surfel constraints from LiDAR data, some ideas from advanced point cloud registration methods [57] could be leveraged, such as Sdrsac [58], Teaser++ [59], [60], OPRANSAC [61], CVO [62], and deep-learning-based methods [63].

VIII. CONCLUSION

In this article, we have developed an observability-aware targetless LI calibration method, termed OA-LICalib, within the continuous-time batch optimization framework. The proposed OA-LICalib calibrates not only the spatial-temporal extrinsic parameters but also the intrinsic parameters of both IMU and LiDAR sensors, while enforcing observability constraints during update to address possible degeneracy. Specifically, the following two observability-aware strategies are employed: 1) the informative data segment selection, and 2) the observability-aware update in back-end optimization. The former selects the most informative data segments automatically for calibration among the long-session data sequence, which lowers the computational consumption for calibration as well as helps nonexpert end users. The latter addresses the degenerate motions via TSVD that is used to update only the observable directions of the sensing parameters. The proposed method has been extensively validated on both Monte Carlo simulations and real-world experiments, showing that the proposed OA-LICalib is able to provide accurate spatial-temporal extrinsic and intrinsic calibration with high

repeatability, even under certain degenerate cases. For future work, the active calibration guiding users to collect informative data deserves to be investigated.

REFERENCES

[1] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," *Robot.: Sci. Syst.*, vol. 2, 2014, Art. no. 9.

[2] T. Shan and B. Englot, "LeGO-lOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.

[3] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d Lidar inertial odometry and mapping," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 3144–3150.

[4] X. Zuo *et al.*, "LIC-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5112–5119.

[5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled Lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.

[6] J. Lin, X. Liu, and F. Zhang, "A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 123–130.

[7] J. Lv, K. Hu, J. Xu, Y. Liu, X. Ma, and X. Zuo, "Clins: Continuous-time trajectory estimation for lidar-inertial system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6657–6663.

[8] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 409–416.

[9] T. Schneider, M. Li, C. Cadena, J. Nieto, and R. Siegwart, "Observability-aware self-calibration of visual and inertial sensors for ego-motion estimation," *IEEE Sensors J.*, vol. 19, no. 10, pp. 3846–3860, May 2019.

[10] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online IMU intrinsic calibration: Is it necessary?," in *Proc. Conf. Robot. Sci. Syst.*, 2020, pp. 716–20.

[11] G. Atanacio-Jiménez *et al.*, "Lidar velodyne HDL-64e calibration using pattern planes," *Int. J. Adv. Robotic Syst.*, vol. 8, no. 5, 2011, Art. no. 59.

[12] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," in *Experimental Robotics*. Berlin, Germany: Springer, 2014, pp. 179–193.

[13] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUS and of individual axes," in *Proc. Int. Conf. Robot. Autom.*, 2016, pp. 4304–4311.

[14] W. Liu and Y. Li, "Error modeling and extrinsic-intrinsic calibration for lidar-IMU system based on cone-cylinder features," *Robot. Auton. Syst.*, vol. 114, pp. 124–133, 2019.

[15] W. Liu, Z. Li, R. Malekian, M. Sotelo, Z. Ma, and W. Li, "A novel multi-feature based on-site calibration method for LiDAR-IMU systems," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9851–9861, Nov. 2020.

[16] J. Lv, J. Xu, K. Hu, Y. Liu, and X. Zuo, "Targetless calibration of lidar-imu system based on continuous-time batch estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9968–9975.

[17] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3D LiDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 452–467, 2012.

[18] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[19] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.

[20] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "3D LiDAR-IMU calibration based on upsampled preintegrated measurements for motion distortion correction," in *Proc. Int. Conf. Robot. Autom.*, 2018, pp. 2149–2155.

[21] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006, no. 3.

[22] S. Mishra, G. Pandey, and S. Saripalli, "Target-free extrinsic calibration of a 3D-LiDAR and an IMU," in *Proc. IEEE Int. Conf. Multisensor Fusion Integration Intell. Syst.*, 2021, pp. 1–7.

[23] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 4666–4672. [Online]. Available: https://github.com/rpng/open_vins

[24] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: Lidar-inertial-camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5848–5854.

[25] K. Qiu, T. Qin, J. Pan, S. Liu, and S. Shen, "Real-time temporal and rotational calibration of heterogeneous sensors using motion correlation analysis," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 587–602, Apr. 2021.

[26] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Trans. Robot.*, to be published, doi: 10.1109/TRO.2022.3141876.

[27] J. Rehder, R. Siegwart, and P. Furgale, "A general approach to spatiotemporal calibration in multisensor systems," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 383–398, Apr. 2016.

[28] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *Proc. Int. Conf. Robot. Autom.*, 2012, pp. 2088–2095.

[29] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.

[30] J. Rehder, P. Beardsley, R. Siegwart, and P. Furgale, "Spatio-temporal laser to visual/inertial calibration with applications to hand-held, large scale scanning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 459–465.

[31] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2013, pp. 473–480.

[32] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 809–816.

[33] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-LiDAR systems with online extrinsic calibration," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 351–371, Feb. 2022.

[34] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," 2019, *arXiv:1911.08860*.

[35] A. Haarbach, T. Birdal, and S. Ilic, "Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 381–389.

[36] H. Ovrén and P.-E. Forssén, "Trajectory representation and landmark projection for continuous-time structure from motion," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 686–701, 2019.

[37] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11148–11156.

[38] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *Proc. 22nd Annu. Conf. Comput. Graph. Interactive Techn.*, 1995, pp. 369–376.

[39] D. Zachariah and M. Jansson, "Joint calibration of an inertial measurement unit and coordinate transformation parameters using a monocular camera," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, 2010, pp. 1–7.

[40] C. Glennie and D. D. Lichti, "Static calibration and analysis of the velodyne HDL-64e s2 for high accuracy mobile scanning," *Remote Sens.*, vol. 2, no. 6, pp. 1610–1624, 2010.

[41] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "Lips: Lidar-inertial 3d plane slam," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 123–130.

[42] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, School Sci. Technol., Örebro Univ., Örebro, Sweden, 2009.

[43] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 39–51, Jan. 2016.

[44] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3D attitude estimation," Dept. Comp. Sci. Eng., Univ. Minnesota, Tech. Rep. 2005-002, Mar. 2005.

[45] M. Bosse and R. Zlot, "Continuous 3d scan-matching with a spinning 2D laser," in *Proc. Int. Conf. Robot. Autom.*, 2009, pp. 4312–4319.

[46] D. De Menezes, D. M. Prata, A. R. Secchi, and J. C. Pinto, "A review on robust m-estimators for regression analysis," *Comput. Chem. Eng.*, vol. 147, 2021, Art. no. 107254.

[47] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to UAVs," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1770–1777, Jul. 2017.

[48] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

[49] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual-inertial navigation," *Int. J. Robot. Res.*, vol. 38, no. 5, pp. 563–586, 2019.

[50] C. Jauffret, "Observability and fisher information matrix in nonlinear regression," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 756–759, Apr. 2007.

[51] P. C. Hansen, "The truncatedsvd as a method for regularization," *BIT Numer. Math.*, vol. 27, no. 4, pp. 534–553, 1987.
[52] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale, "Online self-calibration for robotic systems," *Int. J. Robot. Res.*, vol. 35, no. 4, pp. 357–380, 2016.
[53] D. Droeschel, J. Stückler, and S. Behnke, "Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 5221–5226.
[54] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, 2011.
[55] M. Helmberger *et al.*, "The hilti slam challenge dataset," 2021, *arXiv:2109.11316*.
[56] M. Grupp, "EVO: Python package for the evaluation of odometry and slam." 2017. [Online]. Available: https://github.com/MichaelGrupp/evo
[57] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," 2021, *arXiv:2103.02690*.
[58] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 124–133.
[59] H. Yang and L. Carlone, "A polynomial-time solution for robust registration with extreme outlier rates," in *Proc. Robot.: Sci. Syst. (RSS)*, 2019.
[60] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and certifiable point cloud registration," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 314–333, Apr. 2020.
[61] J. Li, Q. Hu, and M. Ai, "Point cloud registration based on one-point RANSAC and scale-annealing biweight estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9716–9729, Nov. 2021.
[62] R. Zhang *et al.*, "A new framework for registration of semantic point clouds from stereo and RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 12214–12221.
[63] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3D point clouds with low overlap," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4267–4276.

Jiajun Lv received the B.Eng. degree in automation from the Zhejiang University of Technology, Hangzhou, China, in 2018. She is currently working toward the Ph.D. degree in electronic and information engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China, since September 2018.

Her major research interests include sensor calibration, sensor fusion, spatial perception, and cognition.

Xingxing Zuo received the B.Eng. degree in mechanical engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2016, and the Ph.D. degree in control science and engineering with Zhejiang University, Hangzhou, China, in 2021.

He is currently a Postdoctoral Researcher with the Department of Informatics, Technical University of Munich, Munich, Germany. His research interests include computer vision, state estimation, sensor fusion, deep learning, localization, and mapping for autonomous robots in complex environments.

Dr. Zuo was the Finalist for the ICRA 2021 Best Paper Award in Robot Vision.

Kewei Hu received the B.S. degree in electronic and information engineering from the Zhejiang University of Technology, Hangzhou, China, in 2019. He is currently working toward the M.S. degree in control science with the Institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University, Hangzhou, China.

His latest research interests include SLAM and sensor fusion.

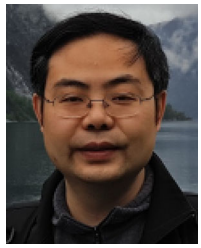
Jinhong Xu received the M.A.Sc. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2018.

He is currently a Researcher with the Institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. His latest research interests include SLAM, information processing, and robotic control.

Guoquan (Paul) Huang (Senior Member, IEEE) received the B.Eng. degree in automation (electrical engineering) from the University of Science and Technology Beijing, Beijing, China, in 2002, and the M.Sc. and Ph.D. degrees in computer science from the University of Minnesota–Twin Cities, Minneapolis, MN, USA, in 2009 and 2012, respectively.

He currently is an Associate Professor of Mechanical Engineering (ME), Computer and Information Sciences (CIS), and Electrical and Computer Engineering (ECE), with the University of Delaware (UD), Newark, DE, USA, where he is leading the Robot Perception and Navigation Group (RPNG). From 2012 to 2014, he was a Postdoctoral Associate with the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA. His research interests focus on state estimation and spatial AI for robotics, including probabilistic sensing, localization, mapping, perception and planning of autonomous ground, aerial and underwater vehicles.

Dr. Huang was the recipient of many honors and awards, including the 2015 UD Research Award (UDRF), 2015 NASA DE Space Research Seed Award, 2016 NSF CRII Award, 2018 SATEC Robotics Delegation (one of ten U.S. experts invited by ASME), 2018 Google Daydream Faculty Research Award, 2019 Google AR/VR Faculty Research Award, 2019 Winner for the IROS FPV Drone Racing VIO Competition, and was the Finalists for the RSS 2009 Best Paper Award, the ICRA 2021 Best Paper Award in Robot Vision, and the ICRA 2022 Best Paper Award in Navigation.



Yong Liu received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2001 and 2007, respectively.

He is currently a Professor with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University. His research interests include machine learning, robotics vision, multiple-sensor fusion, and intelligent systems.