注意论文里没有时间标定

Targetless Calibration of LiDAR-IMU System Based on Continuous-time Batch Estimation

Jiajun Lv, Jinhong Xu, Kewei Hu, Yong Liu, Xingxing Zuo

***Abstract*— Sensor calibration is the fundamental block for a multi-sensor fusion system. This paper presents an accurate and repeatable LiDAR-IMU calibration method (termed LI-Calib), to calibrate the 6-DOF extrinsic transformation between the 3D LiDAR and the Inertial Measurement Unit (IMU). Regarding the high data capture rate for LiDAR and IMU sensors, LI-Calib adopts a continuous-time trajectory formulation based on B-Spline, which is more suitable for fusing high-rate or asynchronous measurements than discrete-time based approaches. Additionally, LI-Calib decomposes the space into cells and identifies the planar segments for data association, which renders the calibration problem well-constrained in usual scenarios without any artificial targets. We validate the proposed calibration approach on both simulated and real-world experiments. The results demonstrate the high accuracy and good repeatability of the proposed method in common human-made scenarios. To benefit the research community, we open-source our code at `https://github.com/APRIL-ZJU/lidar_IMU_calib`**

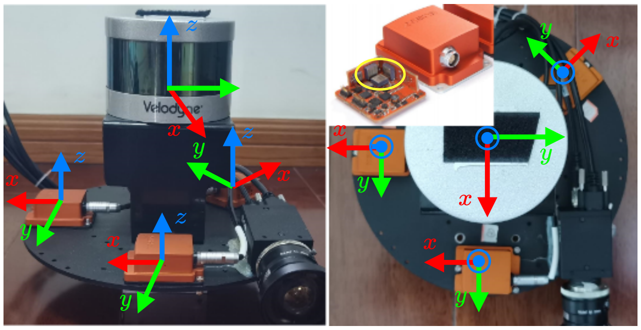


Fig. 1: The self-assembled sensors used in real-world experiments. Note that the camera is left unused in this system. The proposed methods try to calibrate the 6-DOF rigid transformation between LiDAR and IMUs.

I. INTRODUCTION

Multi-sensor fusion has been an essential developing trend in the robotics field. In particular, the LiDAR sensor is widely employed due to its high accuracy, robustness, and reliability, such as for localization, semantic mapping, object tracking, detection [1–5]. Despite these advantages of LiDAR sensors, the downside is from the fact that the LiDAR samples a succession of 3D-points at different times, thus the motion of the sensor carrier introduces distortion like the rolling-shutter effect. To address this issue, the Inertial Measurement Unit (IMU), widely used for the ego-motion estimation at a high frequency, can be utilized as a complementary sensor to correct the distortion. In general, a LiDAR-IMU system benefits from the component sensors and becomes feasible for reliable perception in various scenarios.

The accuracy of ego-motion estimation, localization, mapping, and navigation are highly dependent on the accuracy of the calibration between sensors. Manually measuring the relative translation and rotation between sensors is inaccurate and sometimes impractical. For the LiDAR-IMU calibration problem, it should be noted that the individual points in a LiDAR scan are sampled at different time instants, and the exact position of a point in the LiDAR frame is related to the pose of LiDAR at the perceived time instant. Pose estimation whenever a point is perceived is necessary for removing the motion distortion. Discrete-time based methods, like [6], leverage interpolation between poses at key time instants, which sacrifices the accuracy especially in highly-dynamic cases. Gentil et al. [7] employ Gaussian Process (GP) regression to interpolate IMU measurements, which partially address the issue of using low-frequency IMU readings to interpolate for the higher-frequency LiDAR points. Rehder et al. [8] propose to calibrate the camera-IMU-LiDAR system with two stages based on continuous-time batch estimation. The camera-IMU system is calibrated with a chessboard firstly, then the single-beam LiDAR is calibrated with respect to the camera-IMU system.

Inspired by [8], we propose an accurate LiDAR-IMU calibration method (LI-Calib) in the continuous-time framework and the proposed method is an extension of our previous work [9]. By parameterizing the trajectory with temporal basis functions, the continuous-time based method is capable of getting exact poses at any time instants along the whole trajectory, which is ideal for the calibration problem with high-rate measurements. Additionally, since the angular velocity and linear acceleration measurements from IMU are samples of the derivative of the temporal basis functions, the continuous-time formulation allows for a natural way to incorporate the inertial measurements. To summarize, the main contributions of this paper are three-fold:

- We propose an accurate and repeatable LiDAR-IMU calibration system based on continuous-time batch estimation without additional sensors or specially-designed targets.
- A novel formulation of the LiDAR-IMU calibration problem based on the continuous-time trajectory is proposed and, the residuals induced by IMU raw measurements and LiDAR point-to-surfel distances are minimized jointly in the continuous-time batch optimization,

The authors are with the Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China. (Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn)

which renders the calibration problem well-constrained in general human-made scenarios.

- Both simulated and real-world experiments are carried out, which demonstrate the high accuracy and repeatability of the proposed method. We further make the code open-sourced to benefit the community. To the best of our knowledge, this is the first open-sourced 3D LiDAR-IMU calibration toolbox.

II. RELATED WORK

Motion compensation and data association for LiDAR points are two main difficulties when calibrating the LiDAR-IMU system. For the former problem, most existing works [6, 10] perform linear interpolation for individual LiDAR points based on the assumption that the angular velocity and the linear velocity or linear acceleration remain constant over a certain time interval. This assumption holds for low-speed and smooth movements. When it comes to random motion (e.g., when holding the sensor with hands or when mounting the sensor on a quadrotor), it can easily be broken. To address this issue, Neuhaus et al. [11] tightly integrate inertial data to predict the motion of the LiDAR for relaxing the linear motion assumption. Some recent works [7, 12, 13] model the trajectory with continuous-time representations that efficiently handle high-frequency distorted measurements without any assumption of the motion priors. Our proposed calibration method also employs the continuous-time formulation. Data association for LiDAR points is the other essential problem that requires much attention. Point-to-plane and plane-to-plane data associations are utilized in [7, 14]. To find more reliable correspondences, Deschaud [15] carries out a point-to-model strategy considering the shape information of the local neighbor points. In this paper, we adopt a special point-to-surfel data association. To be specific, the point cloud is discretized and denoted by surfels (tiny planes), and the component points are associated with its corresponding surfel.

In order to calibrate a setup of rigidly-connected LiDAR and IMU, Geiger et al. [16] propose a motion-based calibration approach that performs the extrinsic calibration by hand-eye calibration [17]. However, their approach expects that each sensor's trajectory could be estimated independently and accurately, which is difficult for the commercial or consumer-grade IMU. Gentil et al. [7] adopt Gaussian Process(GP) regression to upsample IMU measurements for removing the motion distortion in LiDAR scans and formulate the calibration as a factor-graph based optimization problem, which consists of LiDAR point-to-plane factors and IMU preintegration factors.

Continuous-time batch optimization with temporal basis functions is also widely studied in calibration problems. Furgale et al. [18] detail the derivation and realization for a full SLAM problem based on B-spline basis functions and evaluate the proposed framework within a Camera-IMU calibration problem, which is further extended to support both temporal and spatial calibration [19]. Subsequently, Rehder et al. [8] adopt a similar framework for calibrating the extrinsic between a camera-IMU system and a single beam LiDAR. And Rehder et al. [20] further extend the previous work [18] to support multiple IMUs calibration. In this work, we propose a continuous-time batch optimization-based LiDAR-IMU calibration method, which is similar to [8] in spirit. However, chessboards and auxiliary sensors are not required in the proposed approach.

III. TRAJECTORY REPRESENTATION AND NOTATION

The notation used in this paper can be defined as follows: IMU frame $\{I\}$ is rigidly connected to the LiDAR frame $\{L\}$, while the map frame $\{M\}$ is located at the first LiDAR scan $\{L_0\}$ at the beginning of calibration. Besides, ${}^I_L\boldsymbol{q}$, ${}^I\boldsymbol{p}_L$ denote the extrinsic rotation and translation from the LiDAR frame to the IMU frame. With a little bit abuse of notation, the matrix ${}^I_L\boldsymbol{R} \in SO(3)$ is also the rotation corresponding to ${}^I_L\boldsymbol{q}$.

We employ B-Spline to parameterize trajectory as it provides closed-form analytic derivatives, which enables to simplify the fusion of high-frequency measurements for state estimation. B-Spline also has a good property of being locally controllable, which means the update of a single control point only impacts several consecutive segments of the spline. This trait yields a sparse system with a limited number of control points. There are several ways to represent a rigid motion trajectory by B-Spline. Some [21, 22] use only one spline to parameterize poses on $SE(3)$ and the others [23, 24] use a split representation of two splines in $\mathbb{R}^3$ and $SO(3)$, respectively. As concluded in [24], the joint form might be torque-minimal but it is hard to control the shape of the translation curve since the translation part is tightly coupled with the rotation. Furthermore, the coupled orientations and translations are difficult to solve in our calibration problem. Consequently, we choose the split representation to parameterize the trajectory.

The Cox-de Boor recursion formula [25] defines the basis functions of B-Spline, and a convenient way to represent splines is to use the matrix formulation [26, 27]. When the knots of B-Spline are uniformly distributed, B-spline is defined fully by its degree. Specifically, for a uniform B-spline of d degree, the translation $\boldsymbol{p}(t)$ over time $t \in [t_i, t_{i+1})$ is controlled only by knots at t_i, t_{i+1}, …, t_{i+d} with their corresponding control points $\boldsymbol{p}_i$, $\boldsymbol{p}_{i+1}$, …, $\boldsymbol{p}_{i+d}$, and the matrix formulation of uniform B-spline could be expressed as follows:

$$\boldsymbol{p}(t) = \sum_{j=0}^{d} \boldsymbol{u}^\top \boldsymbol{M}^{(d+1)}_{(j)} \boldsymbol{p}_{i+j}, \tag{1}$$

where $\boldsymbol{u}^\top = \begin{bmatrix} 1 & u & \dots & u^d \end{bmatrix}$ and $u = (t - t_i)/(t_{i+1} - t_i)$; $\boldsymbol{M}^{(d+1)}_{(j)}$ is the j-th column of the spline $\boldsymbol{M}^{(d+1)}$ which only depends on the corresponding degree of uniform B-Spline. In this paper, cubic ($d = 3$) B-Spline is employed, thus:

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

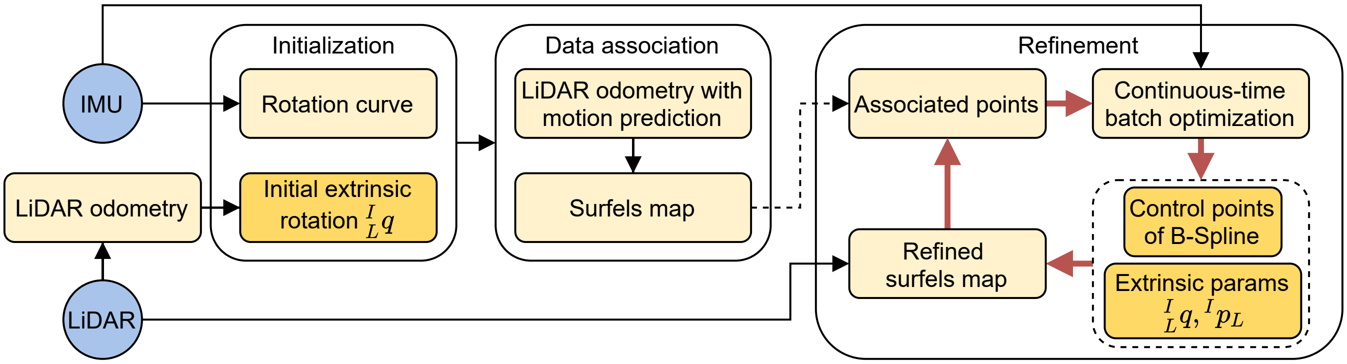

Initialization
Data association
Refinement
IMU
Rotation curve
LiDAR odometry with motion prediction
Associated points
Continuous-time batch optimization
LiDAR odometry
Initial extrinsic rotation ${}^I_L q$
Surfels map
Control points of B-Spline
Extrinsic params ${}^I_L q, {}^I p_L$
Refined surfels map
LiDAR

Fig. 2: The pipeline of proposed LiDAR-IMU calibration method, which allows to leverage all the raw measurements from IMU and LiDAR sensor in a continuous-time batch optimization framework.

Furthermore, Eq. (1) is equal to a cumulative representation:

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

where the corresponding cumulative spline matrix is as follows

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

The cumulative representation of B-spline is also widely used to parameterize orientation on $SO(3)$ [23, 24, 27]. Kim et al. [28] first propose to use unit quaternions as the control points:

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

where $\otimes$ denotes quaternion multiplication, and $\tilde{\boldsymbol{M}}^{(d+1)}_{(j)}$ is the j-th column of the cumulative spline matrix. $\boldsymbol{q}_i$ is an orientation control point with its corresponding knot t_i, and $exp\left(\cdot\right)$ is the operation that mapping Lie algebra elements to S^3, while $log\left(\cdot\right)$ is its inverse operation [28].

In this calibration system, we parameterize the continuous 6-DOF poses of IMU by splines formulated by Eq.(1) and Eq.(5). The derivatives of the splines with respect to time can be easily computed [27], which results in linear accelerations ${}^I\boldsymbol{a}(t)$ and angular velocities ${}^I\boldsymbol{\omega}(t)$ in the local IMU reference frame:

$${}^I\boldsymbol{a}(t) = {}^{I_0}_I\boldsymbol{R}^\top(t)\left({}^{I_0}_I\ddot{\boldsymbol{p}}(t) - {}^{I_0}\boldsymbol{g}\right) \tag{6}$$

$${}^I\boldsymbol{\omega}(t) = {}^{I_0}_I\boldsymbol{R}^\top(t){}^{I_0}_I\dot{\boldsymbol{R}}(t). \tag{7}$$

In the above equation, ${}^{I_0}_I\boldsymbol{R}(t)$ represents the rotation matrix corresponding to unit quaternion ${}^{I_0}_I\boldsymbol{q}(t)$. As described in the above equations, we set the first IMU frame $\{I_0\}$ as the reference frame for the IMU trajectory, and ${}^{I_0}\boldsymbol{g}$ is the gravity in the $\{I_0\}$ frame.

IV. METHODOLOGY

The pipeline of LI-Calib is illustrated in Fig. 2. Firstly, the extrinsic rotation ${}^I_L\boldsymbol{q}$ is initialized by aligning the IMU rotations with LiDAR rotations (Sec. IV-A), where the rotations of LiDAR is obtained from NDT registration [29] based LiDAR odometry. After initialization, we are able to partially remove the motion distortion in LiDAR scans and get better LiDAR pose estimations from LiDAR odometry. LiDAR surfels map is firstly initialized with the LiDAR poses (Sec. IV-B), so are the point-to-surfel correspondences. Subsequently, continuous-time based batch optimization are conducted with the LiDAR and IMU measurements for estimating the states including extrinsics (Sec. IV-C). Finally, with the current best estimated state from optimization, we update the surfels map, point-to-plane data association and optimize the estimated states iteratively (Sec. IV-D).

A. Initialization of Extrinsic Rotation

Inspired by [30], we initialize the extrinsic rotation by aligning two rotation sequences from the LiDAR and the IMU. Given the raw angular velocity measurements $\{{}^{I_0}\boldsymbol{\omega}_m, \cdots, {}^{I_M}\boldsymbol{\omega}_m\}$ from the IMU sensor, we are able to fit the rotational B-Splines ${}^{I_0}_I\boldsymbol{q}(t)$ in the format of Eq.(5). A series of quaternion control points of the fitted B-splines can be computed by solving the following least-square problem:

$$\boldsymbol{q}_0, \cdots, \boldsymbol{q}_N = \arg\min \sum_{k=0}^{M} \left\| {}^{I_k}\boldsymbol{\omega}_m - {}^{I_0}_I\boldsymbol{R}^\top(t_k){}^{I_0}_I\dot{\boldsymbol{R}}(t_k) \right\|, \tag{8}$$

where ${}^{I_k}\boldsymbol{\omega}_m$ is the discrete-time raw gyro measurement reported by IMU at time t_k. It is important to note that the ${}^{I_0}_I\boldsymbol{q}(t_0)$ is fixed to an identity quaternion during the optimization of least-square. In Eq 8, we try to fit the rotation B-Splines by the raw IMU measurements, rather than the integrated IMU measurements (relative pose predictions), which are inaccurate and always affected by drifting IMU biases and noises.

By leveraging scan-to-map matching with NDT-based registration, we can estimate the pose of each LiDAR scan. Thus, it is easy to get the relative rotation between two consecutive LiDAR scans, ${}^{L_k}_{L_{k+1}}\boldsymbol{q}$. Besides, the relative rotation between time interval $[t_k, t_{k+1}]$ in the IMU frame can also be obtained from the fitted B-Splines as ${}^{I_k}_{I_{k+1}}\boldsymbol{q} = {}^{I_0}_{I}\boldsymbol{q}^{-1}(t_k) \otimes {}^{I_0}_{I}\boldsymbol{q}(t_{k+1})$. The relative rotations at any k from two sensors should satisfy the following equation:

$$ {}^{I_k}_{I_{k+1}}\boldsymbol{q} \otimes {}^{I}_{L}\boldsymbol{q} = {}^{I}_{L}\boldsymbol{q} \otimes {}^{L_k}_{L_{k+1}}\boldsymbol{q}. \tag{9} $$

By stacking multiple measurements at different time, we get the following overdetermined equation

$$ \begin{bmatrix} \vdots \\ \alpha_k \left(\left[{}^{I_k}_{I_{k+1}}\boldsymbol{q}\right]_L - \left[{}^{L_k}_{L_{k+1}}\boldsymbol{q}\right]_R \right) \\ \vdots \end{bmatrix} {}^{I}_{L}\boldsymbol{q} = \boldsymbol{Q}_N {}^{I}_{L}\boldsymbol{q} = \boldsymbol{0}, \tag{10} $$

where $[\boldsymbol{q}]_L$ and $[\boldsymbol{q}]_R$ are the left and right quaternion-product matrices [31], respectively; α_k is the weight for each rotations pair, which is determined in a heuristic way for repressing outliers:

$$ r_k = \left\| 2 \left(\arccos({}^{I_k}_{I_{k+1}}q_w) - \arccos({}^{L_k}_{L_{k+1}}q_w) \right) \right\| \tag{11} $$

$$ \alpha_k = \begin{cases} 1, & r_k < \text{threshold} \\ \frac{\text{threshold}}{r_k}, & \text{otherwise} \end{cases} \tag{12} $$

where q_w is the real part of a quaternion. The solution of Eq. (10) can be found as the right unit singular vector corresponding to the smallest singular value of $\boldsymbol{Q}_N$.

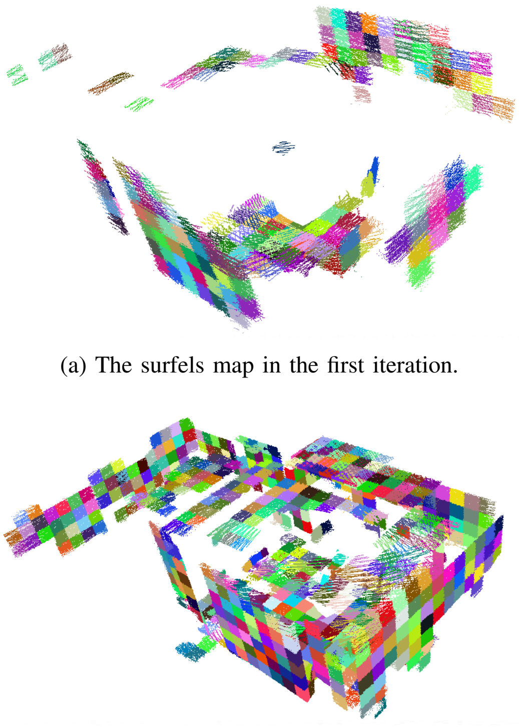
For the initialization of ${}^{I}\boldsymbol{p}_L$, it becomes difficult. Firstly, it should be noted that the acceleration is coupled with gravity and related to the orientation. Thus aligned error in the rotational B-Splines will affect the accuracy of translation B-Splines. Additionally, the B-Spline's quadratic derivative introduces two zero elements in the $\boldsymbol{u}$ vector of Eq. (1) that makes the control points unsolvable. Consequently, initializing the translation curve with raw IMU measurements is unreliable, thus we ignore the initialization of extrinsic translation, which has little impact on the calibration result as the experiments suggest.

B. Surfels Map and Data Association

With the initialized ${}^{I}_{L}\hat{\boldsymbol{q}}$ and orientation curve, the IMU sensor can provide rotation predictions for the LiDAR scan registration and remove rotational distortion in raw scans, which improves the accuracy of LiADR odometry and the quality of LiDAR point cloud map. Subsequently, we discretize the LiDAR point cloud map into 3D cells and calculate the first and second order moments of the points insides each cell. Surfels can be identified based on the plane-likeness coefficient [32]:

$$ \mathcal{P} = 2\frac{\lambda_1 - \lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}, \tag{13} $$

where $\lambda_0 \leq \lambda_1 \leq \lambda_2$, and they are eigenvalues of the second-order moment matrix. For the cell with planar distribution, $\mathcal{P}$ will be close to 1. Therefore, if the plane-likeness $\mathcal{P}$ is larger than a threshold, we will fit a plane π in this cell by using the RANSAC algorithm. The plane π is denoted by its normal vector and the distance from origin $\pi = \left[\boldsymbol{n}^{\top}, d\right]^{\top}$.



(a) The surfels map in the first iteration.

(b) The surfels map in the second iteration.

Fig. 3: 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.

It should be noted that both the planes and the LiDAR map are denoted in frame L_0. As described above, we extract tiny planes from small cells rather than big planes in the environments, which allows us to make full use of all planar areas in the environment and provide more reliable constraints for batch optimization. For robustness, we reject the point-to-plane correspondences when the distance is over a certain threshold. To limit the computational cost, we also downsample the raw LiDAR scans by random sampling.

C. Continuous-time Batch Optimization

The estimated state in the system can be summarized as:

$$ \boldsymbol{x} = [{}^{I}_{L}\boldsymbol{q}^{\top}, {}^{I}\boldsymbol{p}_L^{\top}, \boldsymbol{x}_q^{\top}, \boldsymbol{x}_p^{\top}, \boldsymbol{b}_a^{\top}, \boldsymbol{b}_a^{\top}]^{\top}, \tag{14} $$

where $\boldsymbol{x}_q = [\boldsymbol{q}_0^{\top}, \cdots, \boldsymbol{q}_N^{\top}]^{\top}$ and $\boldsymbol{x}_p = [\boldsymbol{p}_0^{\top}, \cdots, \boldsymbol{p}_N^{\top}]^{\top}$ denote all the control points for the rotation and translation of continuous-time trajectory described in Sec. III. ${}^{I_0}_{G}\bar{\boldsymbol{q}}$ is a two-dimension rotation matrix that aligns the z axis of IMU reference frame I_0 to the gravity. The bias of the accelerometer and gyroscope, $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$, are assumed to be affected by white noises. In general, the calibration problem can be formulated as a graph-based optimization problem. Given a sequence of associated LiDAR points $\mathcal{L}$, linear acceleration measurements $\mathcal{A}$, angular velocity measurements $\mathcal{W}$, and the assumption of independent Gaussian noise corruption on all the above measurements, the maximum likelihood estimation problem $p\left(\boldsymbol{x}|\mathcal{L},\mathcal{A},\mathcal{W}\right)$ can be approximated by solving the following least-square problem:

$$\hat{\boldsymbol{x}}=\arg\min\left\{\sum_{k\in\mathcal{A}}\left\|\boldsymbol{r}_a^k\right\|^2_{\boldsymbol{\Sigma}_a}+\sum_{k\in\mathcal{W}}\left\|\boldsymbol{r}_\omega^k\right\|^2_{\boldsymbol{\Sigma}_\omega}+\sum_{j\in\mathcal{L}}\left\|\boldsymbol{r}_\mathcal{L}^j\right\|^2_{\boldsymbol{\Sigma}_\mathcal{L}}\right\}, \tag{15}$$

where $\boldsymbol{r}_a^k$, $\boldsymbol{r}_\omega^k$, $\mathbf{r}_\mathcal{L}^j$ are the residual errors associated to the accelerometer, gyroscope and LiDAR measurements, respectively. $\boldsymbol{\Sigma}_a$, $\boldsymbol{\Sigma}_\omega$, $\boldsymbol{\Sigma}_\mathcal{L}$ are the corresponding covariance matrices. Particularly, the IMU residuals are defined as:

$$\boldsymbol{r}_a^k={}^{I_k}\boldsymbol{a}_m-{}^{I}\boldsymbol{a}(t_k)-\boldsymbol{b}_a \tag{16}$$

$$\boldsymbol{r}_\omega^k={}^{I_k}\boldsymbol{\omega}_m-{}^{I}\boldsymbol{\omega}(t_k)-\boldsymbol{b}_g, \tag{17}$$

where ${}^{I_k}\boldsymbol{a}_m$ and ${}^{I_k}\boldsymbol{\omega}_m$ are the raw IMU measurements at time t_k. For a associated LiDAR point ${}^{L_j}\boldsymbol{p}_i\in\mathcal{L}$, captured at time t_j and associated with plane π_i, the point-to-plane distance can be calculated as follows:

$$\begin{aligned}
{}^{L_0}\boldsymbol{p}_i &= {}^{I}_{L}\boldsymbol{R}^{\top}\,{}^{I_0}_{I_j}\boldsymbol{R}^{I}_{L}\boldsymbol{R}^{L_j}\boldsymbol{p}_i+{}^{L_0}\boldsymbol{p}_{L_j}\\
{}^{L_0}\boldsymbol{p}_{L_j} &= {}^{I}_{L}\boldsymbol{R}^{\top}\,{}^{I_0}_{I_j}\boldsymbol{R}^{I}\boldsymbol{p}_L+{}^{I}_{L}\boldsymbol{R}^{\top}\,{}^{I_0}\boldsymbol{p}_{I_j}-{}^{I}_{L}\boldsymbol{R}^{\top}\,{}^{I}\boldsymbol{p}_L\\
\mathbf{r}_\mathcal{L}^j &= \left[{}^{L_0}\boldsymbol{p}_i^{\top}\quad 1\right]\pi_j.
\end{aligned} \tag{18}$$

We adopt Levenberg-Marquardt method to minimize Eq.(15) and solve the estimated states iteratively.

D. Refinement

After the continuous-time batch optimization, the estimate states including the extrinsics become more accurate. Thus, we leverage the current best estimates $\hat{\boldsymbol{x}}$ 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, NDT registration based LiDAR odometry is only utilized at the very beginning (first iteration in batch optimization) for initializing the LiDAR poses and LiDAR map . The typical LiDAR surfels maps in the first and second iteration of the batch optimization are shown in Fig. 3. The quality of the map can be improved in a significant margin after just one iteration of batch optimization. With only few iterations (around 4 iterations in our experiments), the proposed method converges and is able to give calibration results with high accuracy.

V. EXPERIMENTS

To validate the proposed method, we conduct extensive experiments on both simulated and real-world datasets. We begin the evaluations with simulation studies, in which we try to analyze the calibration accuracy achieved by LI-Calib. As for the real-world experiments, we try to calibrate the self-assembled sensors with a LiDAR VLP-16[1] and three Xsens IMUs [2], as shown in Fig. 1. In the experiments, the

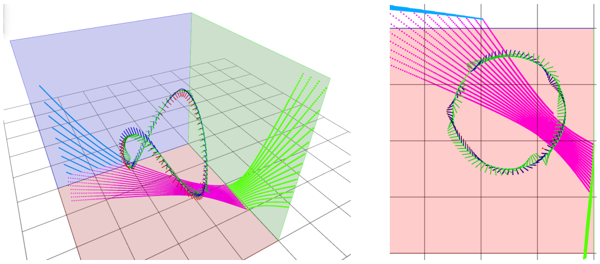


Fig. 4: In the simulation, three planes are orthogonal to each other. The IMU trajectory is denoted by sequential coordinate markers, and the discrete colored points are the LiDAR scans with motion distortion. The size of the background grid is 2m.

LiDAR and IMU are synchronized by hardware [33], and we only focus on the spatial calibration between two sensors. Furthermore, in our implementation, Kontiki [34] toolkit is adopt for the continuous-time batch optimization.

A. Simulation

In the simulation, the simulated sensors' characteristics are consistent with the actual sensors used in the real-world experiments. IMU measurements are reported in 400Hz. LiDAR scans are received in 10Hz with a FOV of 360° horizontally, and ±15° vertically. Fig. 4 shows the simulation environment, where three planes are orthogonal to each other, and the IMU moves in a sinusoidal trajectory. We conduct Monte Carlo experiments by simulating 10 different calibration sequences with the same time span of 10 Sec. Gaussian noises with the same characteristics of real-word sensors are generated and added into the synthetic measurements. Compared to the ground truth, the reported calibration results are with translational errors of 0.0043 ± 0.0006 meters and orientational errors of 0.0224 ± 0.0026 degrees, which demonstrates the high accuracy and effectiveness of the proposed LI-Calib.

B. Real-world Experiments

We collected data in both indoor and outdoor scenarios (see Fig. 8) by the self-assembled sensors in Fig. 1 for the real-world experiments. In each scenario, five sequences with a 25-30 Sec duration are recorded with an average absolute angular velocity of about $47.39°/s$ and an average absolute acceleration of $0.628m/s^2$ overall sequences. Besides, we also collect ten sequences in the Vicon room with a duration of about 15 Sec to evaluate the accuracy of estimated trajectories and the representation capability of the splines. Due to the absence of ground-truth extrinsic transformation between LiDAR and IMU in real-world experiments and the **lack of open-sourced LiDAR-IMU calibration algorithms**, the relative poses of three IMU inferred from CAD assembly drawings are introduced as references, and the repeatability of the calibration results is also examined over different calibration sequences. Furthermore, we also analyze the results computed from different numbers of iterations in the refinement step.

[1] https://velodyneLiDAR.com/vlp-16.html

[2] https://www.xsens.com/products/mti-100-series/

		x(m)	y(m)	z(m)	roll$(^\circ)$	pitch$(^\circ)$	yaw$(^\circ)$
Indoor Scene	IMU1	$-0.0030 \pm \mathbf{0.0020}$	-0.1058 ± 0.0053	-0.1453 ± 0.0014	$180.12 \pm \mathbf{0.18}$	$179.81 \pm \mathbf{0.28}$	$89.15 \pm \mathbf{0.18}$
	IMU2	0.0978 ± 0.0015	-0.0127 ± 0.0035	-0.1437 ± 0.0023	180.05 ± 0.06	179.63 ± 0.21	88.48 ± 0.11
	IMU3	-0.0749 ± 0.0020	$0.0671 \pm \mathbf{0.0062}$	$-0.1456 \pm \mathbf{0.0039}$	-0.35 ± 0.17	0.61 ± 0.20	133.20 ± 0.17
Outdoor Scene	IMU1	-0.0051 ± 0.0034	-0.1027 ± 0.0031	$-0.1623 \pm \mathbf{0.0140}$	179.99 ± 0.07	179.82 ± 0.29	89.27 ± 0.12
	IMU2	$0.0978 \pm \mathbf{0.0052}$	-0.0105 ± 0.0047	-0.1664 ± 0.0114	179.91 ± 0.15	$179.64 \pm \mathbf{0.36}$	88.65 ± 0.10
	IMU3	-0.0753 ± 0.0042	$0.0709 \pm \mathbf{0.0049}$	-0.1681 ± 0.0120	$-0.24 \pm \mathbf{0.29}$	0.56 ± 0.26	$133.35 \pm \mathbf{0.13}$

TABLE I: The mean and SD(standard deviation) of the calibration results for five different trails in two scenarios.

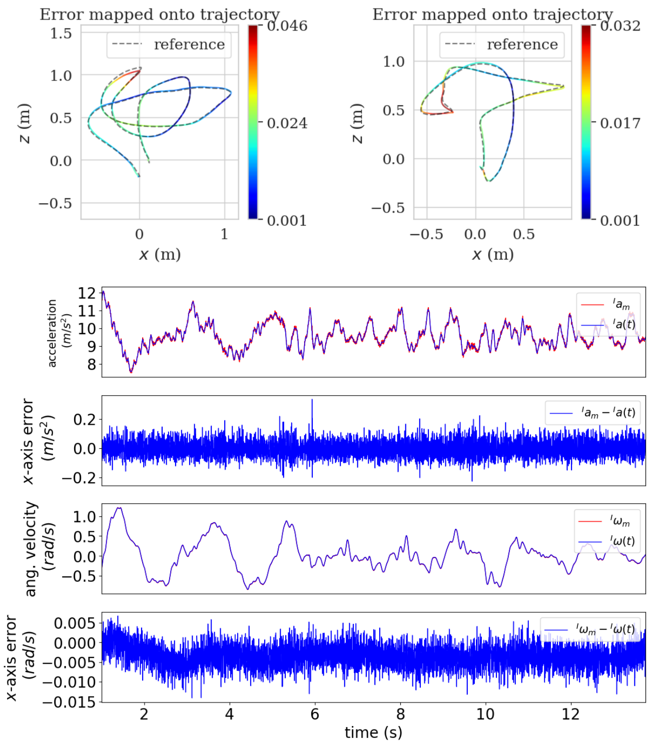


Fig. 5: Top: Estimated continuous-time trajectories aligned with the ground-truth. The color of the trajectory indicates ATE. Bottom: The corresponding fitting results of the top-left trajectory, according to Eq. (16) and Eq. (17). Only the x-axis is plotted. The fitting errors almost have the same order of magnitude as the IMU measurement noise.

In practice, the cell's resolution is set at 0.5 m for indoor cases and 1.0 m for outdoor cases considering the scale of the environment. Additionally, in the first iteration, the cells with the plane-likeness coefficient $\mathcal{P}$ over 0.6 are accepted for data association. After the first batch optimization, motion distortion in LiDAR points will be compensated and a refined surfels map is available, we further improve the threshold of $\mathcal{P}$ to 0.7. Except for the resolution of the cell and plane-likeness coefficient are different, other parameters remain unchanged in the experiment. Particularly, to render all quantities of the calibration observable and achieve more accurate calibration results, we ensure sufficient linear acceleration and rotational velocity over all datasets, and the knot spacing of the spline is set as 0.02 Sec to deal with the highly dynamic motion.

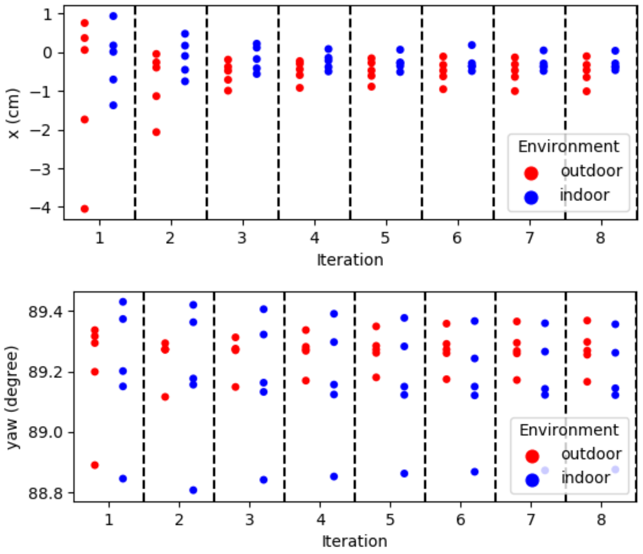


Fig. 6: As the number of iterations increases, both translation and rotation are gradually converging, after about four iterations, the results gradually stabilized and remained unchanged. Comparing the difference between the indoor and outdoor calibration results, we can find that better calibration results can be obtained in indoor environments with more flat surfaces.

Comparing the estimated trajectories with the motion-capture system's reported trajectories, the average absolute trajectory error(ATE) [35] over all ten sequences are 0.0183 m. Fig.5 shows two typical estimated trajectory aligned with the ground truth and also illustrates fitting results according to Eq. (16) and Eq. (17). The estimated B-Spline trajectory fits well with the ground-truth trajectory and the acceleration measurements and the angular velocity measurements, which indicate the high representation capability of the B-Spline.

We also evaluate the repeatability of the proposed method on the calibration sequences collected from two typical scenarios. Table I shows the statistics over the transformation from IMU to LiDAR after eight iterations; the extrinsic rotations have been converted to Euler angle. It can be seen that the final differences are within a few millimeters and milliradians in the indoor environment. Due to the fact that the outdoor environment is more complicated and most planes are uneven, the repeatability of the proposed method over extrinsic translation outdoors is not as good as indoors. In general, the proposed method has a good repeatability

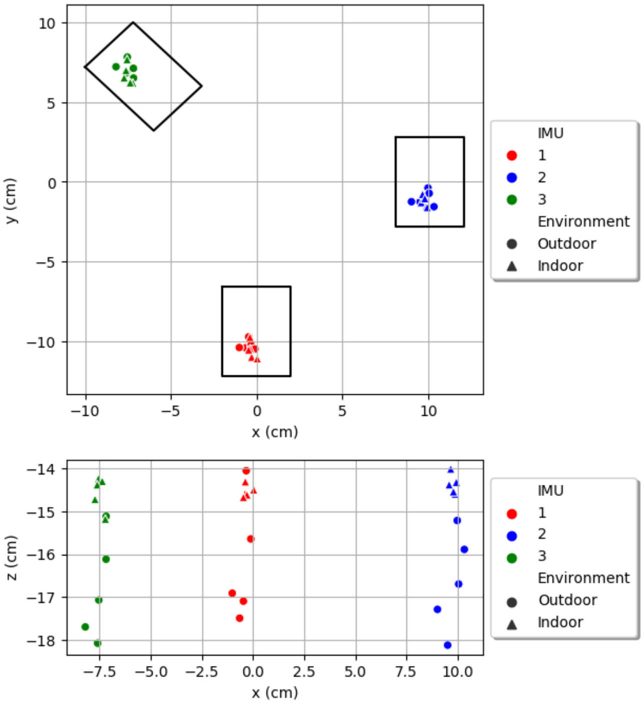


Fig. 7: The calibration results of the sensor setup shown in Fig. 1 after the 8th iteration. Top: Top view of the calibration results. The three rectangular boxes represent the actual installation positions of the three IMUs, respectively. Bottom: Front view of the calibration results. The calibration results over the indoor data are generally more reliable than the results over the outdoor data.

	Estimated(m)	CAD Reference(m)	Difference(m)
IMU2 X	-0.0947 ± 0.0035	-0.0935	**0.0012**
IMU2 Y	0.0993 ± 0.0015	0.1010	**0.0017**
IMU2 Z	0.0011 ± 0.0023	0	**0.0012**
IMU3 Z	-0.0007 ± 0.0011	0	**0.0007**
	Estimated$(^\circ)$	CAD Reference$(^\circ)$	Difference$(^\circ)$
IMU2 Roll	179.93 ± 0.06	180	**0.07**
IMU2 Pitch	179.83 ± 0.21	180	**0.17**
IMU2 Yaw	179.33 ± 0.11	180	**0.67**
IMU3 Yaw	44.05 ± 0.17	45	**0.95**

TABLE II: The relative rotation and translation of IMU2 (or IMU3) w.r.t. IMU1 compared with CAD reference over the indoor datasets.

across different sequences, especially for the calibration of rotation between sensors.

It is also interesting to study the effect of the number of iterations on the calibration results. As shown in the Fig 6, the improvement between the first and second iterations is very significant. We think it origins from the poor initial surfels map and unreliable point-to-surfel data correspondences. After motion compensation, the surfels map and data association are more reliable. After about four iterations, the results gradually converge to the final values and remain unchanged.

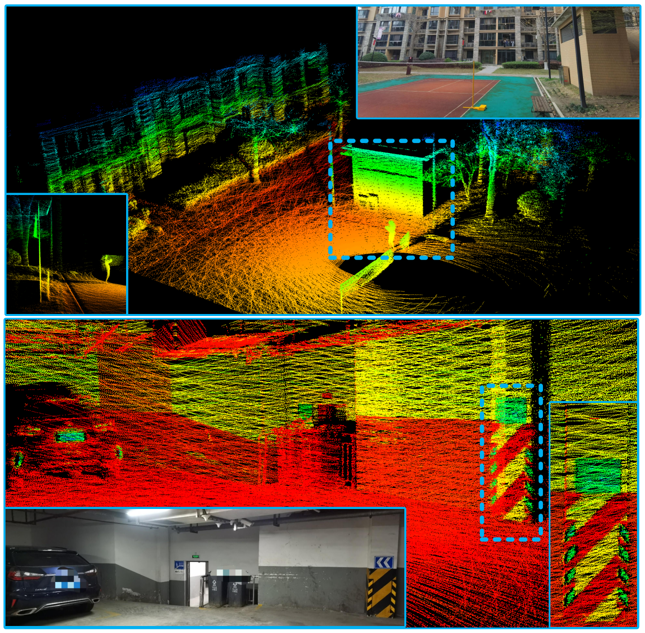


Fig. 8: The two different scenarios in real-world experiments. The point cloud maps consist of all undistorted LiDAR points by the calibration results. The top one is colored with height, and the bottom is with reflective intensity for clarity.

Fig. 7 shows the final refined calibration results, the transformation from the IMU to the LiDAR, and CAD sketch for three IMUs. Table. II details the relative poses results, where we set IMU1 as the origin and calculate the relative rotation and translation from IMU2 (or IMU3) to IMU1. The relative poses calculated from the calibration and the CAD are compared. The difference in relative translation is less than a few millimeters, and that in the rotation angle is less than 1°.

VI. Conclusions and Future Work

In this paper, we propose a novel targetless LiDAR-IMU calibration method termed LI-Calib, based on continuous-time batch estimation. The real-world experiments indicate that the proposed method is highly accurate and repeatable, especially in the human-made structured environment. A current limitation of LI-Calib is the dependence on the NDT registration based LiDAR odometry in the first iteration. If the initial LiDAR odometry is poor, LI-Calib fails to get enough reliable point-to-surfel correspondences, which may lead to an inaccurate calibration. Consequently, it might be worth investigating some different front-ends for LiDAR odometry in the future.

VII. Acknowledgement

This work is supported by the National Key R&D program of China under Grant 2018YFB1305900 and the National Natural Science Foundation of China under Grant 61836015. We owe thanks to Yusu Pan for the help of visualizing calibration results.

References

[1] J. Zhang and S. Singh. “Visual-lidar odometry and mapping: Low-drift, robust, and fast”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2015, pp. 2174–2181.

[2] D. Maturana and S. Scherer. “Voxnet: A 3d convolutional neural network for real-time object recognition”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2015, pp. 922–928.

[3] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang. “LIC-Fusion: LiDAR-Inertial-Camera Odometry”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. Macau, China, 2019.

[4] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. “RangeNet++: Fast and Accurate LiDAR Semantic Segmentation”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2019.

[5] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang. “Visual-Inertial Localization with Prior LiDAR Map Constraints”. In: *IEEE Robotics and Automation Letters* (2019).

[6] S. Ceriani, C. Sánchez, P. Taddei, E. Wolfart, and V. Sequeira. “Pose interpolation slam for large maps using moving 3d sensors”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2015, pp. 750–757.

[7] C. Le Gentil, T. Vidal-Calleja, and S. Huang. “3D Lidar-IMU Calibration based on Upsampled Preintegrated Measurements for Motion Distortion Correction”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2018, pp. 2149–2155.

[8] 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 International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 459–465.

[9] J. Lyu, J. Xu, X. Zuo, and Y. Liu. “An Efficient LiDAR-IMU Calibration Method Based on Continuous-Time Trajectory”. In: *IROS 2019 Workshop on Visual-Inertial Navigation: Challenges and Applications*. Macau, China, Nov. 2019.

[10] H. Ye, Y. Chen, and M. Liu. “Tightly coupled 3d lidar inertial odometry and mapping”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2019, pp. 3144–3150.

[11] F. Neuhaus, T. Koß, R. Kohnen, and D. Paulus. “Mc2slam: Real-time inertial lidar odometry using two-scan motion compensation”. In: *German Conference on Pattern Recognition*. Springer. 2018, pp. 60–72.

[12] A. Nüchter, M Bleier, J. Schauer, and P. Janotta. “Improving Google’s Cartographer 3D mapping by continuous-time slam”. In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2017), p. 543.

[13] D. Droeschel and S. Behnke. “Efficient continuous-time SLAM for 3D lidar-based online mapping”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2018, pp. 1–9.

[14] F. Pomerleau, F. Colas, R. Siegwart, et al. “A review of point cloud registration algorithms for mobile robotics”. In: *Foundations and Trends® in Robotics* 4.1 (2015), pp. 1–104.

[15] J.-E. Deschaud. “IMLS-SLAM: scan-to-model matching based on 3D data”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2018, pp. 2480–2485.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets robotics: The KITTI dataset”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[17] R. Horaud and F. Dornaika. “Hand-eye calibration”. In: *The International Journal of Robotics Research* 14.3 (1995), pp. 195–210.

[18] P. Furgale, T. D. Barfoot, and G. Sibley. “Continuous-time batch estimation using temporal basis functions”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2012, pp. 2088–2095.

[19] P. Furgale, J. Rehder, and R. Siegwart. “Unified temporal and spatial calibration for multi-sensor systems”. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 1280–1286.

[20] 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. International Conference on Robotics and Automation*. IEEE. 2016, pp. 4304–4311.

[21] S. Lovegrove, A. Patron-Perez, and G. Sibley. “Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras.” In: *BMVC*. Vol. 2. 5. 2013, p. 8.

[22] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza. “Continuous-time visual-inertial odometry for event cameras”. In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1425–1440.

[23] H. Ovrén and P.-E. Forssén. “Trajectory representation and landmark projection for continuous-time structure from motion”. In: *The International Journal of Robotics Research* 38.6 (2019), pp. 686–701.

[24] 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: *2018 International Conference on 3D Vision (3DV)*. IEEE. 2018, pp. 381–389.

[25] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor. *A practical guide to splines*. Vol. 27. springer-verlag New York, 1978.

[26] K. Qin. “General matrix representations for B-splines”. In: *The Visual Computer* 16.3 (2000), pp. 177–186.

[27] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers. “Efficient Derivative Computation for Cumulative B-Splines on Lie Groups”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11148–11156.

[28] M.-J. Kim, M.-S. Kim, and S. Y. Shin. “A general construction scheme for unit quaternion curves with simple high order derivatives”. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 1995, pp. 369–376.

[29] M. Magnusson. “The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection”. PhD thesis. Örebro universitet, 2009.

[30] Z. Yang and S. Shen. “Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration”. In: *IEEE Transactions on Automation Science and Engineering* 14.1 (2016), pp. 39–51.

[31] J. Sola. “Quaternion kinematics for the error-state Kalman filter”. In: *arXiv preprint arXiv:1711.02508* (2017).

[32] M. Bosse and R. Zlot. “Continuous 3D scan-matching with a spinning 2D laser”. In: *Proc. International Conference on Robotics and Automation*. IEEE. 2009, pp. 4312–4319.

[33] J. Xu, J. Lv, Z. Pan, Y. Liu, and Y. Chen. “Real-Time LiDAR Data Assocation Aided by IMU in High Dynamic Environment”. In: *IEEE International Conference on Real-time Computing and Robotics* (2018), pp. 202–205.

[34] H. Ovrén. *Kontiki - the continuous time toolkit*. `https://github.com/hovren/kontiki`. 2018.

[35] M. Grupp. *evo: Python package for the evaluation of odometry and SLAM*. `https://github.com/MichaelGrupp/evo`. 2017.