IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 6, 2021, accepted September 11, 2021, date of publication September 22, 2021, date of current version October 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3114618

3D LiDAR/IMU Calibration Based on Continuous-Time Trajectory Estimation in Structured Environments

SHUAIXIN LI[1], LI WANG[1], JIUREN LI[2], BIN TIAN[3], LONG CHEN[4], (Senior Member, IEEE), AND GUANGYUN LI[1]

[1]Department of Geospatial Information, PLA Information Engineering University, Zhengzhou 450001, China
[2]Waytous Infinity Inc., Beijing 100089, China
[3]State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 1000190, China
[4]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China

Corresponding author: Guangyun Li (guangyun_li@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 42071454, and in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0909050001 and Grant 2020B090921003.

ABSTRACT Sensor calibration is a fundamental step for improving the performance in sensor fusion, the aim of which is to spatially and temporally register sensors with respect to each other. This paper presents a high-accuracy autocalibration method to estimate extrinsic parameters between LiDAR and an IMU. LiDAR/IMU calibration is a challenging task since the raw measurements are distorted, biased, noisy, and asynchronous. Our calibration approach adopts continuous-time trajectory estimation wherein the IMU trajectory is modeled by Gaussian process(GP) regression with respect to the independent sampling timestamps. Accordingly, the distorted and delayed LiDAR points sampled at discrete timestamps can be analytically modeled in on-manifold batch optimization. To efficiently and accurately associate laser points with stable environmental objects, the method is carried out in known environments with a point map that is segmented as structured planes and managed by a specially designed octree map. We thoroughly investigated factors relevant to the calibration accuracy and evaluated the performance of the proposed method using both simulated and real-world datasets. The results demonstrate that the accuracy and robustness of our calibration approach are sufficient for most applications.

INDEX TERMS LiDAR/IMU calibration, SLAM, continuous-time batch optimization, octree map.

I. INTRODUCTION

Accurate and reliable 6-degree-of-freedom (DOF) motion estimation plays a major role in the fields of robotics, autonomous driving, unmanned aerial vehicle (UAVs), and mobile mapping [1], [2]. Since LiDAR has many excellent characteristics compared with cameras, e.g., invariant to illumination and a horizontal field of view (FOV) of 360 degrees, it has become a preferred choice for self-localization and mapping systems, which are the so-called eyes of robots.

LiDAR, however, is also defective due to its limited vertical FOV and the discreteness of scanned points. The LiDAR-only ego-motion estimation in unknown environments is accordingly intractable, especially in the case where LiDAR is in highly dynamic motion. An inertial measurement unit (IMU) is an ideal complementary sensor to fuse with visual sensors, e.g., LiDAR and cameras, since it can provide accurate short-term motion estimates with no requirement for surroundings [3]–[10]. Moreover, high-frequency relative transforms exported from IMUs extensively contribute to recovering the distortion of scans caused by high dynamics, and enable the provision of more reliable pose initials for point registration. To fuse multimodal sensor data, the explicit spatial and temporal offsets between sensors must be calibrated. Despite the current popularity of coupling LiDAR with IMUs in the field of robotics, few works have focused on the calibration of them.

In this paper, we present a high-accuracy autocalibration method for LiDAR/IMU rigd that makes use of the high-frequency output of IMUs. The proposed method has no assumptions about the sensor configurations or initial parameter initial estimates. Inspired by GP regression-based

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Mei.

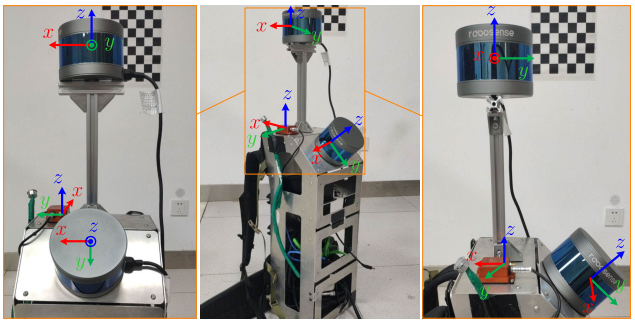


FIGURE 1. The self-assembled LiDAR/IMU backpack. Two Robosense LiDARs, and one Xsens IMU are on-board. The proposed method tries to calibrate the temporal and spatial offsets between the IMU and LiDARs. Back and lateral views of the backpack are displayed in the left and right subfigures, respectively.

simultaneous trajectory estimation and mapping (STEAM) proposed in [11], we formulate the IMU poses as a continuous-time trajectory using the probabilistic, nonparametric GP regression model, which enables the distorted and delayed laser point association equations and their Jacobians to be evaluated analytically in the on-manifold optimization. To guarantee the validity of point-to-map association and to increase the accuracy of the calibration, the method is carried out in known structured environments. We segment planar objects and propose an adaptive voxel map to manage them. The main contributions of our work are three-fold:

- The autonomous structured map construction approach and the efficient map point management approach are presented, which facilitate LiDAR odometry (LO) to efficiently and accurately associate scanned points with the planar map. As a result, accurate and robust LiDAR odometry can be achieved in known environments.
- A novel formulation of the LiDAR/IMU temporal and spatial calibration problem is modeled based on the continuous-time trajectory. The residuals induced by IMU raw measurements and distances of the biased, time-shifted LiDAR points to associated planes are explicitly modeled in the on-manifold batch optimization rendering the calibration problem solvable.
- Both simulated and real-world experiments are conducted to evaluate the performance of the proposed method. In addition, the influences of some relevant factors, e.g., the prior map accuracy and the optimization state frequency, are investigated, and empirically optimal choices are researched.

The remainder of the paper is organized as follows: Sect.II briefly reviews related works. In Sect.III, we first introduce the notation, task, and assumption of the paper and summarize the proposed calibration approach. We detail the processing procedure of our calibration method in Sect.IV. Sect.V provides some implementation discussions and shows experimental results. Finally, the conclusion is drawn in Sect.VI.

II. RELATED WORK

The calibration of extrinsic parameters between LiDAR and inertial navigation systems (INSs) has long been studied in the field of navigation. Early works relied on additional specially designed targets and precise control devices. Liu *et al.* employed a precise rotating platform to drive an INS with precise control and utilized cone/cylinder targets with known world coordinates to estimate global LiDAR poses [12], [13]. The calibration problem was solved by restricted maximum likelihood estimation (MLE). The extensive labor and strict experimental requirements involved substantially limit the repeatability and usability of this calibration method, rendering it inapplicable in the field of robotics.

In terms of autocalibration, the most straightforward idea is aligning the trajectory of LiDAR with that of an IMU and estimating the extrinsic parameters with the closed-form solution [14], [15]. Although this typical calibration approach has been widely applied to multimodal sensor calibration problems [16], [17], it faces some challenges in regard to the calibration of 3D LiDAR and low-cost IMUs. On the one hand, the IMU built-in accelerometer and gyroscope both suffer from drifts, leading to biased IMU trajectories. On the other hand, discrete laser points captured by the moving LiDAR are distorted, especially when the scanner rotates rapidly. In practice, IMUs must follow an accelerated motion to be observable. Such highly dynamic motion, however, decreases the accuracy of LiDAR odometry. To address these issues, [15] adopted the high-accuracy postprocessed trajectory of GPS/INS, and manually aligned consecutive laser sweeps to guarantee the registration accuracy and the LiDAR trajectory accuracy, which is laborious and complicated. Moreover, another issue is that the sampling timestamps of multimodal sensors are asynchronous. Qin *et al.* formulated reprojection residuals as variables with respect to the timeshift between LiDAR and an IMU. By estimating the velocity of the extracted visual feature on an image, the real pixel position delayed by the timeshift can be modeled [18].

Modeling discrete poses as a continuous-time trajectory is an ingenious idea to solve the above issues, and the exact reference poses of each laser point can be queried with scanning timestamps. This method has been widely employed in many autocalibration approaches, such as those in [19]–[23]. Furgale and Rehder *et al.* presented a series of pioneering works related to temporal-spatial calibration of multiple sensors. In [22], for the calibration of cameras and IMUs, they modeled the time-varying IMU trajectory using a B-spline and utilized checkerboards to specify the spatial corner features for visual odometry. On this basis, they extended their work to the calibration of LiDAR/camera/IMU rigs by adding additive point-to-plane constraints into batch optimization [21]. However, the prerequisite of a sufficiently accurate visual/inertial trajectory makes it unable to calibrate LiDAR/IMU directly. Reference [23] proposed a continuous-time batch optimization-based LiDAR/IMU calibration method, which similarly formulated the trajectory of LiDAR as the B-spline to address the discrete scanning issue. However, the temporal-offset is not considered, and usage of the B-spline function results in more parameters

to be estimated. Moreover, although the targetless approach is convenient, the distortion of points cannot be explicitly modeled based on the incrementally accumulated point map. Gentil *et al.* adopted GP regression to upsample IMU measurements for point-based distortion deskewing and formulated preintegration measurements (PIMs) as functions with respect to the temporal shift [20]. However, the temporal offset should be a constant value that shifts the starting and ending timestamps of PIMs, rather than an addition to the original PIM.

In general, the multimodal sensor calibration problem can be solved by modeling it as an optimization problem. The importance of the accuracy of optimization initials has been thoroughly discussed in many works [24]–[26], which have demonstrated that an accurate optimization result requires a robust initialization procedure to obtain reliable initials. Consequently, the two-step calibration scheme has become the most popular framework for auto-calibration of multi-sensors. Reference [19] applied the coarse-to-fine calibration scheme in which the closed-form solution is first estimated and then the continuous-time trajectory batch optimization is employed to refine the result. Reference [25] presented a novel IMU/camera initialization approach to estimate the extrinsic rotation parameter, initial bias, initial velocity, and gravity, and then fed all these initials to the optimizer to refine the result.

Compared to current state-of-the-art methods, the presented work represents the time-continuous trajectory nonparametrically by a GP with time as the independent variable. The idea is driven by the fact that an IMU exports motion information with high frequency. The method aims to accurately estimate the temporal and spatial offset between LiDAR and the IMU in a batch using biased, noisy, and asynchronous sensor measurements directly without either assumptions about the configuration of the sensors or laborious involvement. To formulate the explicit and precise LiDAR point measurement model, the proposed approach is dependent on the known environment point map, which provides s stable reference for distorted laser points.

III. STATEMENT AND OVERVIEW

A. NOTATION AND TASK

We denote the nth full circle of a laser scan as *sweep* $\mathcal{S}_n$ and define the sampling *timestamp* t_n at the end of $\mathcal{S}_n$. Since the frequency of the IMU is much higher than that of LiDAR, we denote t_{n_s} as the sampling timestamp of the IMU in the period of the nth sweep, i.e., $t_{n_s} \in [t_{n-1}, t_n]$. The group of IMU measurements sampled in $[t_{n-1}, t_n]$ is denoted as $\mathcal{I}_{n-1,n}$. In the following sections, the transformation is expressed by $\mathbf{T}_b^a \in SE(3)$, which transforms the point ${}_b\mathbf{p} \in \mathbb{R}^3$ in the $\mathcal{F}_b$ frame into the $\mathcal{F}_a$ frame. $\mathbf{T}_b^a = \left[\mathbf{R}_b^a, \mathbf{t}_b\right]$ is composed of a rotation matrix $\mathbf{R}_b^a \in SO(3)$ and a translation vector $\mathbf{t}_b \in \mathbb{R}^3$. The *Lie algebra* $\mathfrak{se}(3)$ is used to represent the tangent space to the manifold of $SE(3)$. The *exponential* and *logarithm* are used to map a vectorized pose $\boldsymbol{\xi} = \left[\boldsymbol{\theta}^\top, \boldsymbol{\eta}^\top\right]^\top$ to $\mathbf{T}$, where $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ are vectorized rotation and translation, respectively

$$\begin{aligned} Exp:\ \mathbb{R}^6 \to SE(3); \quad \boldsymbol{\phi} \mapsto \exp(\boldsymbol{\phi}^\wedge), \\ Log:\ SE(3) \to \mathbb{R}^6; \quad \mathbf{T} \mapsto (\log \mathbf{T})^\vee. \end{aligned} \tag{1}$$

The *hat* $(\cdot)^\wedge$ and *vee* operators $(\cdot)^\vee$ build the mapping relationship between a vector in $\mathbb{R}^3$ and a skew-symmetric matrix. $\otimes$ is used for the multiplication of $SE(3)$. In this paper, we use the right-disturbance model to update the transformation $\mathbf{T}_{new} = \mathbf{T}_{old} \otimes \delta\mathbf{T}$.

We define the body frame as $\mathcal{F}_b$ coinciding with the origin of the IMU and denote the LiDAR frame as $\mathcal{F}_l$. The world frame $\mathcal{F}_w$ is the mapping frame of the trajectory $\mathcal{T}$, and the map frame $\mathcal{F}_m$ is the reference frame of the prior map $\mathcal{M}$. The spatial and temporal offsets between LiDAR and the IMU are written as $\mathbf{T}_b^l$ and Δt_b^l. Moreover, the operator $\hat{(\cdot)}$ indicates the measurement, while $\bar{(\cdot)}$ and $\tilde{(\cdot)}$ indicate the prediction and estimation, respectively. Specifically, ${}_{b_s}\hat{\mathbf{a}}$ and ${}_{b_s}\hat{\mathbf{w}}$ represent the raw measurements of the IMU at timestamp $t_s \in [t_{n-1}, t_n]$, while ${}_{l_n}\hat{\mathbf{p}}_i$ represents the ith point in the nth sweep. Note that the left subscript representing the mapping frame will be omitted for readability in the following section.

Problem: Given the prior map $\mathcal{M}$, the asynchronous IMU measurements $[{}_{b_s}\hat{\mathbf{a}}, {}_{b_s}\hat{\mathbf{w}}] \in \mathcal{I}_{n-1,n}$ and LiDAR sweeps ${}_{l_n}\hat{\mathbf{P}} \in \mathcal{S}_n$, the problem addressed in this paper is automatic and accurate estimation of the spatial transformation $\tilde{\mathbf{T}}_b^l$ and the temporal offset $\Delta\tilde{t}_b^l$ between LiDAR and the IMU. Accelerometer and gyroscope biases $\tilde{\mathbf{b}} = [\tilde{\mathbf{b}}^a, \tilde{\mathbf{b}}^g]$, the continuous-time trajectory $\tilde{\mathbf{T}}_b^w(t)$, and the linear and angular velocities ${}_w\tilde{\boldsymbol{\varpi}}(t) = [{}_w\tilde{\mathbf{v}}(t), {}_w\tilde{\mathbf{w}}(t)]$ of the IMU are also estimated in a batch together with extrinsic parameters, all of which constitute the complete state $\mathcal{X}_n$ at t_n

$$\begin{aligned} \mathcal{X}_n = [\mathbf{T}_b^l, \Delta t_b^l, \mathbf{T}_{b_0}^w(t), \cdots, \mathbf{T}_{b_s}^w(t), \\ {}_w\boldsymbol{\varpi}_0(t), \cdots, {}_w\boldsymbol{\varpi}_s(t), \mathbf{b}_0, \cdots, \mathbf{b}_s]. \end{aligned} \tag{2}$$

Note that Eq.(2) is not a strict state vector presentation in which both operators $\log(\cdot)$, $(\cdot)^\vee$ and $(\cdot)^\top$ used for mapping vectorized terms to Lie groups are omitted for readability.

B. OVERVIEW

Fig.2 provides an overview of the proposed calibration approach. We summarize the system as four modules below.

1) STRUCTURED MAP CONSTRUCTION

The point cloud map $\mathcal{M}$ of the environment is loaded first to provide an accurate map reference. The planes $\mathcal{M}^{plane}$ are segmented using a voxel-based region growing algorithm. Next, a specially designed octree containing a cluster of voxels $\mathcal{V}^{plane}$ with different resolutions is constructed to manage structured planes. Moreover, the norm of each voxel is stored to facilitate the point-plane association in subsequent LiDAR odometry.

2) PREPROCESS

Planar features ${}_{l_n}\hat{\mathbf{F}}_i, i \in N$ are extracted from the distorted and temporally shifted LiDAR sweep $\hat{\mathcal{S}}_n$, where $n \in \mathbb{Z}^+$

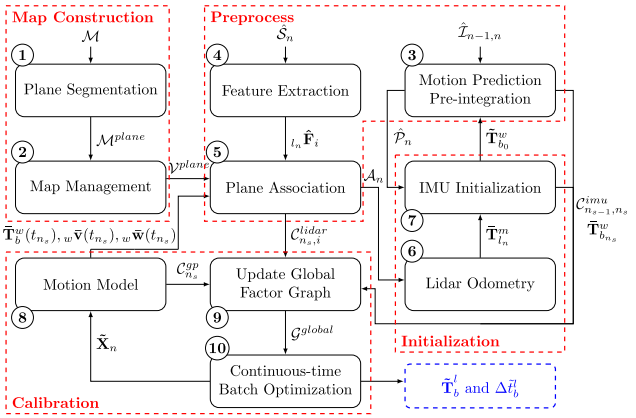


FIGURE 2. The pipeline of the proposed LiDAR/IMU calibration approach enables leveraging the raw measurements from LiDAR and the IMU to construct a well-constrained continuous-time batch optimization backend. The main steps will be introduced in turn.

represents the index of the LiDAR timestamp. Then, feature points are associated with map planes to build point-to-plane associations $\hat{\mathcal{A}}_n$. At the same time, the IMU PIM $\hat{\mathcal{P}}_n$ is calculated using $\hat{\mathcal{I}}_{n-1,n}$. Note that although the pose prediction by IMU dynamics integration is also calculated, it is valid only after the initialization, i.e., the initial pose $\check{\mathbf{T}}^w_{b_0}$ and gravity $\mathbf{g}$ are initialized.

3) INITIALIZATION

The initial estimate of the parameter (i.e., high-frequency IMU states at t_{n_s}, extrinsic parameters, and IMU biases) and observed measurements (i.e., point data associations and PIMs) are estimated for further calibration optimization. Data associations $\hat{\mathcal{A}}_n$ are used to estimate the LiDAR ego-motion $\bar{\mathbf{T}}^m_{l_n}$. PIMs $\hat{\mathcal{P}}_n$ and $\bar{\mathbf{T}}^m_{l_n}$ are stored as pairs, until filling in the buffer to trigger the initialization. In the initialization module, the rotational extrinsic parameters and the initial gravity ${}_w\mathbf{g}$ are estimated using the closed-form solution, which will be used as initials in the further batch optimization. In addition, states of the IMU $\bar{\mathbf{T}}^w_{b_{n_s}}$ in the period of $\hat{\mathcal{S}}_n$ as well as the high-frequency PIMs $\hat{\mathcal{P}}_{n_s}$ relating to consecutive IMU sampling timestamps $t_{n_{s-1}}$ and t_{n_s} can be re-evaluated, where $s \in \mathbb{Z}^+$ represents IMU timestamps.

4) CALIBRATION

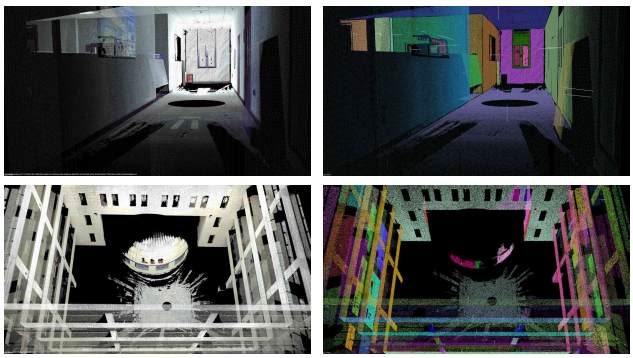
Both the re-evaluated IMU states $\bar{\mathbf{T}}^w_{b_{n_s}}$ and PIMs $\mathcal{C}^{imu}_{n_{s-1},n_s}$ related to the states at $t_{n_{s-1}}$ and t_{n_s} are inserted into the factor graph $\mathcal{G}^{global}$ as initials and observation constraints, respectively to perform the coarse-to-fine calibration estimation. To address the distortion issue caused by LiDAR discrete scanning, we utilized GP regression to model the continuous-time IMU trajectory. A class of exactly sparse GP priors generated by the linear time-varying stochastic differential equations (LTV-SDE) is applied to create a GP prior factor $\mathcal{C}^{gp}_{n_s}$ [11], [27], which links only two nearby states to maintain the sparsity of the factor graph. On the basis of the prior motion model, poses and velocities at anytime $t_\tau \in [t_{n_{s-1}}, t_{n_s}]$ can be easily queried. Accordingly, LiDAR factors $\mathcal{C}^{lidar}_{n_s,i}$ can be explicitly modeled with the consideration of point distortion. Finally, the complete $\mathcal{G}^{global}$ with constituent factors $\mathcal{C}^{imu}_{n_{s-1},n_s}$, $\mathcal{C}^{gp}_{n_s}$, and $\mathcal{C}^{lidar}_{n_s,i}$ can be constructed and optimized in a batch. Note that after batch optimization, the incremental smoothing approach is alternatively employed to decrease the cost of full optimization if the additional follow-up data are still fed to the system.

IV. METHOD

A. STRUCTURED MAP CONSTRUCTION

1) PLANE SEGMENTATION

First, we employ voxelization to overlay the point map and store all voxels in an octree. The maximum bounding box of $\mathcal{M}$ is split recursively into eight identical child voxels until the size of the voxel or the inner point number reaches the criteria. The voxel geometric properties, i.e., normal and curvature, are calculated using the principal component analysis (PCA) algorithm. Then, voxels with similar geometric properties are clustered as the same object. Segmentation examples are shown in Fig.3, and more details can be found in [28].



(a) Original point map (b) Segmented planes with norms

FIGURE 3. The structured point map. (a) The original point map scanned by terrestrial laser scanner (TLS) and colorized with image color. (b) The segmented planes colorized with random colors. The plane norms are displayed as white lines. The scenarios are indoor corridors and open lobbies from top to bottom.

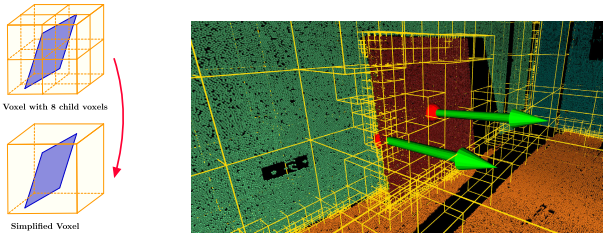
2) MAP MANAGEMENT

In the LiDAR odometry implementation, we want the depth of the octree to be as small as possible to enable more efficient point-to-plane associations. Considering the association is searching for a corresponding plane rather than a point, we truncate the octree of the clustered map $\mathcal{V}^{plane}$ by adding an additional split termination condition. Specifically, we force the octant division to stop if all voxel points belong to the same plane. The demonstration graph can be seen in Fig.4.

B. DATA PREPROCESSING

1) FEATURE EXTRACTION AND PLANE ASSOCIATION

We first arrange points in $\hat{\mathcal{S}}_n$ into ordered lines based on the mechanism of LiDAR, thus M neighbors of the selected point ${}_{l_n}\hat{\mathbf{P}}_i \in \hat{\mathcal{S}}_n$ can be indexed directly with no need to use a tree-based approach. Then, N points with the smallest smoothness



(a) Voxel truncation (b) Voxels example

FIGURE 4. Re-organization of the octree. (a) The voxel will not be split if it contains only one object during re-construction of the octree for the segmented planes. (b) Visualization of our specially designed octree. Different colors represent different planes, and green arrows are normals at the plane centroid (red point). Yellow boxes are the leaf voxels of the octree. The voxel is larger at the central region of the plane but smaller at the plane boundary. Therefore, whether the point is a near-boundary point is easy to judge by the size of the indexed voxel.

c are extracted as planar features ${}_{l_n}\hat{\mathbf{F}}_i, i \in N$:

$$c = \frac{1}{M \cdot \left\| {}_{l_n}\hat{\mathbf{P}}_i \right\|} \cdot \left\| \sum_{j \in M, j \neq i} ({}_{l_n}\hat{\mathbf{P}}_i - {}_{l_n}\hat{\mathbf{P}}_j) \right\| \tag{3}$$

Given planar features ${}_{l_n}\hat{\mathbf{F}}_i$ and the LiDAR pose prediction $\bar{\mathbf{T}}^m_{l_n}$, point-to-plane associations $\hat{\mathcal{A}}_n$ can be established by searching the octree voxel $\mathcal{V}^{plane}$ in which the transformed feature point ${}_m\hat{\mathbf{F}}_i = \bar{\mathbf{T}}^w_{l_n} \cdot {}_{l_n}\hat{\mathbf{F}}_i$ falls in. Note that points near the plane edge are excluded to ensure that the associated planes will not change along with the update of LiDAR poses during the batch optimization. Specifically, the optimization problem is solved by the iteration scheme, indicating that the linearization point needs to be re-evaluated after each iteration; thus, the target plane $\hat{\mathcal{A}}_n$ should be re-associated accordingly. To avoid repeated features association and maintain the validity of $\hat{\mathcal{A}}_n$ after iterative updating, we only retain associations with high voxel resolutions.

2) IMU PREINTEGRATION

To avoid recomputing the IMU integration whenever the linearization point changes, [29] transformed state-dependent IMU integration measurements into state-independent IMU preintegration measurements. The proposed calibration approach utilizes the PIM $\hat{\mathcal{P}}_n = [\Delta\hat{\mathbf{R}}_{n-1,n}, \Delta\hat{\mathbf{v}}_{n-1,n}, \Delta\hat{\mathbf{t}}_{n-1,n}]$, which only relates to IMU raw measurements $\hat{\mathcal{I}}_{n-1,n}$

$$\begin{aligned}
\Delta\hat{\mathbf{R}}_{n-1,n} &= \prod_{k=n-1}^{n} \exp((\hat{\boldsymbol{\omega}}_k - \bar{\mathbf{b}}^g_k)\Delta t), \\
\Delta\hat{\mathbf{v}}_{n-1,n} &= \sum_{k=n-1}^{n} [\Delta\hat{\mathbf{R}}_{n-1,k}(\hat{\boldsymbol{\alpha}}_k - \bar{\mathbf{b}}^a_k)\Delta t], \\
\Delta\hat{\mathbf{t}}_{n-1,n} &= \sum_{k=n-1}^{n} [\Delta\hat{\mathbf{v}}_{n-1,k}\Delta t + \frac{1}{2}\Delta\hat{\mathbf{R}}_{n-1,k}(\hat{\boldsymbol{\alpha}}_k - \bar{\mathbf{b}}^a_k)\Delta t^2],
\end{aligned} \tag{4}$$

where $\Delta t = t_k - t_{k-1}$. Biases $[\bar{\mathbf{b}}^a, \bar{\mathbf{b}}^g]$ are considered constants in Eq.(4) and are modeled with Brownian motion, i.e., actual bias integrated white noise $[\boldsymbol{\eta}^a, \boldsymbol{\eta}^g]$: $\mathbf{b}^a_{n-1} = \mathbf{b}^a_n + \boldsymbol{\eta}^a$, $\mathbf{b}^g_{n-1} = \mathbf{b}^g_n + \boldsymbol{\eta}^g$. Given the world gravity ${}_w\mathbf{g}$, the state residual errors $\mathbf{r}^{imu}_n = [\mathbf{r}_{\Delta\hat{\mathbf{R}}_{n-1,n}}, \mathbf{r}_{\Delta\hat{\mathbf{v}}_{n-1,n}}, \mathbf{r}_{\Delta\hat{\mathbf{t}}_{n-1,n}}]$ and bias residual errors $\mathbf{r}^{bias}_n = \mathbf{r}_{\Delta\hat{\mathbf{b}}_{n-1,n}}$ between the state variable in Eq.(2) and the PIM are easy to write as

$$\begin{aligned}
\mathbf{r}_{\Delta\hat{\mathbf{R}}_{n-1,n}} &= \log(\underbrace{[\Delta\hat{\mathbf{R}}_{n-1,n}]^\top}_{\text{PIM}} \cdot \underbrace{([\mathbf{R}^w_{b_{n-1}}]^\top \mathbf{R}^w_{b_n})}_{\text{state}}), \\
\mathbf{r}_{\Delta\hat{\mathbf{v}}_{n-1,n}} &= \underbrace{[\mathbf{R}^w_{b_{n-1}}]^\top(\mathbf{v}_{b_n} - \mathbf{v}_{b_{n-1}} - {}_w\mathbf{g}\Delta t)}_{\text{state}} - \underbrace{\Delta\hat{\mathbf{v}}_{n-1,n}}_{\text{PIM}}, \\
\mathbf{r}_{\Delta\hat{\mathbf{t}}_{n-1,n}} &= \underbrace{[\mathbf{R}^w_{b_{n-1}}]^\top(\mathbf{t}_{b_n} - \mathbf{t}_{b_{n-1}} - \mathbf{v}_{b_{n-1}}\Delta t - \frac{1}{2}{}_w\mathbf{g}\Delta t^2)}_{\text{state}} \\
&\quad - \underbrace{\Delta\hat{\mathbf{t}}_{n-1,n}}_{\text{PIM}}. \\
\mathbf{r}_{\Delta\hat{\mathbf{b}}_{n-1,n}} &= \underbrace{\mathbf{b}_n - \mathbf{b}_{n-1}}_{\text{state}}.
\end{aligned} \tag{5}$$

The bias covariance is $\boldsymbol{\Sigma}^{bias}_{n-1,n}$, and the PIM covariance $\boldsymbol{\Sigma}^{imu}_{n-1,n}$ is calculated based on the principle of covariance propagation given the covariance of IMU measurements. To sum up, the IMU factor $\mathcal{C}^{imu}_n$ can be expressed as

$$\mathcal{C}^{imu}_{n-1,n} = \frac{1}{2}\|\mathbf{r}^{imu}_n\|^2_{\Sigma^{imu}_{n-1,n}} + \frac{1}{2}\|\mathbf{r}^{bias}_n\|^2_{\Sigma^{bias}_{n-1,n}} \tag{6}$$

The operator represents $\|\mathbf{A}\|^2_{\mathbf{B}} = \mathbf{A}^\top\mathbf{B}^{-1}\mathbf{A}$.

C. INITIALIZATION

1) LiDAR ODOMETRY

Given the point-to-plane associations that link planar feature points at t_n with map planes $\mathcal{V}^{plane}_j = [\mathbf{n}^\top_j, d_j]$, LiDAR pose estimation in the map $\mathbf{T}^m_{l_n}$ can be solved by minimizing the sum of point-to-plane distances

$$\tilde{\mathbf{T}}^m_{l_n} = \min_{\mathbf{T}^m_{l_n}} \left\{ \sum_{i,j \in \hat{\mathcal{A}}_n} \mathbf{n}^\top_j \cdot \mathbf{T}^m_{l_n} \cdot {}_{l_n}\hat{\mathbf{F}}_i + d_j \right\}, \tag{7}$$

where $[\mathbf{n}^\top_j, d_j]$ represents the plane equation parameters (PEP) of the jth plane in the map.

2) CLOSED-FORM INITIALIZATION

Before fine refinement, we coarsely estimate the rotational extrinsic parameters first with a closed-form solution by solving

$$\mathbf{R}^{l_n}_{l_{n+1}} \cdot \mathbf{R}^l_b = \mathbf{R}^l_b \cdot \mathbf{R}^{b_n}_{b_{n+1}}, \tag{8}$$

which is often referred to as the hand-eye calibration equation. $\bar{\mathbf{R}}^l_b$ in Eq.(8) can be solved by decomposing the covariance matrix $\boldsymbol{\Sigma}^l_b$, which is calculated by Lie algebra mapping of the rotation matrix $\mathbf{R} = \exp(\mathbf{r}^\wedge), \mathbf{r} \in \mathbb{R}^3, \mathbf{r}^\wedge \in \mathfrak{se}(3)$

$$\begin{aligned}
\boldsymbol{\Sigma}^l_b &= \sum_i^N \mathbf{r}^{l_n}_{l_{n+1}} \cdot (\mathbf{r}^{b_n}_{b_{n+1}})^\top, \\
\bar{\mathbf{R}}^l_b &= ((\boldsymbol{\Sigma}^l_b)^\top \cdot \boldsymbol{\Sigma}^l_b)^{-\frac{1}{2}} \cdot (\boldsymbol{\Sigma}^l_b)^\top
\end{aligned} \tag{9}$$

Then, we adopt the IMU dynamic initialization approach proposed in VINS [25] to initialize the initial gravity and velocity of the IMU. The method is summarized as follows, and the detailed derivation can be found in Appendix:

- Gyroscope bias calibration. Given the initialized rotation matrix $\bar{\mathbf{R}}_b^l$, the relative rotation can be estimated by LiDAR odometry $\hat{\mathbf{R}}_{l_{n+1}}^{l_n}$ and PIM $\Delta\hat{\mathbf{R}}_{n-1,n}$. Consequently, the gyroscope bias can be found by minimizing residuals between them.
- Gravity vector approximation. Similarly, the IMU position and velocity can also be estimated by LiDAR odometry $\hat{\mathbf{T}}_{l_{n+1}}^{l_n}$ and PIM $\Delta\hat{\mathbf{v}}_{n-1,n}$, $\Delta\hat{\mathbf{t}}_{n-1,n}$, respectively. By minimizing their residuals, we can initialize the gravity vector ${}_{l_0}\hat{\mathbf{g}}$ in the LIDAR reference frame.
- Gravity refinement and velocity initialization. With prior knowledge of the gravity magnitude g, ${}_w\mathbf{g}$ can be parameterized as ${}_w\mathbf{g} = g \cdot {}_{l_0}\hat{\mathbf{g}} + w_1\mathbf{b_1} + w_2\mathbf{b_2}$ to enforce the norm of gravity, where $[\mathbf{b_1}, \mathbf{b_2}]$ is a pair of orthonormal bases in the tangent space of ${}_{l_0}\hat{\mathbf{g}}$. By substituting ${}_w\mathbf{g}$ into the cost function described in the last step, both gravity parameters and initial velocities can be estimated in a batch.

Note that we reintegrate the high-frequency IMU pose predictions at t_{n_s} based on the LiDAR odometry results and IMU measurements after the initialization and recalculate the PIMs $\hat{\mathcal{I}}_{n_{s-1},n_s}$ to render them consistent with the frequency of the to-be-estimated states in the optimizer.

D. CALIBRATION

1) MOTION MODEL

We define the Gaussian process prior factor based on the motion model of the IMU. When the GP is generated by LTV-SED, Barfoot *et al.* demonstrated that the GP regression always maintains sparsity [11], [27]. We summarize how GP priors are defined on Lie groups in this section. For more details, readers are encouraged to view [30] and [31].

Assuming that $\mathbf{T}_{b_{n_s}}^w$ is a discrete IMU state on the trajectory, the local continuous-time trajectory $\mathbf{T}_b^w(t)$ can be expressed as a local GP, where $t \in [t_{n_s}, t_{n_{s+1}}]$. Namely, the interpolated IMU state can be expressed as the state at t_{n_s} right-multiplying state variation $\boldsymbol{\xi}_{b_{n_s}}^w(t)$ between t_{n_s} and t, which is defined as a GP

$$\mathbf{T}_{b_{n_s}}^w(t) = \mathbf{T}_{b_{n_s}}^w \exp\left(\boldsymbol{\xi}_{b_{n_s}}^w(t)^\wedge\right), \quad \boldsymbol{\xi}_{b_{n_s}}^w(t) \sim \mathcal{N}\left(\mathbf{0}, \mathcal{K}(t_{n_s}, t)\right), \tag{10}$$

where $\mathcal{K}(t_{n_s}, t)$ is the covariance of the variation. The local variation of pose representing the transformation from $\mathcal{F}_{b_{n_s}}$ to $\mathcal{F}_t$ around $\mathbf{T}_{b_{n_s}}^w$ is defined by the vector

$$\begin{aligned}\boldsymbol{\xi}_{b_{n_s}}^w(t) &= \log\left(\left(\mathbf{T}_{b_{n_s}}^w\right)^{-1}\mathbf{T}_{b_{n_s}}^w(t)\right)^\vee,\\ \dot{\boldsymbol{\xi}}_{b_{n_s}}^w(t) &= \left(\mathcal{J}_r\left(\boldsymbol{\xi}_{b_{n_s}}^w(t)\right)\right)^{-1} \cdot {}_{b_{n_s}}\boldsymbol{\varpi}_{n_s}(t),\end{aligned} \tag{11}$$

where $\mathcal{J}_r(\mathbf{C})$ is the right Jacobian of $\mathbf{C}$. The LTV-SDE indicates that the local acceleration ${}_{b_n}\mathbf{a}(t)$ can be modeled by the GP with a zero mean

$$\ddot{\boldsymbol{\xi}}_{b_{n_s}}^w(t) = {}_{b_{n_s}}\mathbf{a}(t),\ {}_{b_{n_s}}\mathbf{a}(t) \sim \mathcal{GP}\left(\mathbf{0}, \mathbf{Q}_C\delta(t-t')\right). \tag{12}$$

In Eq.(12), $\mathbf{Q}_C$ is a power-spectral density matrix, which is a hyperparameter, and $\delta(t-t')$ is the Dirac delta function. We define the body-frame Markov state as a 12×1 vector, i.e., the transformation vector augmenting with the body velocity vector introduced in Eq.(11)

$$\boldsymbol{\gamma}_{b_{n_s}}^w(t) = \begin{bmatrix}\boldsymbol{\xi}_{b_{n_s}}^w(t)\\ \dot{\boldsymbol{\xi}}_{b_{n_s}}^w(t)\end{bmatrix}. \tag{13}$$

Then, the local LTV-SDE can be rewritten as

$$\dot{\boldsymbol{\gamma}}_{b_{n_s}}^w(t) = \frac{d}{dt}\begin{bmatrix}\boldsymbol{\xi}_{b_{n_s}}^w(t)\\ \dot{\boldsymbol{\xi}}_{b_{n_s}}^w(t)\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{\xi}}_{b_{n_s}}^w(t)\\ {}_{b_{n_s}}\mathbf{a}(t)\end{bmatrix}. \tag{14}$$

When the time interval is small, $\boldsymbol{\xi}_{b_{n_s}}^w(t)$ approximates a zero vector. Because $\mathcal{J}_r\left(\boldsymbol{\xi}_{b_{n_s}}^w(t)\right) \approx \mathbf{I}$, Eq.(12) and Eq.(14) can be easily proven to be good approximations of the constant body-frame velocity prior, i.e., $\dot{\boldsymbol{\xi}}_{b_{n_s}}^w(t) \approx {}_{b_{n_s}}\boldsymbol{\varpi}_{n_s}(t)$. Considering Eq.(11), the body-frame velocity can be rewritten as

$$\dot{\boldsymbol{\xi}}_{b_{n_s}}^w(t) \approx {}_{b_{n_s}}\boldsymbol{\varpi}_{n_s}(t) = \left(\left(\mathbf{T}_{b_{n_s}}^w(t)\right)^{-1}\dot{\mathbf{T}}_{b_{n_s}}^w(t)\right)^\vee. \tag{15}$$

On the basis of the LTV-SED prior model, the residual errors between the GP prior and states in Eq.(2) are specified as the GP prior factor $\mathbf{r}_{n_s}^{gp}$, where

$$\begin{aligned}\mathbf{r}_{n_s}^{gp} &= \underbrace{\boldsymbol{\Phi}_{s-1,s}\cdot\boldsymbol{\gamma}_{b_{n_{s-1}}}^w(t_{n_{s-1}})}_{\text{GP prior}} - \underbrace{\boldsymbol{\gamma}_{b_{n_{s-1}}}^w(t_{n_s})}_{\text{state}},\\ \mathcal{C}_{n_s}^{gp} &= \frac{1}{2}\|\mathbf{r}_{n_s}^{gp}\|_{\boldsymbol{\Sigma}_{s-1,s}^{gp}}^2.\end{aligned} \tag{16}$$

In the above equation, $\boldsymbol{\Phi}_{s-1,s}$ is the transition matrix, and $\boldsymbol{\Sigma}_{s-1,s}^{gp}$ is the corresponding covariance matrix, where

$$\begin{aligned}\Delta t &= (t_{n_s} - t_{n_{s-1}}),\\ \boldsymbol{\Phi}_{s-1,s} &= \begin{bmatrix}\mathbf{1} & \Delta t\cdot\mathbf{1}\\ \mathbf{0} & \mathbf{1}\end{bmatrix},\\ \boldsymbol{\Sigma}_{s-1,s}^{gp} &= \begin{bmatrix}\frac{1}{3}\Delta t^3\mathbf{Q}_C & \frac{1}{2}\Delta t^2\mathbf{Q}_C\\ \frac{1}{2}\Delta t^2\mathbf{Q}_C & \Delta t^2\mathbf{Q}_C\end{bmatrix}.\end{aligned} \tag{17}$$

2) INTERPOLATED LiDAR FACTORS

Given the continuous-time trajectory and the exact timestamps of each point, the accurate LiDAR point constraint can be modeled.

a: QUERYING THE TRAJECTORY

An advantage of the Gaussian process representation of the trajectory is that any state can be interpolated by two nearby states. Specifically, the continuous-time state $\tilde{\boldsymbol{\gamma}}_{b_{n_s}}^w(t_\tau)$, $t_\tau \in [t_{n_s}, t_{n_{s+1}}]$ is the linear combination of two neighbor states $\tilde{\boldsymbol{\gamma}}_{b_{n_s}}^w$ and $\tilde{\boldsymbol{\gamma}}_{b_{n_{s+1}}}^w$

$$\tilde{\boldsymbol{\gamma}}_{b_{n_s}}^w(t_\tau) = \boldsymbol{\Lambda}_\tau\tilde{\boldsymbol{\gamma}}_{b_{n_s}}^w(t_{n_s}) + \boldsymbol{\Psi}_\tau\tilde{\boldsymbol{\gamma}}_{b_{n_s}}^w(t_{n_{s+1}}), \tag{18}$$

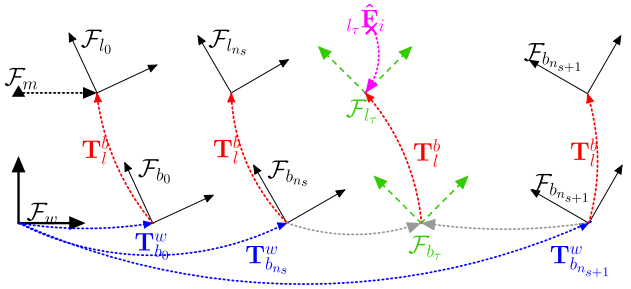


FIGURE 5. An illustration of point transformation. The extrinsic parameters $\mathbf{T}_I^b$ transform a point ${}_{I_\tau}\hat{\mathbf{F}}_i$ in the LiDAR frame to an IMU frame. Then, the pose at $\mathcal{F}_{b_\tau}$ can be interpolated by states at $\mathcal{F}_{b_{n_s}}$ and $\mathcal{F}_{b_{n_s+1}}$; Next, the transformed point in the current IMU frame is transformed into the first IMU frame $\mathcal{F}_{b_0}$ by the queried IMU state. Finally, the transformed point is projected back to the first LiDAR frame $\mathcal{F}_{l_0}$ by $\mathbf{T}_I^b$, which is also the map frame $\mathcal{F}_m$.

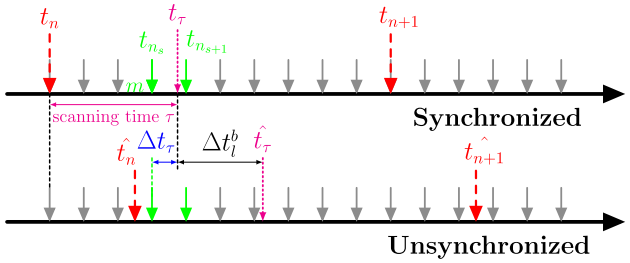


FIGURE 6. An illustration of time offsets between LiDAR and the IMU. The upper plot shows a synchronized case, while the lower shows an unsynchronized case. Gray arrows represent IMU timestamps, and red arrows represent LiDAR timestamps. Affected by the temporal offset Δt_l^b, LiDAR timestamps are shifted. The feature point scanned at t_τ denoted by the magenta arrow is shifted to $\hat{t}_\tau$. With the reading of $\hat{t}_n$ and τ, t_τ is expressed as a value dependent on Δt_l^b. To interpolate the state at t_τ, we further calculate the interval Δt_τ by subtracting the timestamp of the related state t_{n_s}.

where

$$\begin{aligned}\mathbf{\Lambda}_\tau &= \mathbf{\Phi}_{s,\tau} - \mathbf{\Sigma}_{s,\tau}^{gp}\mathbf{\Phi}_{s,\tau}^\top\left(\mathbf{\Sigma}_{s,s+1}^{gp}\right)^{-1}\mathbf{\Phi}_{s,s+1},\\ \mathbf{\Psi}_\tau &= \mathbf{\Sigma}_{s,\tau}^{gp}\mathbf{\Phi}_{s,\tau}^\top\left(\mathbf{\Sigma}_{s,s+1}^{gp}\right)^{-1}.\end{aligned} \tag{19}$$

Consequently, the continuous-time trajectory can be queried using two anchor states, i.e., substituting Eq.(18) into Eq.(10),(11), and (15).

b: FUSION OF MOTION DISTORTION

For the sweep $\hat{\mathcal{S}}_n$, the reference frame of each feature ${}_{l_n}\hat{\mathbf{F}}_i$ cannot be simply treated as $\mathcal{F}_{l_n}$, because the LiDAR is moving. Given the real scanning timestamp $t_\tau \in [t_{n_s}, t_{n_{s+1}}]$, the real transform $\mathbf{T}_{b_\tau}^w$ can be queried using Eq.(18). Therefore, feature points can be transformed into the map frame $\mathcal{F}_m$ (see Fig.(5)) by

$${}_m\hat{\mathbf{F}}_i = \tilde{\mathbf{T}}_b^l\left(\tilde{\mathbf{T}}_{b_0}^w\right)^{-1}\tilde{\mathbf{T}}_{b_\tau}^w\left(\tilde{\mathbf{T}}_b^l\right)^{-1}\cdot{}_{l_\tau}\hat{\mathbf{F}}_i. \tag{20}$$

c: FUSION OF TEMPORAL OFFSETS

Considering temporal offset Δt_l^b, the point timestamp reading $\hat{t}_n$ should compensate for the offset to obtain the actual queried time t_τ

$$t_\tau = \underbrace{\hat{t}_n + \tau}_{\hat{t}_\tau} - \Delta t_l^b. \tag{21}$$

Consequently, the time interval between the nearest anchor state and the queried point is $\Delta t_\tau = t_\tau - m\cdot\Delta t$, where $m = \lfloor\tau/\Delta t\rfloor$, and Δt is the interval of IMU states (see Fig.6).

d: INTERPOLATED LiDAR FACTOR

As described above, we define the LiDAR factor as the distance of the feature point to the associated plane, which is a variable dependent on the state of $\tilde{\mathbf{T}}_{b_\tau}^w$. Meanwhile, $\tilde{\mathbf{T}}_{b_\tau}^w$ is related to the point scanning timestamp τ and the time shift Δt_l^b. In general, a LiDAR constraint links the states declared in Eq.(2).

$$\begin{aligned}\mathbf{r}_{n_s,i}^{lidar} &= \underbrace{\mathbf{n}_j^\top}_{\text{PEP}}\cdot\underbrace{\tilde{\mathbf{T}}_b^l\left(\tilde{\mathbf{T}}_{b_0}^w\right)^{-1}\tilde{\mathbf{T}}_{b_\tau}^w\left(\tilde{\mathbf{T}}_b^l\right)^{-1}}_{\text{states}}\cdot\underbrace{{}_{l_\tau}\hat{\mathbf{F}}_i}_{\text{feature}} + \underbrace{d_j}_{\text{PEP}},\\ \mathcal{C}_{n_s,i}^{lidar} &= \frac{1}{2}\|\mathbf{r}_{n_s,i}^{lidar}\|^2_{\Sigma_{n,i,j}^{lidar}},\end{aligned} \tag{22}$$

where $\mathbf{\Sigma}_{n,i,j}^{lidar}$ is the covariance matrix of the ith point associated with the jth plane in the nth sweep, which can also be calculated based on the principle of noise propagation with known LiDAR observation noise.

3) UPDATING OF THE GLOBAL FACTOR GRAPH BY CONTINUOUS-TIME BATCH OPTIMIZATION

The factor graph $\mathcal{G}^{global}$ is composed of all constrained factors introduced above, i.e., PIM factors, GP prior factors and interpolated LiDAR factors (see Fig.7).

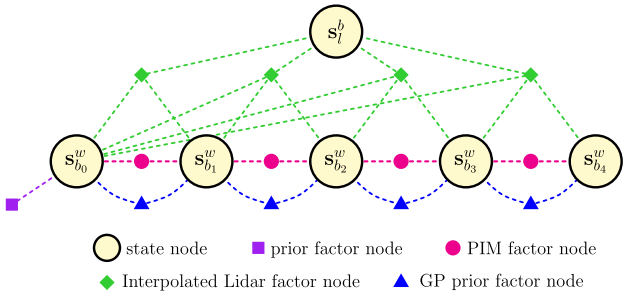


FIGURE 7. Visualization of the factor graph.

The calibration problem is formulated as maximum a posteriori estimation (MAP) in which the unbiased optimal estimations of states can be solved by minimizing the sum of all constrained factors, i.e.,

$$\tilde{\mathcal{X}} = \min_{\tilde{\mathcal{X}}}\left\{\sum\mathcal{C}_{n_{s-1},n_s}^{imu} + \sum\mathcal{C}_{n_s}^{gp} + \sum\mathcal{C}_{n_s,i}^{lidar}\right\} \tag{23}$$

TABLE 1. Simulation noise parameters.

Sensor	Noise term	Value
IMU	gyroscope white noise $\sigma_g(rad/s\sqrt{Hz})$	0.01
	gyroscope random walk $\sigma_b g(rad/s^2\sqrt{Hz})$	0.0025
	gyroscope initial bias $\sigma_{ing}(rad/s)$	0.2
	accelerometer white noise $\sigma_g(m/s^2\sqrt{Hz})$	0.6
	accelerometer random walk $\sigma_g(rad/s^3\sqrt{Hz})$	0.0075
	accelerometer initial bias $\sigma_g(m/s^2)$	0.05
LIDAR	LiDAR range white noise $\sigma_l(m)$	0.03
TLS	TLS range white noise $\sigma_m(m)$	0.003

The Levenberg-Marquardt algorithm is utilized to solve the well-constrained factor graph considering the speed of the iterative convergence for such a large optimization problem.

V. EXPERIMENTS

To thoroughly validate the proposed method, we conducted extensive experiments on both simulated and real-world data. The simulation experiment was conducted first to investigate the influences of some relevant factors on calibration accuracy, e.g., the choice of map accuracy, state frequency, and gather implementation experience. Then, the self-assembled backpack shown in Fig.1 was calibrated and used to collect indoor/outdoor mapping data from a local environment. In the real-world calibration experiment, the reference map was obtained by the high-accuracy FARO FocusS 150 TLS.[1] The map can also be built by some LiDAR-only mapping approaches [7], [32], [33] with smooth sampling motion since the validity of this map was verified in the simulation experiment. Note that the approximate initial pose of LiDAR in the prior map must be provided to address the global localization issue. The initial pose was refined by aligning the first sweep with the reference map using the generalized iterative closest point (GICP) algorithm.

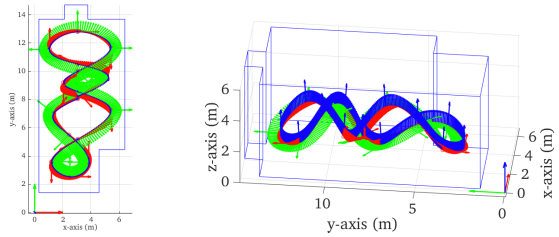
A. SIMULATION

Considering the difficulty of acquiring the exact calibration ground truth for an LiDAR/IMU device, we carried out a series of simulation experiments to quantitatively evaluate the performance of the proposed approach. We simulated an indoor office and randomly sampled a 6-DOF trajectory using the 3D spline interpolation method [34]. The simulated experiment example is shown in Fig.8. The simulated 9-axis IMU exports accelerations, angular velocities, and orientations at $400Hz$, while the LiDAR exports 16-line sweeps at $10Hz$ with a horizontal FOV of 360° and a vertical FOV of ±15°. The point map is obtained by registering simulated TLS points from different control sites in the virtual office. Other simulated parameters are listed in Tab.1.

1) MONTE-CARLO ACCURACY ANALYSIS

We utilized 10 random sequences with the same length of $35s$ to conduct a Monte-Carlo analysis. One example can

[1]https://www.faro.com/Products/Hardware/Focus-Laser-Scanners



(a) Bird-eye view (b) Lateral view

FIGURE 8. Simulated indoor office and LiDAR poses. Blue polygons represent walls, while axes represent the spline trajectory (the red, green, blue correspond to [*x*, *y*, *z*] respectively).

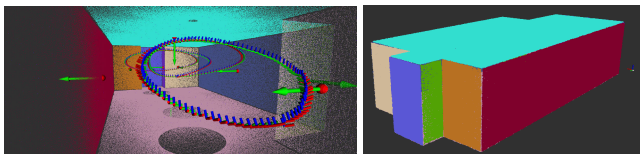


FIGURE 9. Simulation experiment example. The structured simulated point map and estimated LiDAR odometry trajectory.

be viewed in Fig.9. In the simulation, the map accuracy was set as $\sigma_m = 0.003m$, the *extrinsic positional and angular parameters* were randomly set as $[0.0, 0.05, -0.1]m$ and $[67.0, 11.0, 16.0]°$ respectively, and the *temporal offset* was set as 10.0ms. The calibrated angular error was $\mathbf{r}_{\bar{\boldsymbol{\theta}}_b^l} = [0.0366, 0.0141, 0.0186]°$ with standard deviations of $\boldsymbol{\sigma}_{\boldsymbol{\theta}} = \pm[0.0030, 0.0022, 0.0019]$. The displacement error was $\mathbf{r}_{\bar{\mathbf{t}}_b^l} = [0.0026, 0.0017, 0.0042]m$ with $\boldsymbol{\sigma}_{\mathbf{t}} = \pm[0.0005, 0.0003, 0.0007]$. The calibrated temporal offset error was $\mathbf{r}_{\Delta\bar{t}_b^l} = 0.82ms$ with standard deviations of $\boldsymbol{\sigma}_{\Delta t} = \pm 0.024$, showing that the calibration accuracy of spatial parameters is achieved at centimeter-level and that of the time shift is less than $1ms$.

2) THE CHOICE OF MAP

To explore the sensitivity to the accuracy of the prior map, we conducted the same experiments in the last section but substituted the prior map with different accuracies, i.e., $\sigma_m = 0.05m, 0.1m$. Generally, the accuracy of the map generated by LiDAR mapping approaches is no less than $0.1m$. Hence, the simulation experiment can provide credible evidence that LiDAR-based mapping approaches can also be used to build the map. The calibration errors are shown in Fig.10. The range of errors becomes larger from blue markers to green markers, indicating that the precision of calibration decreases with increasing map noise. The average angular calibration errors were $\mathbf{r}_{\bar{\boldsymbol{\theta}}_b^l} = [0.0613, 0.0582, 0.0424]°$ for $\sigma_m = 0.05m$ (magenta markers in the top subfigure) and $\mathbf{r}_{\bar{\boldsymbol{\theta}}_b^l} = [0.2259, 0.1048, 0.1653]°$ for $\sigma_m = 0.1m$ (green markers in the top subfigure). In addition, the average translational errors were $\mathbf{r}_{\bar{\mathbf{t}}_b^l} = [0.0082, 0.0055, 0.0136]m$ for $\sigma_m = 0.05m$ and $\mathbf{r}_{\bar{\mathbf{t}}_b^l} = [0.0315, 0.0201, 0.0356]m$ for $\sigma_m = 0.1m$. The result demonstrates that the centimeter-level accuracy (i.e., no less

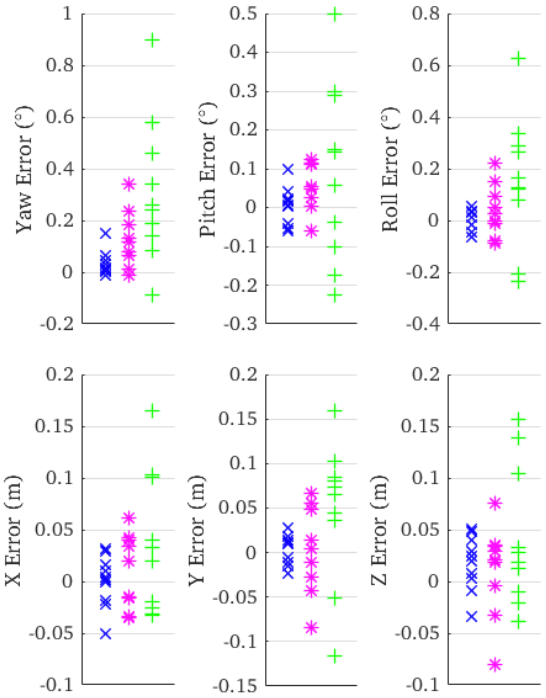


FIGURE 10. Performance with respect to map accuracy. Blue markers represent the result with map noise of 3*mm*, while magenta and green represent results with 5*cm* and 10*cm* of noise, respectively.

than 5cm) of the prior map is sufficient for our calibration approach.

3) THE CHOICE OF STATE FREQUENCY

As per the data-driven nature of GP regression associated with the constant velocity prior, it's interesting to analyze the choice of state frequencies. We selected 5 different frequencies, i.e., 400Hz, 100Hz, 50Hz, 10Hz, and 5Hz, as contrast objects, implying that the numbers of states and constraints in the optimizer are different. For instance, we insert 400 PIM constraints and 400 GP constraints per second if the state frequency is set as 400Hz. In the experiment, sequences with the same trajectory but different lengths were simulated to vary the sensor motion velocities. The length ranged from 35s to 17s with a decreasing step of 2s. The comparison is shown in Fig.11. The plot shows that the error curve of 400Hz is the steadiest, but that of 5Hz fluctuates over different sequences, indicating that higher selected frequency corresponds to more robust motion aggressivity. In addition, the calibration result is reasonable as long as the state frequency is higher than 10Hz.

B. REAL-WORLD DATA

In the real-world experiment, we tried to calibrate the self-assembled backpack with 2 × Robosense 16-line LiDARs[2] and 1 × Xsens Mti-G-710 IMU[3] shown in Fig.1.

[2]http://www.robosense.ai/rslidar/RS-LiDAR-16
[3]https://www.xsens.com/products/mti-100-series

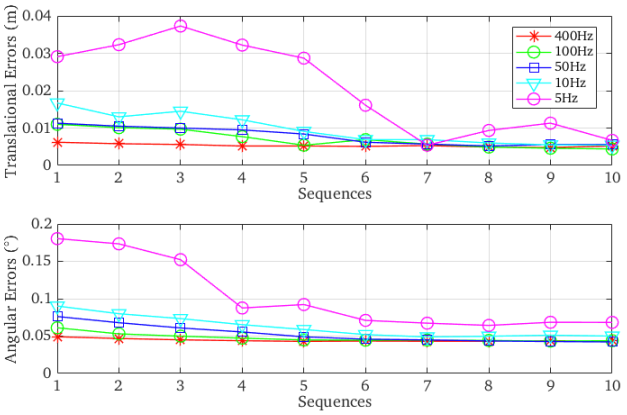


FIGURE 11. Performance with respect to the state frequency. The norms of angular and translational errors are displayed in the graph to demonstrate the changes in errors over different sequences. The lateral axis represents the sequence number, i.e., 1 to 10 indicating the fastest sequence with a length of 17s to the slowest sequence with a length of 35s, which implies varying motion aggressivity.

The IMU and LiDARs are synchronized by hardware. We collected calibration data in two typical indoor environments, i.e., a narrow corridor and an open building lobby(see Fig.3), with a length of 30∼40s. We sampled the data along the spatial sinusoidal trail to ensure the observability of the IMU. In the experiment, the first 30s of data were accumulated for the coarse-to-fine calibration, i.e., initialization first, and then continuous-time batch optimization. Next, the follow-up data were consecutively fed to the system, triggering incremental smoothing to keep the efficiency of the full-optimization. For accuracy and efficiency, the state frequency in the optimizer was set as 100Hz in our experiments. Three tests were conducted to analyze the convergence, repeatability, and accuracy of our method.

1) CONVERGENCY

Optimization convergence is important to investigate since the calibration procedure should output stable results after calibration optimization. We utilized the sequence collected in the narrow corridor to calibrate our device. The first 30s of data were accumulated to perform the initialization and batch optimization, and the follow-up 5s of data were fed to the system to employ the incremental smoothing optimization. The results are depicted in Fig.12. Both the translation and rotational results generally stabilize after batch optimization, which validates the convergence of the proposed calibration method. Although the translational error in the z-axis shows the largest fluctuation, it is still no greater than 0.01m.

2) REPEATABILITY

The repeatability of calibration results must be discussed based on the different segments of the same dataset. In other words, the calibration results should be consistent using different segments of the same calibration dataset. In the test, we utilized both corridor and lobby sequences to calibrate

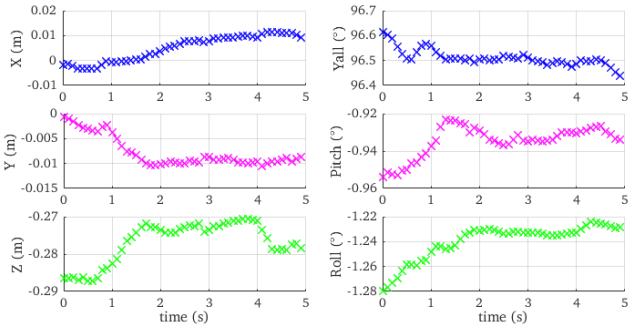


FIGURE 12. The calibration results with respect to the sampling time. The calibration results stabilize after batch optimization.

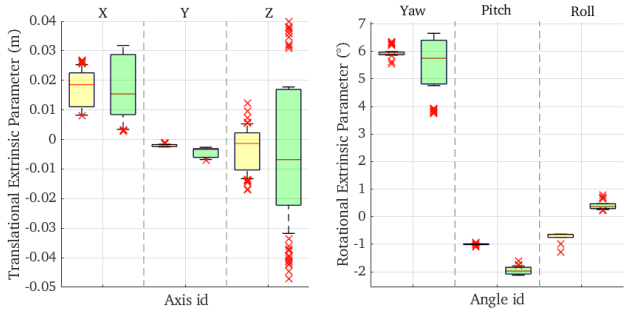


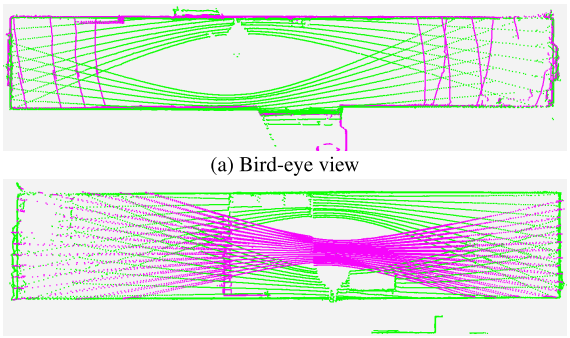
FIGURE 13. Box plots of extrinsic parameters for both corridor and lobby data. Yellow and green boxes represent the results of corridors and lobbies, respectively. The box ranges from 25% to 75% of the samples, and the red line in each box is the median. Outliers are identified by a red cross. Note that *Z* and *Yaw* are normalized by their CAD references, i.e., 0.3*m* and −90°, respectively, to fit the figure size.

TABLE 2. Calibration results.

Method	LiDAR Id	Extrinsic Parameters					
		$x(m)$	$y(m)$	$z(m)$	$yaw(°)$	$pitch(°)$	$roll(°)$
Proposed method	Upright LiDAR	0.0049	-0.0078	-0.2768	96.5111	-0.9346	-1.2397
	Tilted LiDAR	0.0078	0.1274	-0.1789	91.5344	0.8692	-47.4180
LI-Calib	Upright LiDAR	0.0076	-0.0014	-0.2681	96.2840	-1.3544	-0.9503
	Tilted LiDAR	0.0032	0.1199	-0.1803	91.3834	1.2198	-48.0202

TABLE 3. Relative calibration accuracy.

Method	Extrinsic Parameters						Relative Accuracy	
	$x(m)$	$y(m)$	$z(m)$	$yaw(°)$	$pitch(°)$	$roll(°)$	translation	rotation
Proposed method	-0.0203	0.0953	0.1356	-4.9386	1.9074	-46.2634	0.0097	0.5128
Li-Calib	-0.0103	0.0848	0.1231	-4.8613	2.6501	-47.1882	0.0228	1.3806
Dual-LiDAR-Calib	-0.0131	0.1009	0.1390	-5.0278	2.2375	-45.8812	-	-



(a) Bird-eye view

(b) Lateral view

FIGURE 14. Registration result of the corresponding sweeps of two LiDARs. The green points are obtained by the tilted LiDAR, and the magenta by the upright LiDAR.

the backpack and made use of the statistic to investigate the repeatable performance of the proposed approach. Each sequence is subsampled from 30∼40*s* with an increasing step of 1*s*, i.e., $t = 30 + \Delta, \Delta \in [0, 10]$. To enlarge the statistical samples, we recalculated the results 5 times for each sequence; thus, 55 calibration results were obtained in total. The results are displayed using box plots in Fig.13. The length of the box represents the degree of instability. We found that all yellow boxes are shorter than the green boxes, indicating that the calibration is more stable in narrow spaces. In addition, the translational errors are generally centimeter-scale, and the rotational errors with respect to all axes are no greater than 1.5°.

3) RELATIVE ACCURACY

In the third test, we calibrated the extrinsic parameters T_{l1}^{b} and T_{l2}^{b} between two LiDARs and an IMU. The state-of-art calibration method Li-Calib [23] was selected for comparison. The results are listed in Tab.2. The differences between our method and Li-Calib were 0.0087*m* and 0.2254° for the upright LiDAR/IMU and 0.0032*m* and 0.1482° for the tilted LiDAR/IMU. The small differences illustrate that our approach achieves similar accuracy to LI-Calib.

We designed a tricky method to quantitatively evaluate the relative accuracies of the backpack calibration results. Given the extrinsic LiDAR/IMU parameters, the transformation between two LiDARs can be estimated by $\tilde{T}_{l2}^{l1} = {T_{l1}^{b}}^{-1} \cdot T_{l2}^{b}$. Consequently, the relative accuracies of our approach and Li-Calib can be indirectly evaluated by comparing the results with those estimated by the dual-LiDAR calibration method [35]. The errors are 9.7*mm*/0.5128° for our method and 22.8*mm*/1.3806° for Li-Calib (see Tab.3), which shows that the proposed calibration approach has better performance to some extent. Moreover, to intuitively and qualitatively display the relative accuracy, we registered two laser sweeps scanned at the same timestamp using the calibration result of the proposed method (see Fig.14). The two laser sweeps register well, indicating that the relative accuracy of the calibration result is accurate.

In addition, a large-scale indoor-outdoor dataset was collected, and the mapping results were used to indirectly reflect the calibration accuracy. Three different methods were utilized to incrementally construct the map, i.e., the IMU/LiDAR coupled mapping approach with accurate extrinsic parameters, the IMU/LiDAR approach with coarse extrinsic parameters, and the LiDAR-only mapping approach. We projected the point cloud map onto the satellite image and displayed some map details. The comparison can be viewed in Fig.15. The total distance of the trajectory

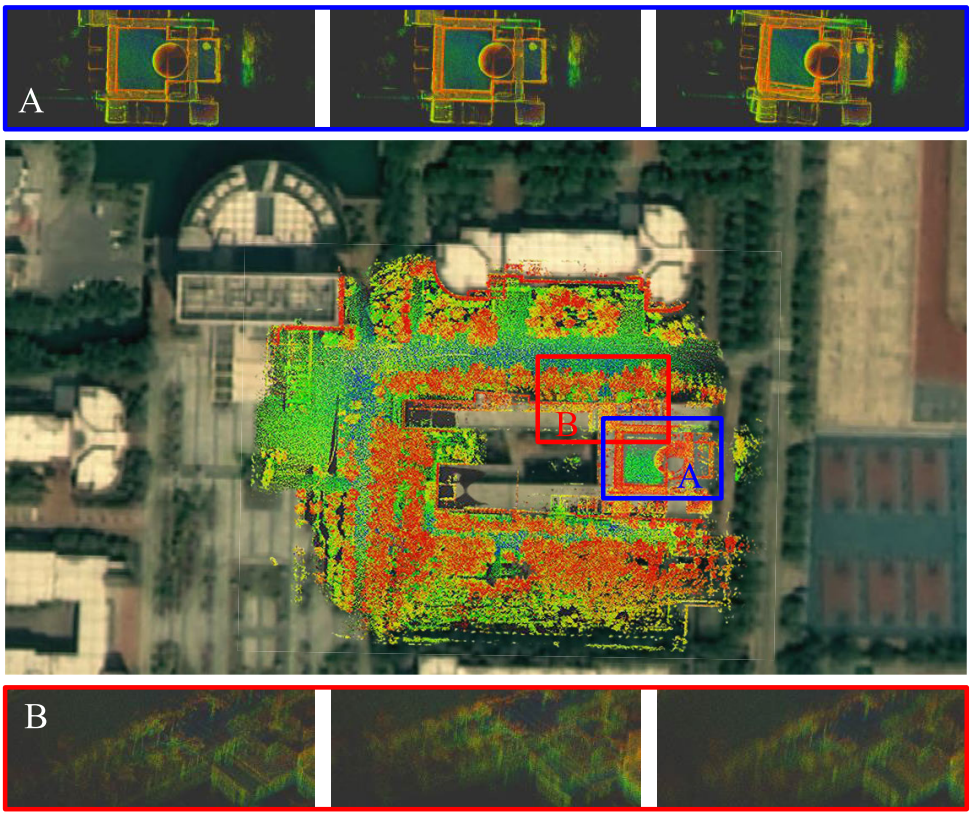


FIGURE 15. Mapping results and point map details. Middle: birds-eye view of the point map projected onto the satellite image of the mapping area. Top/bottom: details circled out in the map. Each column from left to right represents the result with accurate calibration parameters, the result with an inaccurate calibration estimate, and the result without an IMU.

is approximately $1km$, and the sampling time is $500s$. The bottom row of the figure clearly shows that the LiDAR-only approach is likely to fail at turns. From the top row, trees along roads can be clearly distinguished in the left figure but are blurred in the other two, demonstrating that the consistency of the map is improved using the accurate calibration result.

VI. CONCLUSION

In this paper, we present a high-accuracy autocalibration method for LiDAR/IMU rigs. The proposed approach adopts a continuous-time trajectory estimation wherein the trajectory is viewed as a GP regression. The distorted and temporally shifted points are explicitly modeled by querying the exact transform on the continuous-time trajectory. To mitigate the effect of incorrect data associations in the LiDAR odometry module, we construct a structured map in advance and manage the map by the specially designed octree to facilitate robust point-to-map association. Finally, extensive simulation experiments were conducted to investigate the sensitivity to the accuracy of the prior map and state frequency. This study shows that the proposed method is relevant to the accuracy of the prior map and the state frequency in the optimizer. Moreover, the experiments show that our calibration method is highly accurate, robust, and repeatable, especially in narrow spaces.

APPENDIX
IMU DYNAMIC INITIALIZATION

In this section, we derive the formulation for IMU dynamic initialization, which is discussed in Sect.IV-C. The problem can be solved in closed-form under the assumption that the biases of the IMU are constant. The approach is an adaption of the method introduced in [25]. Let us start from the obtained data pairs of LiDAR odometry estimates $\bar{\mathbf{T}}_{l_n}^{m}$ and PIM $\hat{\mathcal{P}}_n$. Then, three steps, i.e., a. gyroscope bias calibration, b. initial gravity vector approximation, and c. gravity refinement and velocity initialization, are carried out consecutively.

A. GYROSCOPE BIAS CALIBRATION

Considering two consecutive frames, $\mathcal{F}_{b_n}$ and $\mathcal{F}_{b_{n+1}}$, the rotation part of the IMU PIM $\Delta\hat{\mathbf{R}}_{n,n+1}$ is the rotation increment with clear physical meaning. Hence, the gyroscope biases can be estimated by minimizing the rotation increment residual of LiDAR odometry and IMU PIM:

$$\min_{\mathbf{b}_g} \sum_{n} \underbrace{\bar{\mathbf{T}}_l^b(\bar{\mathbf{R}}_{l_0}^{l_n} \cdot \bar{\mathbf{R}}_{l_{n+1}}^{l_0})[\bar{\mathbf{T}}_l^b]^{-1}}_{\text{LO}} - \underbrace{\Delta\bar{\mathbf{R}}_{n,n+1}}_{\text{PIM}},$$

$$\Delta\bar{\mathbf{R}}_{n,n+1} \approx \Delta\hat{\mathbf{R}}_{n,n+1} + \frac{\partial\Delta\bar{\mathbf{R}}_{n,n+1}}{\partial\mathbf{b}_g}\delta\mathbf{b}_g. \quad (24)$$

Then, the other two parts of PIM related to gyroscope biases can be updated:

$$\Delta\bar{\mathbf{v}}_{n,n+1} \leftarrow \Delta\hat{\mathbf{v}}_{n,n+1} + \frac{\partial\Delta\hat{\mathbf{v}}_{n,n+1}}{\partial\mathbf{b}_g}\delta\mathbf{b}_g,$$
$$\Delta\bar{\mathbf{t}}_{n,n+1} \leftarrow \Delta\hat{\mathbf{t}}_{n,n+1} + \frac{\partial\Delta\hat{\mathbf{v}}_{n,n+1}}{\partial\mathbf{b}_g}\delta\mathbf{b}_g. \quad (25)$$

B. INITIALIZING THE INITIAL GRAVITY VECTOR

Referring to [29], the translation part of the PIM can be rewritten as:

$$\Delta\hat{\mathbf{t}}_{n,n+1} = [\mathbf{R}^w_{b_n}]^\top(\mathbf{t}_{b_{n+1}} - \mathbf{t}_{b_n} - \mathbf{v}_{b_n}\Delta t_{ij} - \frac{1}{2}\sum_{k=i}^{j-1} {}_w\mathbf{g}\Delta t^2). \quad (26)$$

Given the LO, the above equation can be written as:

$$[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_{n,n+1} = [\bar{\mathbf{R}}^{l_0}_{l_n}]^\top(\bar{\mathbf{t}}_{l_{n+1}} - \bar{\mathbf{t}}_{l_n} - \bar{\mathbf{v}}_{l_n}\Delta t_{n,n+1} - \frac{1}{2}\sum_{k=n}^{n+1} {}_{l_0}\mathbf{g}\Delta t^2). \quad (27)$$

Then, both sides of the Eq.(27) are multiplied by $\mathbf{R}^{l_0}_{l_n}$:

$$\frac{1}{2}\sum_{k=n}^{n+1} {}_{l_0}\mathbf{g}\Delta t^2 = (\bar{\mathbf{t}}_{l_{n+1}} - \bar{\mathbf{t}}_{l_n}) - \bar{\mathbf{R}}^{l_0}_{l_n}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_{n,n+1} - \bar{\mathbf{v}}_{l_n}\Delta t_{n,n+1}. \quad (28)$$

However, the estimation of LiDAR odometry does not include the velocity of LiDAR $\bar{\mathbf{v}}_{l_n}$, indicating that the term should be offset. Considering the velocity part of PIM:

$$\Delta\hat{\mathbf{v}}_{n,n+1} = [\mathbf{R}^w_{b_n}]^\top(\mathbf{v}_{b_{n+1}} - \mathbf{v}_{b_n} - {}_w\mathbf{g}\Delta t_{n,n+1}), \quad (29)$$

the above equation can be transformed as:

$$[\bar{\mathbf{R}}^b_l]^{-1}\Delta\hat{\mathbf{v}}_{n,n+1} = [\bar{\mathbf{R}}^{l_0}_{l_n}]^\top(\bar{\mathbf{v}}_{l_{n+1}} - \bar{\mathbf{v}}_{l_n} - {}_{l_0}\mathbf{g}\Delta t_{n,n+1}). \quad (30)$$

Similarly, both sides of the Eq.(30) are multiplied by $\mathbf{R}^{l_0}_{l_n}$:

$$\bar{\mathbf{v}}_{l_n} = \bar{\mathbf{v}}_{l_{n+1}} - [\bar{\mathbf{R}}^{l_0}_{l_n}][\bar{\mathbf{R}}^b_l]^{-1}\Delta\hat{\mathbf{v}}_{n,n+1} - {}_{l_0}\mathbf{g}\Delta t_{n,n+1}. \quad (31)$$

Substituting Eq.(31) into the Eq.(28) yields:

$$\begin{aligned}&{}_{l_0}\mathbf{g}\Delta t^2_{n,n+1} + \frac{1}{2}\sum_{k=n}^{n+1} {}_{l_0}\mathbf{g}\Delta t^2\\ &= (\bar{\mathbf{t}}_{l_{n+1}} - \bar{\mathbf{t}}_{l_n}) - \bar{\mathbf{R}}^{l_0}_{l_n}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_{n,n+1}\\ &\quad - (\bar{\mathbf{v}}_{l_{n+1}} - [\bar{\mathbf{R}}^{l_0}_{l_n}][\bar{\mathbf{R}}^b_l]^{-1}\Delta\hat{\mathbf{v}}_{n,n+1})\Delta t_{n,n+1}, \quad (32)\end{aligned}$$

By combining Eq.(32) with Eq.(28), the above equation can be constructed as:

$$\begin{aligned}&{}_{l_0}\mathbf{g}\Delta t^2_{n,n+1} + \frac{1}{2}\sum_{k=n}^{n+1} {}_{l_0}\mathbf{g}\Delta t^2\\ &= (\bar{\mathbf{t}}_{l_{n+1}} - \bar{\mathbf{t}}_{l_n}) - \bar{\mathbf{R}}^{l_0}_{l_n}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_{n,n+1}\\ &\quad - (\underline{\bar{\mathbf{v}}_{l_{n+1}}} - [\bar{\mathbf{R}}^{l_0}_{l_n}][\bar{\mathbf{R}}^b_l]^{-1}\Delta\hat{\mathbf{v}}_{n,n+1})\underline{\Delta t_{n,n+1}},\end{aligned}$$

$$\begin{aligned}&\frac{1}{2}\sum_{k=n+1}^{n+2} {}_{l_0}\mathbf{g}\Delta t^2\\ &= (\bar{\mathbf{t}}_{l_{n+2}} - \bar{\mathbf{t}}_{l_{n+1}})\\ &\quad - \bar{\mathbf{R}}^{l_0}_{l_{n+1}}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_{n+1,n+2} - \underline{\bar{\mathbf{v}}_{l_{n+1}}\Delta t_{n+1,n+2}}. \quad (33)\end{aligned}$$

By eliminating the underlined term above and simplifying the subscript $(*)_{n,n+1} = (*)_1$, $(*)_{n+1,n+2} = (*)_2$, we obtain:

$$\begin{aligned}&\frac{1}{2}{}_{l_0}\mathbf{g}\left[(\sum_{k=n}^{n+1}\Delta t^2)\Delta t_2 - (\sum_{k=n+1}^{n+2}\Delta t^2)\Delta t_1 + 2(\Delta t_1)^2\Delta t_2\right]\\ &= \Delta\bar{\mathbf{t}}_1\Delta t_2 - \Delta\bar{\mathbf{t}}_2\Delta t_1 - [\Delta t_2\bar{\mathbf{R}}^{l_0}_{l_n}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_1\\ &\quad - \Delta t_1\bar{\mathbf{R}}^{l_0}_{l_{n+1}}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_2] + [\bar{\mathbf{R}}^{l_0}_{l_n}][\bar{\mathbf{R}}^b_l]^{-1}\Delta\hat{\mathbf{v}}_1\Delta t_1\Delta t_2. \quad (34)\end{aligned}$$

In summary, the gravity vector can be solved with a closed-form solution by solving Eq.(34).

C. VELOCITY INITIALIZATION AND GRAVITY REFINEMENT

The world gravity ${}_w\mathbf{g}$ can be estimated by transforming ${}_{l_0}\mathbf{g}$ to the world frame. To obtain a more accurate result, the known magnitude of world gravity is incorporated as a constraint. Since the direction and magnitude of ${}_w\mathbf{g}$ are known, gravity can be parameterized as:

$${}_w\hat{\mathbf{g}} = g\cdot{}_{l_0}\bar{\hat{\mathbf{g}}} + w_1\mathbf{b}_1 + w_2\mathbf{b}_2 \quad (35)$$

where $g = 9.81$ is the magnitude; ${}_{l_0}\bar{\hat{\mathbf{g}}}$ is the normalized vector of the estimated gravity; $\mathbf{b}_1$ and $\mathbf{b}_2$ are two orthogonal basis spanning the tangent plane. w_1 and w_2 are the corresponding coefficients towards $\mathbf{b}_1$ and $\mathbf{b}_2$. A pair of $\mathbf{b}_1$ and $\mathbf{b}_2$ can be easily determined by Gram-Schmidt orthogonalization algorithm. BY considering Eq.(31) with Eq.(28), and substituting Eq.(35) into them, we can obtain:

$$\begin{aligned}&\frac{1}{2}\sum_{k=n}^{n+1}[-\bar{w}_1\mathbf{b}_1 - \bar{w}_2\mathbf{b}_2]\Delta t^2 + \bar{\mathbf{v}}_{l_n}\Delta t_{n,n+1}\\ &= \bar{\mathbf{t}}_{l_{n+1}} - \bar{\mathbf{t}}_{l_n} - \bar{\mathbf{R}}^{l_0}_{l_n}[\bar{\mathbf{T}}^b_l]^{-1}\Delta\hat{\mathbf{t}}_{n,n+1} - \frac{1}{2}\sum_{k=n}^{n+1} {}_w\bar{\hat{\mathbf{g}}}\Delta t^2,\\ &[-\bar{w}_1\mathbf{b}_1 - \bar{w}_2\mathbf{b}_2]\Delta t_{n,n+1} - \bar{\mathbf{v}}_{l_{n+1}} + \bar{\mathbf{v}}_{l_n}\\ &= -[\bar{\mathbf{R}}^{l_0}_{l_n}][\bar{\mathbf{R}}^b_l]^{-1}\Delta\hat{\mathbf{v}}_{n,n+1} - {}_w\bar{\hat{\mathbf{g}}}\Delta t_{n,n+1}. \quad (36)\end{aligned}$$

where ${}_w\bar{\hat{\mathbf{g}}}$, $\bar{w}_1$, and $\bar{w}_2$ are all normalized by g. For the nth measurement, if we define the to-be-estimated state as $\mathbf{X}_n = \left[\mathbf{v}^\top_{l_n}, \mathbf{v}^\top_{l_{n+1}}, \bar{w}_1, \bar{w}_2\right]^\top$, the above equation can be rewritten as matrix format, and solved with the least square closed-form solution:

$$\begin{aligned}\mathbf{A}_n\mathbf{X}_n - \mathbf{b}_n &= 0;\\ \tilde{\mathbf{X}}_n &= \left(\mathbf{A}^\top_n\mathbf{A}_n\right)^{-1}\mathbf{A}^\top_n\mathbf{b}_n. \quad (37)\end{aligned}$$

Finally, after estimating the refined gravity and the initial velocity, we can obtain the rotation transformation from $\mathcal{F}_{l_0}$ to $\mathcal{F}_w$, which is the initial orientation of the IMU.

REFERENCES

[1] A. Motroni, A. Buffi, and P. Nepa, "A survey on indoor vehicle localization through RFID technology," *IEEE Access*, vol. 9, pp. 17921–17942, 2021.

[2] S. Kumar and S. K. Das, "Target detection and localization methods using compartmental model for internet of Things," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2234–2249, Jun. 2019.

[3] Z. Wang, J. Zhang, S. Chen, C. Yuan, J. Zhang, and J. Zhang, "Robust high accuracy visual-inertial-laser slam system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6636–6641.

[4] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys, "LIC-fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking," 2020, *arXiv:2008.07196*. [Online]. Available: https://arxiv.org/abs/2008.07196

[5] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, Dec. 2018.

[6] C. L. Gentil, T. Vidal-Calleja, and S. Huang, "IN2LAMA: INertial LiDAR localisation and MApping," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 6388–6394.

[7] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," 2020, *arXiv:2007.00258*. [Online]. Available: https://arxiv.org/abs/2007.00258

[8] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "R-LINS: A robocentric LiDAR-inertial state estimator for robust and efficient navigation," 2019, *arXiv:1907.02233*. [Online]. Available: https://arxiv.org/abs/1907.02233

[9] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-LiDAR-inertial odometry and mapping," 2020, *arXiv:2010.13150*. [Online]. Available: https://arxiv.org/abs/2010.13150

[10] T.-M. Nguyen, M. Cao, S. Yuan, Y. Lyu, T. H. Nguyen, and L. Xie, "LIRO: Tightly coupled LiDAR-inertia-ranging odometry," 2020, *arXiv:2010.13072*. [Online]. Available: http://arxiv.org/abs/2010.13072

[11] T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression," in *Proc. Robot., Sci. Syst.*, Berkeley, CA, USA, 2014.

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

[13] 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 system," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9851–9861, Nov. 2019.

[14] J. Wu, Y. Sun, M. Wang, and M. Liu, "Hand-eye calibration: 4-D procrustes analysis approach," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 2966–2981, Jun. 2020.

[15] 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.

[16] Z. Zhang, L. Zhang, and G.-Z. Yang, "A computationally efficient method for hand–eye calibration," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 10, pp. 1775–1787, 2017.

[17] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1021–1028, Apr. 2019.

[18] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3662–3669.

[19] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-LiDAR calibration: A targetless and structureless approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1556–1563, Apr. 2020.

[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. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 2149–2155.

[21] 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.*, Sep. 2014, pp. 459–465.

[22] 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.*, Nov. 2013, pp. 1280–1286.

[23] 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. (IROS)*, Jan. 2020, pp. 9968–9975.

[24] C. Campos, J. M. M. Montiel, and J. D. Tardós, "Inertial-only optimization for visual-inertial initialization," 2020, *arXiv:2003.05766*. [Online]. Available: https://arxiv.org/abs/2003.05766

[25] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4225–4232.

[26] C. Campos, M. J. MM, and J. D. Tardós, "Fast and robust initialization for visual-inertial slam," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 1288–1294.

[27] X. Yan, V. Indelman, and B. Boots, "Incremental sparse GP regression for continuous-time trajectory estimation and mapping," *Robot. Auto. Syst.*, vol. 87, pp. 120–132, Jan. 2017.

[28] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 88–100, Jun. 2015.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2016.

[30] J. Dong, B. Boots, and F. Dellaert, "Sparse Gaussian processes for continuous-time trajectory estimation on matrix lie groups," 2017, *arXiv:1705.06020*. [Online]. Available: https://arxiv.org/abs/1705.06020

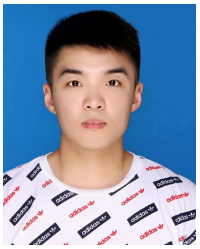
[31] J. Dong, M. Mukadam, B. Boots, and F. Dellaert, "Sparse Gaussian processes on matrix lie groups: A unified framework for optimizing continuous-time trajectories," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 6497–6504.

[32] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Auton. Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

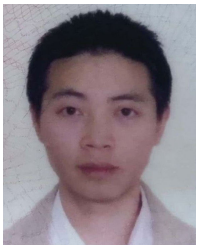
[33] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419841532.

[34] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "LIPS: LiDAR-inertial 3D plane SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 123–130.

[35] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, "A novel dual-LiDAR calibration algorithm using planar surfaces," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1499–1504.



SHUAIXIN LI received the bachelor's degree in surveying and mapping engineering from Central South University, in 2015, and the master's degree in control science and engineering from the PLA Information Engineering University, in 2018, where he is currently pursuing the Ph.D. degree with the College of Geospatial Information. His research interests include multisensor SLAM, mobile mapping, and automated driving.



LI WANG received the Ph.D. degree in surveying and mapping science from the PLA Information Engineering University, in 2014. He is currently a Lecturer with the College of Geospatial Information, PLA Information Engineering University. His research interests include point cloud data processing, mobile mapping, and 3D reconstruction.

JIUREN LI received the Ph.D. degree in control science and engineering from the National University of Defense Technology, in 2011. He is currently a Senior Engineer with Waytous Infinity Inc. His research interests include navigation, and control and automated driving.

BIN TIAN received the B.S. degree from Shandong University, Jinan, China, in 2009, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include automated driving, vision sensing and perception, and machine learning.

LONG CHEN (Senior Member, IEEE) received the Ph.D. degree from Wuhan University. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include computer vision, machine learning, and data mining with a focus on image and point cloud processing, stereo vision, and driver behavior analysis based on mobile devices.

GUANGYUN LI received the master's degree in surveying and mapping science from the PLA Institute of Surveying and Mapping, in 1987. He is currently a Professor with the College of Geospatial Information, PLA Information Engineering University. His research interests include precise engineering and industry measurement, navigation, and location services and applications.