Robust Real-time LiDAR-inertial Initialization

Fangcheng Zhu*, Yunfan Ren*, Fu Zhang

Abstract— For most LiDAR-inertial odometry, accurate initial states, including temporal offset and extrinsic transformation between LiDAR and 6-axis IMUs, play a significant role and are often considered as prerequisites. However, such information may not be always available in customized LiDAR-inertial systems. In this paper, we propose LI-Init: a full and real-time LiDAR-inertial system initialization process that calibrates the temporal offset and extrinsic parameter between LiDARs and IMUs, and also the gravity vector and IMU bias by aligning the state estimated from LiDAR measurements with that measured by IMU. We implement the proposed method as an initialization module, which can automatically detects the degree of excitation of the collected data and calibrate, on-the-fly, the temporal offset, extrinsic, gravity vector, and IMU bias, which are then used as high-quality initial state values for real-time LiDAR-inertial odometry systems. Experiments conducted with different types of LiDARs and LiDAR-inertial combinations show the robustness, adaptability and efficiency of our initialization method. The implementation of our LiDAR-inertial initialization procedure LI-Init and test data are open-sourced on Github[1] and also integrated into a state-of-the-art LiDAR-inertial odometry system FAST-LIO2.

I. INTRODUCTION

Sensors are called the eyes of robots, which endow them with capability of exploring surroundings and performing self-localization and navigation. Camera is a commonly used sensor due to the ability to provide rich RGB information with low cost and light weight, but it is vulnerable to inadequate illumination and is lack of direct depth measurement, leading to high computation complexity when reconstructing 3D environments. Compared with cameras, light detection and ranging (LiDAR) sensors can offer direct, accurate 3D measurements and is robust to illumination changes, making it a preferred choice for robot localization [1, 2] and mapping [3] applications.

To answer up the emergency such as sensor failure and to keep the whole system robust, multi-sensor fusion is becoming the main trend in recent years. Inertial Measurement Unit (IMU) is an excellent complementary sensor to fuse with camera or LiDAR since it can provide short-term ego-motion estimations without any external references. IMU is an ideal option to mitigate short-term odometry failure caused by degeneration, such as dim light scenes for cameras and structure-less environments for LiDARs. Moreover, high-frequency kinematic measurements from IMU help to compensate motion distortion of LiDAR scans especially when the robot is in high-speed motion [4]. More and more multi-sensor based simultaneous localization and mapping (SLAM) methods show up, including visual-inertial system

*These two authors contribute equally to this work.
F. Zhu, Y. Ren and F. Zhang are with Department of Mechanical Engineering, University of Hong Kong. {*zhufc,renyf*} *@connect.hku.hk*, *fuzhang@hku.hk*
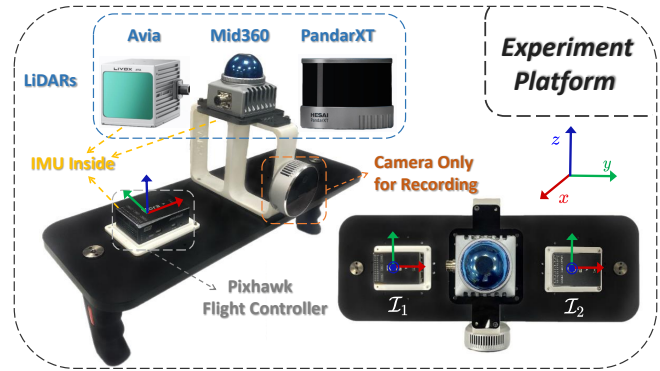[1]https://www.github.com/hku-mars/LiDAR_IMU_Init

Fig. 1. Experiment platform including multiple LiDARs (Non-repetitive scanning Livox Mid360, mechanical spinning Hesai PandarXT, small FoV Livox Avia) and built-in IMUs of LiDARs and Pixhawk flight controller. RealSense L515 camera is only used for recording videos in first personal view.

[5, 6], LiDAR-inertial system [1, 7, 8], and LiDAR inertial visual system [9, 10].

Owing to the strong non-linearity, the performance of sensor fusion system is heavily dependent on accurate initial states provided by efficient initialization module. Initialization of visual-inertial system has been widely studied [11, 12, 13], but few researches have focused on initialization for LiDAR-inertial system, which is necessary due to reasons below: 1) For self-assembled devices, the LiDAR and IMU are often not time-synchronized and with unknown extrinsic, necessitating extra, laborious temporal and spatial calibration in advance. 2) Points of a LiDAR scan are sampled at different instants, leading to inevitable motion distortion. In case the temporal offset is unknown, IMU aided motion distortion compensation methods adopted by [1, 14, 15] are no longer viable. 3) IMU raw measurements suffer from significant noises and the true values of linear accelerations and angular velocity are coupled with unknown bias. All these challenges drive us to find a well rounded LiDAR-inertial initialization method, capable of providing high-quality initial states including extrinsic transformation, gravity vector, IMU bias, and synchronizing the two sensors without any dedicated hardware setup.

Motivated by this, we propose a fast, robust LiDAR-inertial initialization method, which can automatically and accurately calibrate temporal offset and provide acceptable initial states without requiring any target or extra sensor, enabling a LiDAR-inertial odometry to run on a customized sensor setup without any dedicated prior calibration or hardware setup. Our contributions are highlighted as follows:

- We propose an efficient, accurate, hardware-free temporal calibration method based on cross-correlation and a unified temporal-spatial optimization, to estimate unknown but constant LiDAR-inertial temporal offset.

- We propose a novel optimization formulation to perform spatial initialization and a method to assess the degree of excitation in data. By further aligning states estimated from LiDAR with noise-mitigated IMU measurements, our initialization can automatically extract initialization data and estimate extrinsic transformation, gravity vector, gyroscope bias and accelerometer bias on the fly.
- We conduct experiments on multiple types of LiDARs and LiDAR-inertial combinations (see Fig. 1) to validate the efficiency and accuracy of our initialization procedure. As far as we know, the proposed method is the first open-sourced temporal and spatial initialization algorithm for 3D LiDAR-inertial system, supporting both mechanical spinning LiDARs and non-repetitive scanning LiDARs.

II. RELATED WORKS

There is a wide variety of initialization methods for visual-inertial systems. For example, an efficient IMU initialization method named VI-ORB-SLAM was introduced by Mur-Artal *et al*. [11]. The initialization problem is divided into three simple sub-problems and achieves high accuracy in a short time. Authors of [12] propose a robust initialization framework to recover the metric scale of monocular camera and to estimate extrinsic transformation and IMU bias. Huang *et al*. [13] propose a coarse-to-fine method to calibrate scale factor, gravity, and extrinsic transformation online. For the calibration of temporal offset between camera and inertial sensors, Mair *et al*. [16] propose a method based on cross-correlation and phase congruency analysis, and calibrate extrinsic rotation following standard hand-eye calibration. Qin *et al*. [17] propose an online method to calibrate temporal offset by jointly optimizing time offset, camera and IMU states.

In contrast to visual-inertial initialization, the initialization of LiDAR-inertial system is much less studied. Some of the existing initialization methods of LiDAR-inertial system require the sensors to be priorly synchronized, or rely on extra sensor, or ignore some initial states. Specifically, Wang *et al*. [18] present an online initialization method to estimate the temporal and spatial offset between LiDAR and IMU, but camera is needed as an extra auxiliary sensor. Similarly, GPS/GNSS are used in [19] for acquiring accurate position and attitude of the inertial sensor. Some LiDAR-inertial odometry systems have built-in initialization process, but these initialization modules are usually simple and incomplete. For example, [1] initializes the gyroscope bias, gravity vector, and temporal offset. But the initialization is fairly rough. For instance, the temporal offset is calibrated by assuming the sensor data receiving time as the sampling time, while data transmission and processing delay are totally neglected, leading to imprecise time offset estimation. The gyroscope bias in [1] is calibrated by keeping the sensor still for a few seconds in operation. Since gravity and accelerometer bias are coupled when the device stays still, accelerometer bias is not calibrated in its initialization. Although [1] calibrates the extrinsic online, the extrinsic initialization is not taken into account. So, good initial guess is required otherwise the convergence and robustness of the subsequent LiDAR-inertial odometry will be severely impacted. Similar to [1], initialization of [20] requires the device to stay still for a while. The accelerometer bias and LiDAR-inertial extrinsic parameters are obtained by prior offline calibration while the temporal offset is assumed to be priorly known. Compared to [18, 19], our proposed method does not require any extra sensor. Compared to [1, 20, 21], our work is more complete by initializing all the temporal offset, extrinsic, IMU bias, and gravity vector without any special requirements on the initial motion (*e.g*., keeping still) or any dedicated time synchronization or pre-calibration.

One of the main goals of our LiDAR-inertial initialization is to calibrate the extrinsic between LiDAR and IMU without any initial estimate. Some existing extrinsic calibration methods are based on batch optimization with tight data association, causing large time consumption. For example, Lv *et al*. [14] propose a continuous-time batch optimization based calibration. The usage of B-splines leads to more parameters to be estimated and would result in large computation cost. [15] uses an extended Kalman filter to estimate the extrinsic transformation with complicated motion compensation, which has limited convergence speed. Compared with these methods, our method is more lightweight, being able to run on the fly, while still achieving accurate extrinsic calibration sufficient for subsequent online estimation (*e.g*., by [1]). Our methods also calibrates the temporal offset that are not considered in [14, 15]. Besides, NDT based scan-to-scan matching adopted by [14, 15] usually does not work well for LiDARs with non-repetitive scanning pattern. In contrast, our method adopt scan-to-map matching strategy, which can be easily applied to both repetitive and non-repetitive scanning LiDARs.

III. METHODOLOGY

A. Framework Overview

Since IMU is only excited when it is in motion [15], our initialization procedure is a motion-based approach, which means sufficient excitation is necessary. The overview of our workflow is shown in Fig. 2 and some important notations are shown in Table I. The LiDAR odometry (see Section III-B) we propose is modified from FAST-LIO2 [1], by adopting a constant (both angular and linear) velocity (CV) model to predict the LiDAR motion and compensate the point distortion in a scan. To mitigate the mismatch between the constant velocity model and the actual sensor motion, the LiDAR odometry rate is increased by splitting an input frame into several sub-frames. If the LiDAR odometry does not fail (*e.g*., due to degeneration) and the estimated LiDAR angular and linear velocity satisfy our proposed assessment criterion (see Section III-C.5), the excitation is considered to be sufficient and both LiDAR odometry output and the corresponding IMU data are fed to the initialization module (see Section. III-C). In the initialization, the time offset is first calibrated by shifting IMU measurements to align with the LiDAR odometry, and then followed by an optimization process further refining the time offset, calibrating extrinsic transformation, and estimating IMU bias and gravity vector. The initialized states can be fed to a tightly-coupled LiDAR-inertial odometry (*e.g*., [1]) for online state estimation by fusing subsequent LiDAR and IMU data.
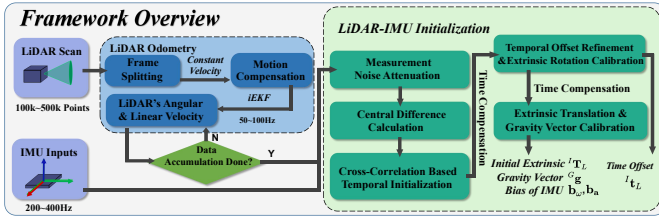
Fig. 2. Framework of our LiDAR-inertial initialization procedure.

TABLE I
SOME IMPORTANT NOTATIONS

Notation	Explanation
$\boxplus/\boxminus$	The encapsulated "boxplus" and "boxminus" operations on the state manifold.
t_k	Timestamp of the k-th LiDAR scan.
ρ_j	Timestamp of the j-th point in a LiDAR scan.
τ_i	Timestamp of the i-th IMU measurement.
L_j, L_k	The LiDAR body frame at the time ρ_j and t_k.
$\mathbf{x}, \widehat{\mathbf{x}}, \bar{\mathbf{x}}$	The ground-true, predicted, and updated state value.
$\check{\mathbf{x}}$	Estimation of $\mathbf{x}_j$ relative to $\mathbf{x}_k$ in backward propagation.
${}^I\mathbf{R}_L, {}^I\mathbf{p}_L$	The extrinsic rotation and translation from LiDAR to IMU.
${}^I t_L$	The total time offset between LiDAR and IMU.
$\mathbf{b}_{\boldsymbol{\omega}}, \mathbf{b}_{\mathbf{a}}$	The bias of gyroscope and accelerometer.
${}^G\mathbf{g}$	The gravity vector in global frame.
$\mathcal{I}_i, \mathcal{I}_k$	IMU data sequence used in initialization step with timestamp τ_i, t_k respectively.
$\bar{\mathcal{I}}_k$	IMU data sequence after compensating the initialized time offset, with synchronized timestamp t_k.
$\mathcal{L}_k$	LiDAR data used in initialization step with timestamp t_k.

B. LiDAR Odometry

Our LiDAR-only odometry and mapping is built on a constant velocity (CV) motion model, which assumes the angular and linear velocity are constant between two consecutive scans received at t_k and t_{k+1} respectively, *i.e.*,

$$\mathbf{x}_{k+1} = \mathbf{x}_k \boxplus (\Delta t \mathbf{f}(\mathbf{x}_k, \mathbf{w}_k)) \tag{1}$$

where Δt is the time interval between the two scans, the state vector $\mathbf{x}$, noise $\mathbf{w}$, and discrete state transition function $\mathbf{f}$ are defined as:

$$\mathbf{x} = \begin{bmatrix} {}^G\mathbf{R}_L \\ {}^G\mathbf{p}_L \\ {}^G\mathbf{v}_L \\ \boldsymbol{\omega}_L \end{bmatrix}, \mathbf{w} = \begin{bmatrix} \mathbf{n}_{\mathbf{v}} \\ \mathbf{n}_{\boldsymbol{\omega}} \end{bmatrix}, \mathbf{f}(\mathbf{x}, \mathbf{w}) = \begin{bmatrix} \boldsymbol{\omega}_L \\ {}^G\mathbf{v}_L \\ \mathbf{n}_{\mathbf{v}} \\ \mathbf{n}_{\boldsymbol{\omega}} \end{bmatrix} \tag{2}$$

where ${}^G\mathbf{R}_L \in SO(3), {}^G\mathbf{p}_L$ are the attitude and position of LiDAR in the global frame (here is the first LiDAR body frame L_0), ${}^G\mathbf{v}_L$ is LiDAR's linear velocity described in global frame, and $\boldsymbol{\omega}_L$ is LiDAR's angular velocity in LiDAR body frame, which are modelled as a random walk process driven by Gaussian noises $\mathbf{n}_{\mathbf{v}}$ and $\mathbf{n}_{\boldsymbol{\omega}}$, respectively. In (1), we used the notation $\boxplus/\boxminus$ defined in [22] to compactly represent the "plus" on the state manifold. Specifically, for the state manifold $SO(3) \times \mathbb{R}^n$ in (2), the $\boxplus$ operation and its inverse $\boxminus$ are defined as

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{a} \end{bmatrix} \boxplus \begin{bmatrix} \mathbf{r} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{R}\mathrm{Exp}(\mathbf{r}) \\ \mathbf{a}+\mathbf{b} \end{bmatrix}; \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{a} \end{bmatrix} \boxminus \begin{bmatrix} \mathbf{R}_2 \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathrm{Log}(\mathbf{R}_2^T\mathbf{R}_1) \\ \mathbf{a}-\mathbf{b} \end{bmatrix}$$

where $\mathbf{R}, \mathbf{R}_1, \mathbf{R}_2 \in SO(3), \mathbf{r}, \mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, $\mathrm{Exp}(\cdot) : \mathbb{R}^3 \mapsto SO(3)$ is the exponential map on $SO(3)$ [22] and $\mathrm{Log}(\cdot) : SO(3) \mapsto \mathbb{R}^3$ is its inverse logarithmic map.

In practice, the sensor motion may not have a constant velocity. To mitigate the effect of this model error, we can split an input LiDAR scan into multiple sub-frames of smaller duration, over which the sensor motion agrees more with the CV model.

1) Error State Iterated Kalman Filter: Based on the on-manifold system representation (1), we use an Error State Iterated Kalman Filter (ESIKF) [23] to estimate its states. The prediction step of the ESIKF consists of state prediction and covariance propagation as follows:

$$\widehat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k \boxplus (\Delta t \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{0})) \tag{3}$$

$$\widehat{\mathbf{P}}_{k+1} = \mathbf{F}_{\tilde{\mathbf{x}}}\bar{\mathbf{P}}_k\mathbf{F}_{\tilde{\mathbf{x}}}^T + \mathbf{F}_{\mathbf{w}}\mathbf{Q}\mathbf{F}_{\mathbf{w}}^T \tag{4}$$

where $\mathbf{P}, \mathbf{Q}$ are covariance matrix of state estimation and process noise $\mathbf{w}$ respectively. $\mathbf{F}_{\tilde{\mathbf{x}}}$ and $\mathbf{F}_{\mathbf{w}}$ are as follows:

$$\mathbf{F}_{\tilde{\mathbf{x}}} = \frac{\partial((\bar{\mathbf{x}}_k \boxplus \delta\mathbf{x}_k) \boxplus (\Delta t \mathbf{f}(\bar{\mathbf{x}}_k \boxplus \delta\mathbf{x}_k, \mathbf{0})) \boxminus (\bar{\mathbf{x}}_k \boxplus \Delta t \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{0})))}{\partial\delta\mathbf{x}_k}$$

$$= \begin{bmatrix} \mathrm{Exp}(-\widehat{\boldsymbol{\omega}}_{L_k}\Delta t) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\Delta t \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{I}_{3\times3}\Delta t & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}$$

$$\mathbf{F}_{\mathbf{w}} = \frac{\partial(\bar{\mathbf{x}}_k \boxplus (\Delta t \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{w}_k)) \boxminus (\bar{\mathbf{x}}_k \boxplus \Delta t \mathbf{f}(\bar{\mathbf{x}}_k, \mathbf{0})))}{\partial\mathbf{w}_k}$$

$$= \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3}\Delta t & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\Delta t \end{bmatrix} \tag{5}$$

where $\delta\mathbf{x}$ represents the error state.

2) Motion Compensation: In our considered problem, IMU and LiDAR are unsynchronized, hence IMU-aided motion compensation methods adopted by [14, 15] are not viable. After receiving a new LiDAR scan at timestamp t_{k+1}, to compensate the motion distortion, we project each contained point ${}^{L_j}\mathbf{p}_j$ sampled at timestamp $\rho_j \in (t_k, t_{k+1})$ into the scan-end LiDAR frame L_{k+1} as follows. With the constant velocity model, we have ${}^G\widehat{\mathbf{v}}_{L_{k+1}} = {}^G\bar{\mathbf{v}}_{L_k}$, $\widehat{\boldsymbol{\omega}}_{L_{k+1}} = \bar{\boldsymbol{\omega}}_{L_k}$, which leads to a relative transformation ${}^{L_{k+1}}\check{\mathbf{T}}_{L_j} = ({}^{L_{k+1}}\check{\mathbf{R}}_{L_j}, {}^{L_{k+1}}\check{\mathbf{p}}_{L_j})$ from time ρ_j to t_{k+1} as:

$$\begin{aligned} {}^{L_{k+1}}\check{\mathbf{R}}_{L_j} &= \mathrm{Exp}(-\widehat{\boldsymbol{\omega}}_{L_{k+1}}\Delta t_j) \\ {}^{L_{k+1}}\check{\mathbf{p}}_{L_j} &= -{}^G\widehat{\mathbf{R}}_{L_{k+1}}^T {}^G\widehat{\mathbf{v}}_{L_{k+1}}\Delta t_j \\ \Delta t_j &= t_{k+1} - \rho_j. \end{aligned} \tag{6}$$

Then the local measurement ${}^{L_j}\mathbf{p}_j$ can be projected to scan-end LiDAR frame as

$$ {}^{L_{k+1}}\mathbf{p}_j = {}^{L_{k+1}}\check{\mathbf{T}}_{L_j} {}^{L_j}\mathbf{p}_j \tag{7}$$

Then the distortion-compensated scan $\{{}^{L_{k+1}}\mathbf{p}_j\}$ provides an implicit measurement of the unknown state ${}^G\mathbf{T}_{L_{k+1}}$ expressed as the point-to-plane distance residual, based on which the full state $\mathbf{x}_{k+1}$ is iteratively estimated in an iterated Kalan filter framework until convergence. The converged state estimate, denoted as $\bar{\mathbf{x}}_{k+1}$, will then be
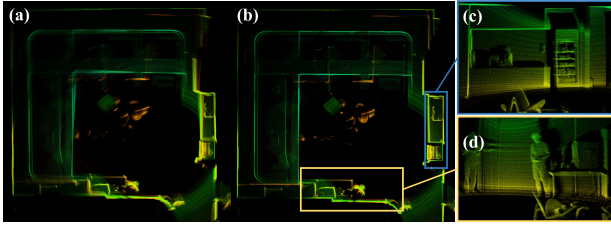
Fig. 3. The mapping result comparison using Livox Mid360 LiDAR. (a) Map without motion compensation. (b) Point cloud map using distortion compensated scans following (6) and (7). (c, d) Mapping details of (b)

used to propagate the subsequent IMU measurements as in Section III-B.1. Details of this iterative estimation can be referred to FAST-LIO2 [1] or [23] for a more general treatment of manifold constraints. The mapping comparison of our LiDAR odometry using scans with and without motion compensation is shown in Fig. 3.

C. LiDAR-inertial Initialization

The LiDAR odometry in Section III-B outputs the Li-DAR's angular velocity $\boldsymbol{\omega}_{L_k}$ and linear velocity $^G\mathbf{v}_{L_k}$ at each scan-end time with timestamp t_k. Meanwhile, IMU provides raw measurements, which are body angular velocity $\boldsymbol{\omega}_{m_i}$ and linear acceleration $\mathbf{a}_{m_i}$ with timestamp τ_i. These data are accumulated and repeatedly assessed by the excitation criterion shown in Section. III-C.5. Once data of sufficient excitation is collected, the initialization module is called, which eventually outputs time offset $^I t_L \in \mathbb{R}$, extrinsic $^I_I\mathbf{T}_L = (^I\mathbf{R}_L, ^I\mathbf{p}_L) \in SE(3)$, IMU bias $\mathbf{b}_\omega, \mathbf{b}_\mathbf{a} \in \mathbb{R}^3$, and gravity vector $^G\mathbf{g} \in \mathbb{R}^3$ in the global frame.

1) Data Preprocess: The IMU raw measurements suffer from noises $\mathbf{n}_{\omega_i}$ and $\mathbf{n}_{\mathbf{a}_i}$. The IMU measurement model is:

$$\boldsymbol{\omega}_{m_i} = \boldsymbol{\omega}_i^{\text{gt}} + \mathbf{b}_\omega + \mathbf{n}_{\omega_i}, \ \mathbf{a}_{m_i} = \mathbf{a}_i^{\text{gt}} + \mathbf{b}_\mathbf{a} + \mathbf{n}_{\mathbf{a}_i} \quad (8)$$

where $\boldsymbol{\omega}_i^{\text{gt}}, \mathbf{a}_i^{\text{gt}}$ are the ground-truth of IMU angular velocity and linear acceleration. Similarly, the estimations $\boldsymbol{\omega}_{L_k}, ^G\mathbf{v}_{L_k}$ from the LiDAR odometry contain noise as well.

To mitigate these noises, which are usually of high frequency, a non-causal zero phase low-pass filter [24] is used to filter the noise without introducing any filter delay. The zero phase filter is implemented by running a Butterworth low-pass filter forward and backward [24], producing noise-attenuated IMU measurements $\boldsymbol{\omega}_{I_i} = \boldsymbol{\omega}_i^{\text{gt}} + \mathbf{b}_\omega, \mathbf{a}_{I_i} = \mathbf{a}_i^{\text{gt}} + \mathbf{b}_\mathbf{a}$. The noise-attenuated LiDAR estimations are still denoted as $\boldsymbol{\omega}_{L_k}, ^G\mathbf{v}_{L_k}$ for notation simplicity.

From the LiDAR odometry $\boldsymbol{\omega}_{L_k}, ^G\mathbf{v}_{L_k}$, we obtain the LiDAR angular and linear accelerations $\boldsymbol{\Omega}_{L_k}, ^G\mathbf{a}_{L_k}$ by non-causal central difference [25]. The resultant LiDAR odometry data can be collectively denoted as

$$\mathcal{L}_k = \{\boldsymbol{\omega}_{L_k}, ^G\mathbf{v}_{L_k}, \boldsymbol{\Omega}_{L_k}, ^G\mathbf{a}_{L_k}\} \quad (9)$$

Similarly, we obtain the IMU angular acceleration $\boldsymbol{\Omega}_{I_i}$ from noise-attenuated gyroscope measurements $\boldsymbol{\omega}_{I_i}$, leading to:

$$\mathcal{I}_i = \{\boldsymbol{\omega}_{I_i}, \mathbf{a}_{I_i}, \boldsymbol{\Omega}_{I_i}\} \quad (10)$$

Since IMU frequency is usually higher than that of LiDAR odometry, the two sequence $\mathcal{I}_i$ and $\mathcal{L}_k$ are not of the same size. To fix this, we extract the LiDAR and IMU data received within the same time period, and down-sample $\mathcal{I}_i$ by linearly

interpolating it at each LiDAR odometry time t_k (see Fig. 4). The down-sampled IMU data is denoted as $\mathcal{I}_k$:

$$\mathcal{I}_k = \{\boldsymbol{\omega}_{I_k}, \mathbf{a}_{I_k}, \boldsymbol{\Omega}_{I_k}\} \quad (11)$$

which has the same timestamp t_k with $\mathcal{L}_k$ (but the data is really delayed by the known temporal constant $^I t_L$).
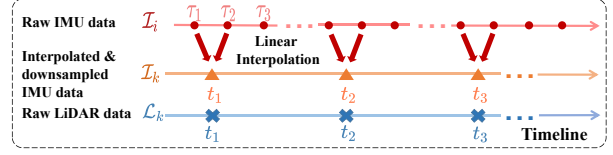

Fig. 4. Down sample IMU data by interpolating it at each LiDAR odometry timestamp.

2) Temporal Initialization by Cross-Correlation: In most cases, due to inevitable transmission and processing delay prior to its reception by LiDAR-inertial odometry module, an unknown but constant offset $^I t_L$ between the LiDAR $\mathcal{L}_k$ and IMU $\mathcal{I}_k$ will exist, such that the IMU measurement $\mathcal{I}_k$, if advanced by $^I t_L$, will be aligned with the LiDAR odometry $\mathcal{L}_k$. Since the LiDAR data (9) and IMU data (11) are at discrete times t_k, advancing the IMU data is essentially made in discrete steps $d = ^I t_L / \Delta t$, where Δt is the time interval between two LiDAR scans. Specifically, for the angular velocity, we have

$$\boldsymbol{\omega}_{I_{k+d}} = ^I\mathbf{R}_L\boldsymbol{\omega}_{L_k} + \mathbf{b}_\omega \quad (12)$$

Ignoring the gyroscope bias $\mathbf{b}_\omega$, which is usually small, we find that the magnitude of $\boldsymbol{\omega}_{I_{k+d}}$ and $\boldsymbol{\omega}_{L_k}$ should be the same, regardless of the extrinsic $^I\mathbf{R}_L$. Inspired by [16], we use the zero-centered cross-correlation to quantify the similarity between their magnitude. Then, the offset d can be solved from the following optimization problem

$$d^* = \arg\max_d \sum \|\boldsymbol{\omega}_{I_{k+d}}\| \cdot \|\boldsymbol{\omega}_{L_k}\| \quad (13)$$

by enumerating the offset d in the index range of $\mathcal{L}_k$.

3) Unified Extrinsic Rotation and Temporal Calibration: The cross-correlation method is robust against noise and small-scale gyroscope bias. But one obvious defect of (13) is that the calibration resolution of the temporal offset can only be made up to one sampling interval Δt of the LiDAR odometry, any residual offset δt smaller than Δt cannot be identified. Let $^I t_L$ be the total offset between LiDAR odometry $\boldsymbol{\omega}_L$ and IMU data $\boldsymbol{\omega}_I$, then $^I t_L = d^*\Delta t + \delta t$. Similar to (12), the IMU measurement $\boldsymbol{\omega}_I$, if advanced by time $^I t_L$, will be aligned with the LiDAR odometry $\boldsymbol{\omega}_L$:

$$\boldsymbol{\omega}_I(t + ^I t_L) = ^I\mathbf{R}_L\boldsymbol{\omega}_L(t) + \mathbf{b}_\omega \quad (14)$$

Since the actual LiDAR odometry $\boldsymbol{\omega}_L$ in (9) is only available at timestamps t_k, substituting $t = t_k$ and $^I t_L = d^*\Delta t + \delta t$ into (14) and noticing $\boldsymbol{\omega}_L(t_k) = \boldsymbol{\omega}_{L_k}$, we have

$$\boldsymbol{\omega}_I(t_k + d^*\Delta t + \delta t) = ^I\mathbf{R}_L\boldsymbol{\omega}_{L_k} + \mathbf{b}_\omega. \quad (15)$$

Notice that $\boldsymbol{\omega}_I(t_k + d^*\Delta t + \delta t)$ is the IMU angular velocity right after time $t_k + d^*\Delta t$, where the angular velocity and acceleration are $\boldsymbol{\omega}_I(t_k + d^*\Delta t) = \boldsymbol{\omega}_{I_{k'}}$ and $\boldsymbol{\Omega}_I(t_k + d^*\Delta t) = \boldsymbol{\Omega}_{I_{k'}}$, respectively, where $k' = k + d^*$. We can interpolate
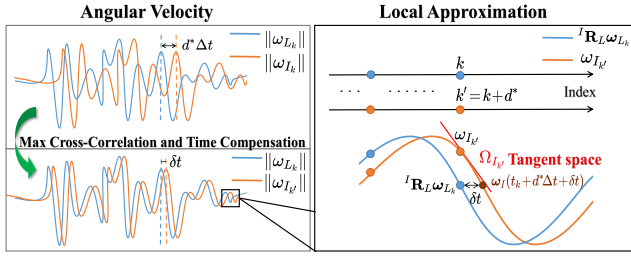
3951

Fig. 5. Illustration of time offset calibration and the first order approximation shown in Equation (16).

the value of $\omega_I(t_k + d^*\Delta t + \delta t)$ by assuming the angular acceleration is constant over the small δt (see Fig. 5):

$$\omega_I(t_k + d^*\Delta t + \delta t) \approx \omega_{I_{k'}} + \delta t \Omega_{I_{k'}} \qquad (16)$$

which can be substituted into (15) to obtain

$$\omega_{I_{k'}} + \delta t \Omega_{I_{k'}} = {}^I\mathbf{R}_L \omega_{L_k} + \mathbf{b}_\omega \qquad (17)$$

Finally, based on the constraint in (17), the unified temporal-spatial optimization problem can be stated as:

$$\underset{{}^I\mathbf{R}_L, \mathbf{b}_\omega, \delta t}{\arg\min} \sum \|{}^I\mathbf{R}_L \omega_{L_k} + \mathbf{b}_\omega - \omega_{I_{k'}} - \delta t \cdot \Omega_{I_{k'}}\|^2 \qquad (18)$$

which is solved iteratively (due to the nonlinear constraint ${}^I\mathbf{R}_L \in SO(3)$) by Ceres Solver[2] from an initial value of $({}^I\mathbf{R}_L, \mathbf{b}_\omega, \delta t) = (\mathbf{I}_{3\times3}, \mathbf{0}_{3\times1}, 0)$.

4) Extrinsic Translation and Gravity Initialization: In Section III-C.3, we obtained the extrinsic rotation ${}^I\mathbf{R}_L$, gyroscope bias $\mathbf{b}_\omega$ and the temporal offset ${}^I t_L$. In this section, we fix these values and proceed to the calibration of extrinsic translation, gravity vector, and acceleromter bias.

First, we align the IMU data $\mathcal{I}_k$ with that of LiDAR $\mathcal{L}_k$ using the offset d^* and residual δt previously calibrated. The aligned IMU data is denoted as $\bar{\mathcal{I}}_k$, which is now assumed to be perfectly aligned with $\mathcal{L}_k$ without time offset. Specifically, the IMU angular velocity $\bar{\omega}_{I_k}$ corresponding to LiDAR angular velocity ω_{L_k} at time t_k is (see (15))

$$\bar{\omega}_{I_k} = \omega_I(t_k + d^*\Delta t + \delta t) \approx \omega_{I_{k+d}} + \delta t \Omega_{I_{k+d}}, \qquad (19)$$

Similarly, the IMU acceleration $\bar{\mathbf{a}}_{I_k}$ corresponding to LiDAR acceleration ${}^G\mathbf{a}_{L_k}$ at time t_k is

$$\begin{aligned}\bar{\mathbf{a}}_{I_k} &= \mathbf{a}_I(t_k + d^*\Delta t + \delta t) \\ &\approx \mathbf{a}_{I_{k+d}} + \frac{\delta t}{\Delta t}(\mathbf{a}_{I_{k+d+1}} - \mathbf{a}_{I_{k+d}}),\end{aligned} \qquad (20)$$

Then, similar to (14), we can find the acceleration constraint between IMU and LiDAR. As marked in [26], the accelerations of two frames A, B with fixed extrinsic have the following relationship:

$$^A\mathbf{R}_B\mathbf{a}_B = \mathbf{a}_A + \lfloor\omega_A\rfloor^2_\wedge {}^A\mathbf{p}_B + \lfloor\Omega_A\rfloor_\wedge {}^A\mathbf{p}_B \qquad (21)$$

where ${}^A\mathbf{R}_B, {}^A\mathbf{p}_B$ represent the extrinsic transformation from frame B to frame A. Both $\mathbf{a}_A, \mathbf{a}_B$ are described in their own body frame.

For LiDAR-inertial system, we have two choices: A for IMU and B for LiDAR, or the opposite situation. Noticing

that in the first case the accuracy of $\omega_A = \bar{\omega}_{I_k} - \mathbf{b}_\omega$ is influenced by gyroscope bias estimation, and the error of Ω_A would be amplified due to the noise in angular velocity measurements. To avoid this problem and increase the robustness of extrinsic translation calibration, we set LiDAR as A and IMU as B. Since LiDAR's acceleration ${}^G\mathbf{a}_{L_k}$ is described in the global frame, we need to calculate LiDAR's instant acceleration described in body frame, denoted as $\mathbf{a}_{L_k}$:

$$\mathbf{a}_{L_k} = {}^G\mathbf{R}_L^T({}^G\mathbf{a}_{L_k} - {}^G\mathbf{g}) \qquad (22)$$

where ${}^G\mathbf{R}_L$ is the LiDAR's attitude in the global frame and is obtained from the LiDAR odometry in Section III-B.

Finally, the extrinsic translation, accelerometer bias, and gravity vector can be jointly estimated from the following optimization problem:

$$\underset{{}^I\mathbf{p}_L, \mathbf{b}_\mathbf{a}, {}^G\mathbf{g}}{\arg\min} \sum \|{}^I\mathbf{R}_L^T(\bar{\mathbf{a}}_{I_k} - \mathbf{b}_\mathbf{a}) - \mathbf{a}_{L_k} - (\lfloor\omega_{L_k}\rfloor^2_\wedge + \lfloor\Omega_{L_k}\rfloor_\wedge)^L\mathbf{p}_I\|^2 \qquad (23)$$

which can be solved iteratively (due to the constraint ${}^G\mathbf{g} \in \mathbb{S}_2$) by Ceres Solver from the initial value $({}^I\mathbf{p}_L, \mathbf{b}_\mathbf{a}, {}^G\mathbf{g}) = (\mathbf{0}_{3\times1}, \mathbf{0}_{3\times1}, 9.81\mathbf{e}_3)$. After ${}^L\mathbf{p}_I$ is estimated, the translation from LiDAR to IMU can be computed as ${}^I\mathbf{p}_L = -{}^I\mathbf{R}_L {}^L\mathbf{p}_I$.

5) Data Accumulation Assessment: The proposed initialization method relies on sufficient excitation (adequate motion) of LiDAR-inertial device. Thus, the system should be capable of assessing whether the excitation is sufficient to perform initialization all by itself. Ideally, the excitation can be assessed by the rank of the full Jacobian matrix of (18) for $({}^I\mathbf{R}_L, \mathbf{b}_\omega, \delta t)$ and (23) for $({}^I\mathbf{p}_L, \mathbf{b}_\mathbf{a}, {}^G\mathbf{g})$. In practice, we found that it is sufficient to assess the Jacobian w.r.t. the extrinsic rotation ${}^I\mathbf{R}_L$ and extrinsic translation ${}^I\mathbf{p}_L$ only, since excitation on the extrinsic usually require complicated motion that excite the other states as well. Denote $\mathbf{J}_r$ the Jacobian of (18) w.r.t. ${}^I\mathbf{R}_L$ and $\mathbf{J}_t$ the Jacobian of (23) w.r.t. ${}^I\mathbf{p}_L$,

$$\mathbf{J}_r = \begin{bmatrix} \vdots \\ -{}^I\mathbf{R}_L\lfloor\omega_{L_k}\rfloor_\wedge \\ \vdots \end{bmatrix}, \mathbf{J}_t = \begin{bmatrix} \vdots \\ \lfloor\omega_{L_k}\rfloor^2_\wedge + \lfloor\Omega_{L_k}\rfloor_\wedge \\ \vdots \end{bmatrix}. \qquad (24)$$

Then the excitation can be assessed from the rank of $\mathbf{J}_r^T\mathbf{J}_r = \sum\lfloor\omega_{L_k}\rfloor^T_\wedge\lfloor\omega_{L_k}\rfloor_\wedge$ and $\mathbf{J}_t^T\mathbf{J}_t = \sum(\lfloor\omega_{L_k}\rfloor^2_\wedge + \lfloor\Omega_{L_k}\rfloor_\wedge)^T(\lfloor\omega_{L_k}\rfloor^2_\wedge + \lfloor\Omega_{L_k}\rfloor_\wedge)$. More quantitatively, the extent of excitation is indicated by the singular values of $\mathbf{J}_r^T\mathbf{J}_r$ and $\mathbf{J}_t^T\mathbf{J}_t$. Based on this principle, we developed an assessment program that can instruct the users how to move their devices to obtain sufficient excitation. We quantify the excitation based on the singular values of the Jacobian matrix, and set a threshold to assess if the excitation is sufficient.

IV. EXPERIMENTS

We evaluate our initialization method mainly on datasets collected by our self-assembled LiDAR-inertial handheld setup (Fig. 1). We test our initialization algorithm with multiple types of LiDAR (Livox[3] Avia/Mid360 and Hesai

---

[2]http://ceres-solver.org/

[3]https://www.livoxtech.com

TABLE II

TEMPORAL INITIALIZATION RESULTS

LiDAR	$^I t_L$ (s)	**Mean**[s]	**RMSE**[s]	**NEES**
	0.05	0.0490	0.0016	3.2%
Avia	0.1	0.0988	0.0017	1.7%
	0.5	0.4989	0.0018	0.36%
	0.05	0.0479	0.0034	6.8%
Mid360	0.1	0.0982	0.0028	2.8%
	0.5	0.4984	0.0029	0.58%

Temporal initialization results of Livox LiDARs and their built-in IMUs. RMSE means root mean square error and NEES denotes normalized estimation error squared of calibrated time offset [17], computed as NEES = RMSE/$^I t_L$.

PandarXT[4]) and 6-axis IMUs (Bosch BMI088 inside both Pixhawk flight controller[5] and Livox LiDARs). The original data frequency of LiDAR is set as 10 Hz (10 scans per second), the frequency of IMU raw data is 200 Hz. All the experiments are conducted on a desktop computer with Intel i7-10700 @2.90 GHz with 32 GB RAM. In all experiments, the initial states are set to $(^I\mathbf{R}_L, \mathbf{b}_\omega, \delta t) = (\mathbf{I}_{3\times3}, \mathbf{0}_{3\times1}, 0)$ in (18) and $(^I\mathbf{p}_L, \mathbf{b_a}, ^G\mathbf{g}) = (\mathbf{0}_{3\times1}, \mathbf{0}_{3\times1}, 9.81\mathbf{e}_3)$ in (23). The initialization of other states does not need any initial values.

A. Temporal Initialization Evaluation

In order to show the effectiveness and accuracy of our temporal initialization (calibration) method, we test it on data collected by Livox LiDARs (Livox Avia and Livox Mid360) and their built-in IMUs. Since each Livox LiDAR and its built-in IMU are hardware synchronized in factory, the ground-true time offset is around 0 seconds, with microsecond accuracy level. To demonstrate the capability of our temporal calibration, we manually shift the input IMU timestamps to construct an artificial time offset [17], by adding a fixed value $^I t_L$ to IMU timestamps. We collect 5 data sequences in a laboratory scene using each Livox LiDAR and its built-in IMU. The calibration results are shown in Table II. As can be seen, the temporal calibration error is in microsecond level which suffices the requirements of most LiDAR-inertial sensor fusion algorithms. The results show our temporal calibration approach has great accuracy and consistency.

For unsynchronized self-assembled LiDAR-inertial system, the ground-true time offset is hard to obtain. An indirect way that can validate the effectiveness of our approach is to examine the performance of tightly-coupled LiDAR-Inertial-Odometry (LIO) with the calibrated time offset. The LIO we adopt is FAST-LIO2 [1], a state-of-the-art LiDAR inertial odometry that provides a temporal synchronization and state initialization internally. We test FAST-LIO2 with temporal offset calibrated by itself and by our approach, while keeping other state initialization unchanged with extrinsic from CAD reference, on data sequences collected by unsynchronized Hesai PandarXT LiDAR and Pixhawk IMU (Fig. 1). The localization and mapping results are shown in Fig. 6. It can be seen that the overall LIO performance with our
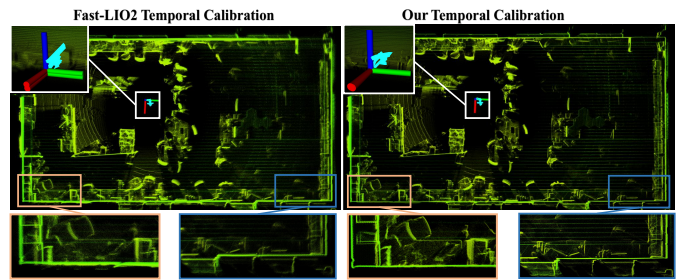
Fig. 6. Mapping result comparison (bird view) using FAST-LIO2 on the same dataset collected in laboratory. The device is waved back and forth to obtain sufficient excitation, and return back to the origin. The entire sequence lasts 55.8 seconds, where the first 20 seconds are used to calibrate temporal offset. We rerun FAST-LIO2 on the entire sequence with the calibrated offset. (a) Mapping result with time synchronization in FAST-LIO2, the map accuracy is limited by the inaccurate internal synchronization.

temporal calibration is better than that with the LIO's internal time synchronization. The odometry end-to-end drift is only 0.0102 m over a 11.3627 m trajectory for our method while 0.246 m for the LIO's internal time synchronization, the map with our approach also preserves more fine structural details.

B. Spatial Initialization Evaluation

1) Multiple LiDAR Types Test: We test our spatial (extrinsic) initialization method on data sequences acquired by platform shown in Fig. 1. As the LiDARs are covered with shell and the IMU is built in Pixhawk flight controller, their precise extrinsic are hard to know. However, we can validate our extrinsic initialization approach in an indirect way. We fix the Pixhawk at two poses $\mathcal{I}_1$ and $\mathcal{I}_2$ shown in Fig. 1, their ground-true relative pose is designed in CAD and ensured in actual assembly. The ground-true relative pose is (0, 0, 0) degrees in Euler angle for rotation and (0.25, 0, 0) meters for translation, the manufacturing accuracy level of CAD can achieve 0.01 degrees and millimeters. In each pose, the extrinsic of LiDAR and IMU is calibrated, denoted as $^{\mathcal{I}_1}\mathbf{T}_L$, $^{\mathcal{I}_2}\mathbf{T}_L$, respectively. The relative pose is then computed as $^{\mathcal{I}_1}\mathbf{T}_{\mathcal{I}_2} = {}^{\mathcal{I}_1}\mathbf{T}_L {}^{\mathcal{I}_2}\mathbf{T}_L^{-1}$, which can be compared with ground-truth provided by CAD reference. Aiming to prove that our calibration method supports multiple LiDARs with different scanning patterns, we collect 5 data sequences for each LiDAR, including Livox Mid360, Livox Avia, and Hesai PandarXT, with the Pixhawk IMU at both two poses.

In all experiments, we set initial extrinsic as $(^I\mathbf{R}_L, {}^I\mathbf{p}_L) = (\mathbf{I}_{3\times3}, \mathbf{0}_{3\times1})$, even though the ground-truth is far from the initial values (*e.g.*, for Mid360, IMU is at $\mathcal{I}_1$, the actual extrinsic is about $(0, -2, 178)$ degrees for rotation and $(0.12, 0, 0.11)$ meters for translation). We calculate the absolute values of the relative IMU pose errors, in average and standard deviation. The results are shown in Table III. As can be seen, the translation errors are in centimeter level, the rotational errors are less than $1°$. The overall small errors indicates the extrinsic initialization results are close across different datasets with the same sensor setup, and can be used as high-quality initial values for the subsequent LiDAR-inertial odometry.

Another interesting phenomenon from Table III is that, the extrinsic calibration errors of Livox Avia and Livox Mid360 are smaller than that of Hesai PandarXT. This mainly benefits

TABLE III

EXTRINSIC INITIALIZATION RESULTS

LiDAR		Livox Mid360	Livox Avia	PandarXT
Relative Error	Rot(°)	0.2472±0.2043	0.4019±0.1708	**0.7244±0.5076**
	Trans(m)	0.0081±0.0075	0.0064±0.0069	**0.0133±0.0102**

The mean value and SD (standard deviation) of extrinsic initialization.

TABLE IV

EXTRINSIC CALIBRATION COMPARISON

Method	Rotation(°)	Translation(m)	Time(s)
Proposed	**0.6208**	**0.0162**	**10.2**
LI-Calib [14]	1.0375	×	332.6
Target-Free [15]	0.8483	0.0187	115.7

The first 40 seconds of the dataset is used for calibration. × denotes the refinement of LI-Calib [14] fails and calibration result diverges due to ignorance of gravity vector initialization. So we use its rotation calibrated in the coarse calibration.

from the non-repetitive scanning of Livox LiDARs. Since our LiDAR odometry is based on constant velocity (CV) model, the input LiDAR frame is splitted into several subframes according to sampling time, to increase the odometry frequency and hence mitigate the CV model mismatch. For LiDARs with non-repetitive scanning, the frame splitting would not change the FOV of subframes. Also, the point map would get denser when the LiDAR stays still, which benefits our scan-to-map strategy. In contrast, for mechanical spinning LiDARs like Hesai PandarXT, the frame splitting would decrease the FOV of subframes and the robustness of LiDAR odometry. Thus, we can not move quickly when collecting data, which lowers the IMU excitation and the signal-to-noise ratio (SNR).

2) Accuracy and Robustness Comparison: To further verify the accuracy of our extrinsic initialization, we select the state-of-the-art LiDAR-inertial extrinsic calibration methods for comparison, which are LI-Calib [14] and Target-Free [15]. Since these two methods only support mechanical spinning LiDAR, we select two sequences collected by Hesai PandarXT and Pixhawk in Section IV-B.1, and use the same relative IMU pose error to assess the calibration accuracy. Fig. 7 (e) shows the scene of the two sequences. We use the first 40 seconds (including 400 LiDAR scans and corresponding IMU measurements) to run all calibration methods. The temporal offset is compensated in advance when testing LI-Calib and Target-Free since they are unable to do temporal calibration. The default parameters of the two methods are used on both sequences. The time-consumption and average relative IMU pose errors are shown in Table IV. As can be seen, our method is more accurate while consuming much less time. Fig. 7 shows some qualitative results. LI-Calib [14] adopts a coarse-to-refine calibration. In the coarse calibration, it registers LiDAR scans by NDT matching and calibrates the extrinsic rotation; in the refine stage, it performs batch optimization of all the extrinsic parameters. The map constructed in the coarse calibration is shown in Fig. 7 (a). The refinement of LI-Calib fails due to the ignorance of gravity initialization, leading to a messy
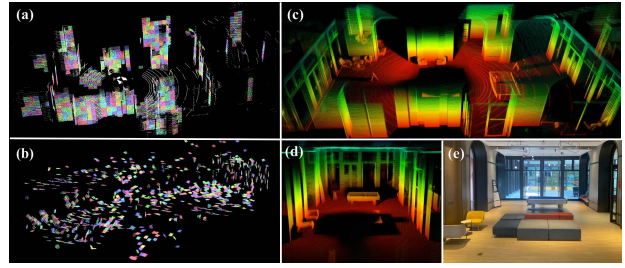

Fig. 7. (a) Surfel map constructed during the coarse calibration of LI-Calib [14]. (b) Refinement of LI-Calib fails, surfel map in a mess. (c,d) Accurate point cloud map constructed during our calibration, see Section III-B. (e) The calibration scene.

surfel map shown in Fig. 7 (b). In contrast, the LiDAR-only odometry in our calibration method is robust and accurate, the point map is shown in Fig. 7 (c,d). Target-Free [15] does not provide visualization module to show its map. These results show our method has better robustness than [14], and similar accuracy compared with [15], and much less computation cost than both [14, 15].

C. Time Consumption Evaluation

Compared with [14, 15], two extrinsic calibration methods with high computation load, which cannot be processed in real-time, our approach is fast and can be implemented in real-time. Different from minimizing point-to-plane with batch optimization, our LiDAR odometry is very efficient, the average processing time of a subframe is about 8 ms. Once sufficient data is collected, the total time consumption of initialization solver, including data pre-processing, temporal initialization, extrinsic and gravity initialization is less than 500 ms, which is far smaller than the actual data collection time.

To evaluate the efficiency, we test [14], [15] and our method on the same dataset collected in an apartment hallway, and compare the time consumption of motion compensation for each scan. Also, we compare the total calibration time when the data length, measured by LiDAR input size, is different. All the results are shown in Fig. 9, which suggests that our method has high computation efficiency. Moreover, as the input data amount increases, the processing time of our approach grows slowly.

D. Gravity and Bias Initialization Evaluation

Our method is able to calibrate gyroscope bias, accelerometer bias and gravity vector, which can be used as high-quality initial states for real-time LiDAR-inertial odometry system. To examine the accuracy of our method, we integrate it into FAST-LIO2 [1], which further refines all the states, including bias and gravity vector by tightly fusing the subsequent LiDAR data with IMU. We plot in Fig. 8 the difference between the states estimated online by FAST-LIO2 and their initial values supplied by our initialization. As can be seen, initial gyroscope bias, accelerometer bias and gravity vector are already very accurate and the subsequent refinement are small.

V. CONCLUSION

This paper proposes a fast, robust, temporal and spatial initialization method for LiDAR-inertial system. An accurate and efficient coarse-to-fine temporal calibration method is
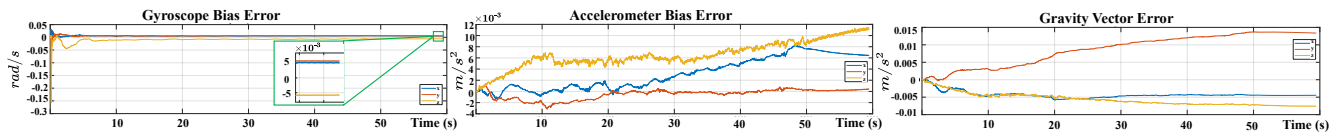
Fig. 8. Error of gyroscope bias, accelerometer bias and gravity vector refined by FAST-LIO2 and their initial values supplied by our initialization. Test dataset is collected by Livox Mid360 LiDAR and IMU inside Pixhawk flight controller.
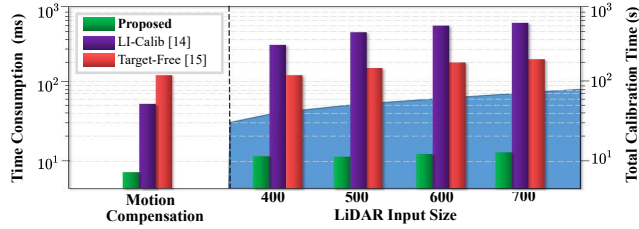

Fig. 9. Time consumption comparison. All test data are collected by Hesai PandarXT LiDAR with 10 Hz output. The blue area is where the initialization time is below the data collection time, so it can run in real-time.

proposed for unsynchronized LiDAR and IMU which is independent of any hardware synchronization setup. Also, we propose a fast, novel data association function to initialize LiDAR-inertial extrinsic transformation, gravity vector, as well as the bias of gyroscope and accelerometer. Various experiments show the consistency, robustness and high quality of our initialization method. Moreover, experiments using multiple types of LiDAR demonstrate the applicability to LiDARs with different scanning patterns.

VI. ACKNOWLEDGEMENT

REFERENCES

[1] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 2022.

[2] Jiarong Lin and Fu Zhang. Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131. IEEE, 2020.

[3] Zheng Liu and Fu Zhang. Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021.

[4] Yunfan Ren, Fangcheng Zhu, Wenyi Liu, Zhepei Wang, Yi Lin, Fei Gao, and Fu Zhang. Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors. *arXiv preprint arXiv:2202.12177*, 2022.

[5] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016.

[6] Michael Bloesch, Michael Burri, Sammy Omari, Marco Hutter, and Roland Siegwart. Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback. *The International Journal of Robotics Research*, 36(10):1053–1072, 2017.

[7] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020.

[8] Kailai Li, Meng Li, and Uwe D Hanebeck. Towards high-performance solid-state-lidar-inertial odometry and mapping. *IEEE Robotics and Automation Letters*, 6(3):5167–5174, 2021.

[9] Jiarong Lin, Chunran Zheng, Wei Xu, and Fu Zhang. R^2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping. *IEEE Robotics and Automation Letters*, 6(4):7469–7476, 2021.

[10] Jiarong Lin and Fu Zhang. R^3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10672–10678. IEEE, 2022.

[11] Raúl Mur-Artal and Juan D Tardós. Visual-inertial monocular slam with map reuse. *IEEE Robotics and Automation Letters*, 2(2):796–803, 2017.

[12] Tong Qin and Shaojie Shen. Robust initialization of monocular visual-inertial estimation on aerial robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4225–4232. IEEE, 2017.

[13] Weibo Huang and Hong Liu. Online initialization and automatic camera-imu extrinsic calibration for monocular visual-inertial slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5182–5189. IEEE, 2018.

[14] Jiajun Lv, Jinhong Xu, Kewei Hu, Yong Liu, and Xingxing Zuo. Targetless calibration of lidar-imu system based on continuous-time batch estimation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9968–9975. IEEE, 2020.

[15] Subodh Mishra, Gaurav Pandey, and Srikanth Saripalli. Target-free extrinsic calibration of a 3d-lidar and an imu. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 1–7. IEEE, 2021.

[16] Elmar Mair, Michael Fleps, Michael Suppa, and Darius Burschka. Spatio-temporal initialization for imu to camera registration. In *2011 IEEE International Conference on Robotics and Biomimetics*, pages 557–564. IEEE, 2011.

[17] Tong Qin and Shaojie Shen. Online temporal calibration for monocular visual-inertial systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3662–3669. IEEE, 2018.

[18] Yan Wang and Hongwei Ma. Online spatial and temporal initialization for a monocular visual-inertial-lidar system. *IEEE Sensors Journal*, 2021.

[19] Zachary Taylor and Juan Nieto. Motion-based calibration of multimodal sensor arrays. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4843–4850. IEEE, 2015.

[20] Chao Qin, Haoyang Ye, Christian E Pranata, Jun Han, Shuyang Zhang, and Ming Liu. Lins: A lidar-inertial state estimator for robust and efficient navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8899–8906. IEEE, 2020.

[21] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

[22] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Hertzberg, christoph and wagner, rené and frese, udo and schröder, lutz. *Information Fusion*, 14(1):57–77, 2013.

[23] Dongjiao He, Wei Xu, and Fu Zhang. Embedding manifold structures into kalman filters. *arXiv preprint arXiv:2010.05957*, 2021.

[24] Fredrik Gustafsson. Determining the initial states in forward-backward filtering. *IEEE Transactions on signal processing*, 44(4):988–992, 1996.

[25] Dumitru Baleanu, Ozlem Defterli, and Om P. Agrawal. A central difference numerical scheme for fractional optimal control problems. *Journal of Vibration and Control*, 15(4):583–597, 2009.

[26] Wei Xu, Dongjiao He, Yixi Cai, and Fu Zhang. Robots' state estimation and observability analysis based on statistical motion models. *IEEE Transactions on Control Systems Technology*, 2022.