Ct-LVI: A Framework Towards Continuous-time Laser-Visual-Inertial Odometry and Mapping

Zhong Wang, Lin Zhang, *Senior Member, IEEE,* Shengjie Zhao, *Senior Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—Owing to the inherent complementarity among LiDAR, camera, and IMU, a growing effort has been paid to laser-visual-inertial SLAM recently. The existing approaches, however, are limited in two aspects. First, at the front-end, they usually employ a discrete-time representation that requires high-precision hardware/software synchronization and are based on geometric laser features, leading to low robustness and scalability. Second, at the backend, visual loop constraints suffer from scale ambiguity and the sparseness of the point cloud deteriorates the scan-to-scan loop detection. To solve these problems, for the front-end, we propose a continuous-time laser-visual-inertial odometry which formulates the carrier trajectory in continuous time, organizes point clouds in probabilistic submaps, and jointly optimizes the loss terms of laser anchors, visual reprojections, and IMU readings, achieving accurate pose estimation even with fast motion or in unstructured scenes where it is difficult to extract meaningful geometric features. At the backend, we propose building 5-DoF laser constraints by matching projected 2D submaps and 6-DoF visual constraints via laser-aided visual relocalization, ensuring mapping consistency in large-scale scenes. Results show that our framework achieves high-precision estimation and is more robust than its counterparts when the carrier works in large scenes or with fast motion. The relevant codes and data are open-sourced at https://cslinzhang.github.io/Ct-LVI/Ct-LVI.html.**

***Index Terms*—Laser-Visual-Inertial Odometry, SLAM, Loop Detection, Data Fusion**

I. Introduction

FOR agents like unmanned aerial vehicles and mobile robots, real-time accurate estimation of their positions in the environment is a prerequisite for intelligent applications. In open outdoors, this task is usually fulfilled by GNSS systems (GPS etc.). However, in indoor environments, clustered parks, or high-rise blocks, the instability of satellite signals makes such systems no longer available. At this time, agents often seek onboard sensors to sense the environment and determine their positions simultaneously, which is called a Simultaneous Localization and Mapping (SLAM) problem. LiDAR, camera, and IMU are the three most common types of sensors used to build such SLAM systems [1]–[5]. Due to the natural complementarity among these three types of sensors, recent years have witnessed a research upsurge of laser-visual-inertial SLAM (LVI-SLAM) [6]–[11].

The early LVI-SLAM schemes fuse the multi-sensor data in a loosely-coupled way, for example, using laser scan to provide depth to visual points, or employing IMU to predict a rough transformation for inter-frame matching [6], [12], [13]. Unlike loosely-coupled ones, in addition to data association, recent methods also make a tightly-coupled estimation of sensor measurements, such as jointly optimizing carrier poses, visual/laser features, and IMU biases in the visual-inertial/laser-inertial subsystems [7]–[11]. Relatively speaking, the latter ones deeply fuse the multi-sensor data and thereby obtain more accurate results. However, they are limited in scalability and robustness either in their front-ends or backends due to the following reasons.

Their front-ends often encounter two critical challenges. Firstly, ensuring the time synchronization of incoming data from different sources is an inevitable issue. One option is to synchronize the time of the three sensors in hardware, such as some visual-inertial modules or some LiDARs equipped with built-in IMUs [14], [15] However, currently, there are no hardware solutions available in the market to synchronize the three. Alternatively, one way is to seek soft synchronization, for example, to find the nearest neighbor time or perform interpolation approximation. It is worth thinking that when the number of sensors to be fused increases (for example, self-driving cars will carry several LiDARs or cameras), either the hardware or the soft synchronization will be too cumbersome. Therefore, it is imperative to explore more feasible and rational time synchronization schemes. Secondly, existing research predominantly relies on the extraction of line or plane features from point clouds to facilitate laser data association and achieve precise scan registration, as seen in methods like LOAM [16]. However, such a registration technique is subject to two limitations. a) Its feature extraction is coupled with the scanning pattern of LiDAR, making it hard to support multi-LiDAR inputs. b) Feature-based methods are sensitive to noise and therefore do not perform well when the carrier works with fast motion or in unstructured scenes with few

This work was supported in part by the National Key Research and Development Project under Grant 2020YFB2103900; in part by the National Natural Science Foundation of China under Grant 62272343, Grant 61973235, and Grant 61972285; in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400; in part by the Shuguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 21SG23; and in part by the Fundamental Research Funds for the Central Universities. (*Corresponding author: Lin Zhang.*)

Zhong Wang, Lin Zhang, and Shengjie Zhao are with the School of Software Engineering, Tongji University, Shanghai 201804, China (email: {2010194, cslinzhang, shengjiezhao}@tongji.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@um.edu.mo).

geometric features.

The backend of a SLAM system usually resorts to loop detection technologies to eliminate accumulated drifts. For visual data, the loop detection based on bag of words has become a golden standard, while the scale of pure visual loop constraint is left with ambiguity [17], [18]. One possible remedy for this issue is to utilize IMU to capture scale information [19], but how to ensure long-term scale consistency is still an open problem. Compared with images, it is more difficult to extract stable position features from sparse laser point clouds. With the aid of image processing and deep learning technologies, scan-to-scan loop detection has developed to some extent in recent years [20], [21]. However, it is still difficult to detect the loop when there is a big difference in the pose between the two scans. Although assisting laser constraint construction with the help of visual information is an easy-to-think-of strategy, considering loop detection only while ignoring loop constraint construction still cannot effectively eliminate the cumulative error [22]. Therefore, how to skillfully fuse the laser-visual data so that they can complement each other both in loop detection and constraint construction is a key problem that needs to be solved.

To deal with the aforementioned problems, for the front-end, we propose a **C**ontinuous-**t**ime **L**aser-**V**isual-**I**nertial **O**dometry, **Ct-LVIO** for short. Specifically, considering the time asynchrony among different sensors, we formulate the carrier trajectory in a continuous-time representation, making the constraints between the data at any time and the trajectory conveniently established. To avoid the feature extraction of point clouds and support the multi-LiDAR inputs with arbitrary scanning patterns, laser data is organized and associated with probability submaps. Further, the front-end odometry is modeled as a Maximum A Posterior estimation problem and the loss terms among the continuous-time trajectory, the incoming laser scans, the IMU readings, and the camera images are specially designed and jointly optimized in a local time window, enabling high-precision pose estimation even when the carrier works in unstructured scenes or moves intensely. For the backend, considering the sparseness of point clouds and the scale ambiguity of visual loop constraints, we propose a loop detection and constraint construction strategy that integrates the projected laser submap with visual information. First, rough 5-DoF laser loop constraints are built from the submap-to-submap matching and 6-DoF visual loop constraints are constructed via the local visual bundle adjustment, respectively. Afterwards, these initial values are refined by the precise scan-to-submap registration, so as to further construct the global pose graph and eliminate the accumulated errors in time. Hereinafter, the complete framework with the front-end Ct-LVIO and the backend optimization will be referred to as **Ct-LVI**.

To verify the effectiveness of Ct-LVI, we developed a handheld device that consists of a multi-beam LiDAR and a binocular camera with a built-in IMU (as shown in Fig. 1(a)) and collected a wealth of real-world data, including sequences from common structured scenes, complex unstructured areas, as well as those gathered with aggressive maneuvers. Experimental results show that Ct-LVI can produce accurate maps whether in structured environments or not. Moreover, when mapping in large scenes and when the carrier is moving intensely, Ct-LVI performs much better than its state-of-the-art counterparts.

To summarize, our contributions are threefold:

1) We are the first to fully merge the merits of the continuous-time trajectory and probabilistic submap representation, yielding Ct-LVIO which enables the tightly coupled fusion of time-unsynchronized laser-visual-inertial data and supporting multi-LiDAR inputs with any scanning patterns. Ct-LVIO jointly optimizes the loss terms of laser anchors, visual reprojections, and IMU readings regarding the continuous-time trajectory, enabling high-precision pose estimation even in unstructured scenes or with fast motion.
2) At the backend, we propose a strategy that integrates the projected laser submaps with visual information to detect loop closures and construct global constraints. The submap-based loop detection makes full use of the place features from located multi-frame point clouds, overcoming the degradation problem caused by the sparsity of a single scan. Also, the laser-aided visual constraint brings the place features from dense visual data and the absolute scale information provided by LiDAR into full play.
3) We developed a handheld device and gathered a challenging real-world dataset for LVI-SLAM evaluation. To ensure the reproducibility of all our results and facilitate related extended studies, all the relevant data and codes are made publicly available at https://cslinzhang.github.io/Ct-LVI/Ct-LVI.html.

The remainder of this paper is organized as follows. Sec. II introduces related studies. Details of the proposed continous-time laser-visual-inertial SLAM framework are presented in Sec. III. Experimental results are reported in Sec. IV. Finally, Sec. V concludes the paper.

II. Related Work

In this part, we first review the discrete-time SLAM frameworks which are closely relevant to our work from the perspective of data fusion manners and afterward review the continuous-time SLAM systems.

A. Discrete-time Laser-Visual-Inertial SLAM

Loosely Coupled. Zhang *et al.* [12] associated and estimated the depths of visual features from point clouds, which improved the speed and accuracy of the visual odometry. Likewise, in [13], laser odometry with higher accuracy and stronger robustness was fused to reduce the drift of visual odometry. Further, based on their previous research on laser-visual fusion [12], [13], Zhang *et al.* [6] utilized IMU data to improve the SLAM performance in fast motion, resulting in a laser-visual-inertial SLAM framework, in which the state was estimated from coarse to fine hierarchically. Although this method advanced the laser-visual-inertial SLAM in part, its loosely coupled fusion led to the insufficient exploration of the internal relations of sensors.

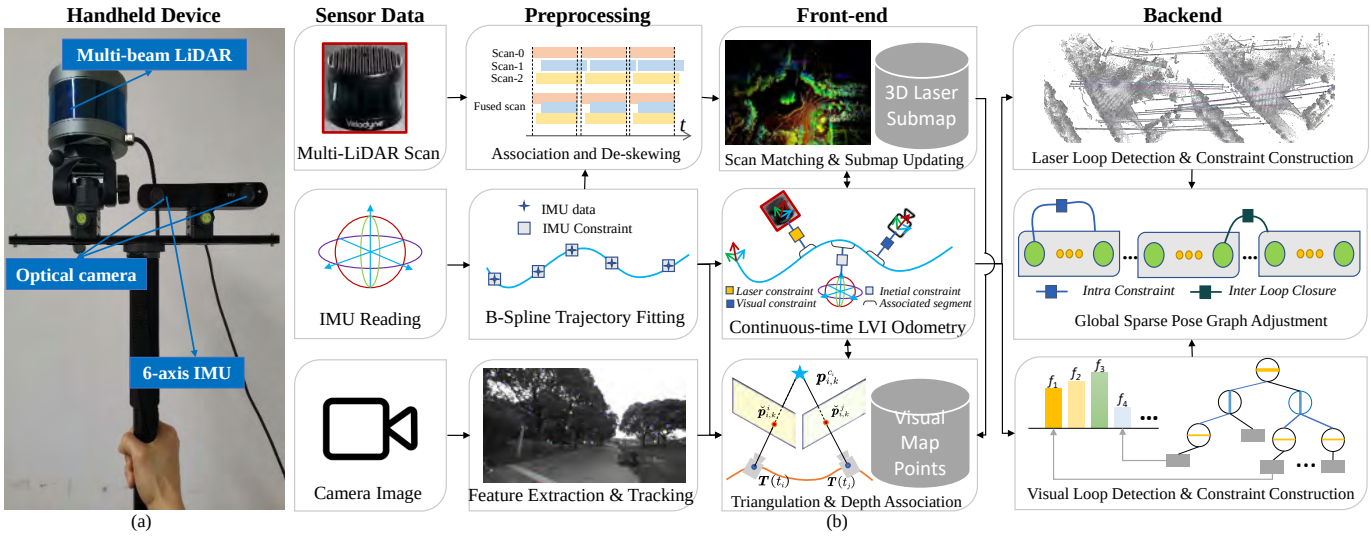

Fig. 1. (a) Self-developed handheld device. (b) The framework of Ct-LVI: "Preprocessing" produces synchronized and de-skewed point clouds, roughly fitted trajectory, and tracked visual features; "Front-end" conducts continuous-time laser-visual-inertial odometry; and "Backend" eliminates drifts via laser-visual fused loop detection and constraint construction, as well as global pose graph adjustment.

Tightly Coupled. In the past decade or so, SLAM research on visual-inertial fusion has made great progress. Relevant schemes fused visual and inertial information in a tightly coupled way with the help of Kalman filters [23] or joint optimization techniques [19]. In recent years, a line of studies migrated the techniques used in visual-inertial SLAM to laser-inertial SLAM. For example, based on Kalman filter, LINS [24], Fast-LIO [25], Fast-LIO2 [26], and Point-LIO [27] fused laser measurements with inertial data by carefully designing the state propagation and update mechanisms. In a different way, LIO-Mapping [28], LIO-SAM [29], and D-LIOM [30] conducted joint optimization to achieve tightly coupled state estimation.

Inspired by the mechanisms of visual-inertial and laser-inertial SLAM, researchers tried to conduct LVI-SLAM in a tightly coupled manner. According to the joint estimation methods adopted, tightly coupled LVI-SLAM frameworks can be classified into two types, filter-based ones and optimization-based ones. The filter-based schemes focused on the front-end odometry and mostly resorted to the Kalman filter to fuse data. For instance, based on the visual odometry framework MSCKF [23], Zuo *et al.* [7] proposed LIC-Fusion, in which laser measurement models of point-to-surface and point-to-line were constructed. Further, a more robust point-to-surface association mechanism was designed to boost LIC-Fusion, resulting in LIC-Fusion2 [8]. Recently, the error-state Kalman filter has shown a promising potential in multi-sensor fused SLAM. For example, Lin *et al.* [9] performed state prediction via IMU propagation and updated the state by constructing laser point-to-surface and visual reprojection measurement models. Built atop the same framework, R^{3}LIVE [10] employed color information to assist the state estimation of the visual-inertial subsystem, enabling colorful reconstruction. Unlike these filter-based methods, which usually focused on the current incoming data, a few approaches incorporated multi-frame data to estimate states. To fuse multi-frame information, a common way is to conduct joint optimization. For example, Shan *et al.* proposed LVI-SAM [11] based on the "smooth and mapping" framework [31], [32], which estimated the states of visual-inertial and laser-inertial subsystems separately and subsequently jointly optimized the results of visual-inertial odometry, IMU pre-integration, and laser-inertial odometry. Although LVI-SAM produced promising results, its system-wise fusion of visual-inertial and laser-inertial odometries limited its scalability and its location-triggered loop detection was vulnerable in practice. Very recently, Zheng *et al.* managed to deeply integrate the information from a visual-inertial subsystem and a laser-inertial subsystem, yielding a tightly-coupled and direct odometry framework, FAST-LIVO [33].

B. Continuous-time SLAM

The continuous-time trajectory representation was first applied to the extrinsic calibration of camera and IMU [34]. After that, based on Bayesian rule, Furgale *et al.* [35] took the lead in establishing a complete continuous-time SLAM theory and verified its effectiveness by the joint visual-inertial calibration [36]. Due to the high computational complexity, only a few scholars have tried to estimate the visual/laser odometry in the continuous-time SLAM framework until recent years. For instance, Mueggler *et al.* [37] adopted the continuous-time representation to develop visual-inertial odometry for event cameras, Mo *et al.* [38] employed the pose nodes from the visual odometry to optimize the continuous-time trajectory, and Lv *et al.* [39], [40] adjusted the trajectory by optimizing the geometric distances between the points and the corresponding surfaces. Recent relevant studies focused on improving the efficiency of optimizing continuous-time trajectories. For example, Sommer *et al.* [41] proposed a recursive formula for conveniently computing the derivatives of continuous-time B-splines with respect to time. In [42], Hug and Chli proposed a non-uniform continuous-time B-

spline split interpolation approach to improve the computation efficiency.

III. Methodology

A. Framework Overview

The overall framework of our Ct-LVI is illustrated in Fig. 1(b). In preprocessing, the incoming LiDAR, IMU, and camera data will be spatial-temporally associated and de-skewed, used to roughly fit the trajectory, and tracked by their features, respectively. In the front-end, the results of scan matching, the IMU readings, and the depth initialized visual features will be jointly optimized with the continuous-time trajectory, conducting sliding-window laser-visual-inertial odometry. At the backend, we conduct laser-visual fused loop detection and build a global sparse pose graph to eliminate the drift in time.

B. Notation and Trajectory Representation

Notation. A quantity in the world frame $\mathscr{F}_w$, the body frame (IMU frame) $\mathscr{F}_b$, the camera frame $\mathscr{F}_c$, and the LiDAR frame $\mathscr{F}_l$ are denoted by $(\cdot)^w$, $(\cdot)^b$, $(\cdot)^c$, and $(\cdot)^l$, respectively. A rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}(\det(\boldsymbol{R}) = 1)$, or a unit quaternion $\boldsymbol{q} = [q_w, \boldsymbol{q}_v^T]^T \in \mathbb{R}^4$ ($\boldsymbol{q}_v \in \mathbb{R}^3, \|\boldsymbol{q}\| = 1$) is indistinguishably utilized to denote a 3D rotation.

Trajectory Representation. In our case, on the one hand, it is cumbersome to establish the constraints among multiple discrete nodes for multi-source measurements that are not hardware synchronized. On the other hand, the frequencies of multi-source measurements are quite high (e.g., 400 Hz for the IMU and higher than 100,000 Hz for single-point acquisitions of the LiDAR). Therefore, we would like to have a unified representation of the trajectory, which facilitates the querying of the poses at arbitrary timestamps, the construction of laser-visual-inertial constraints, and the de-skewing of the LiDAR points. Also, since the high-frequency sensor data needs to be processed in real time, we hope that the trajectory representation has the merit of locality, that is, the update of the local trajectory will not affect the rest. Besides, to calculate the state conveniently, the trajectory representation should be analytically second-order derivable. To meet the abovementioned requirements, a B-spline with order $d+1$ is an ideal choice with such properties since it is a piecewise polynomial with degree d and is C^{d-1} continuous [43].

According to [43], for a given control point set $\{\boldsymbol{p}_i\} \in \mathbb{R}^3$ with size $N+1$ ($i, N \in Z^+$ and $0 \le i \le N$), its corresponding B-spline over knots domain $[t_0, t_{N+d+1})$ is defined by,

$$\boldsymbol{p}(t) = \sum_{i=0}^{N} B_{i,d}(t)\boldsymbol{p}_i, \tag{1}$$

where $B_{i,d}(t)$ are the B-spline basis functions which are given by Cox-de Boor recursion formula,

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t \in [t_i, t_{i+1}), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

$$B_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(t). \tag{3}$$

By denoting,

$$\tilde{B}_{i,d}(t) = \sum_{s=i}^{N} B_{s,d}(t), \tag{4}$$

Eq. 1 can also be given in a cumulative form,

$$\boldsymbol{p}(t) = \tilde{B}_{0,d}(t)\boldsymbol{p}_0 + \sum_{i=1}^{N} \tilde{B}_{i,d}(t)(\boldsymbol{p}_i - \boldsymbol{p}_{i-1}). \tag{5}$$

Analogically, for a given control quaternion set $\{\boldsymbol{q}_i\}$, $0 \le i \le N$, in the special orthogonal group $\mathbb{SO}(3)$, its corresponding B-spline is defined by [44],

$$\boldsymbol{q}(t) = \boldsymbol{q}_0^{\tilde{B}_{0,d}(t)} \otimes \prod_{i=1}^{N} \exp(\log(\boldsymbol{q}_{i-1}^{*} \otimes \boldsymbol{q}_i)\tilde{B}_{i,d}(t)), \tag{6}$$

where $\boldsymbol{q}_{i-1}^{*}$ is the conjugate quaternion of $\boldsymbol{q}_{i-1}$, $\otimes$ means the quaternion multiplication, $\exp(\cdot)$ maps an element in $\mathfrak{so}(3)$ (the Lie algebra of $\mathbb{SO}(3)$) to $\mathbb{SO}(3)$, and $\log(\cdot)$ is the inverse operator of $\exp(\cdot)$.

In our framework, the individual sensors do not need to be hardware time-synchronized. Their measurements will be associated with two continuous-time B-splines with time as the variable. One is the position trajectory, the other is the rotation trajectory. At any timestamp, the carrier's pose can be queried from these trajectories, thus obtaining the sensor's observation pose using sensor-to-sensor extrinsics. In this way, the objective to be optimized is transformed into the local segments of these trajectories instead of a discrete-time pose.

C. System Initialization

Reasonable initialization is necessary for a laser-visual-inertial SLAM system. On the one hand, since the IMU acceleration reading is coupled with gravity, the direction of gravity needs to be determined via initialization, so that the estimated trajectory is consistent with the actual physical movement of the carrier. On the other hand, as there are many variables to be estimated, we need to estimate the carrier state at the beginning reasonably, so as to ensure the quick convergence of the state estimator.

Via experiments, we find that the laser odometry is with higher accuracy and stability than the visual one when the carrier works in slow motion. Therefore, we extend the initialization of laser-inertial odometry in discrete-time representation to that in continuous-time. Specifically, for the multi-frame point clouds and IMU readings in the time window, we first resort to Normal Distribution Transform (NDT) [45] to obtain the relative motion of the multi-frame point clouds in a short-period time window, and at the same time obtain the IMU pre-integration values resorting to Forster's theory [46]. Subsequently, the relative state of the laser odometry is combined with the corresponding state quantity of the IMU pre-integration to obtain the gravity and velocities, so as to align the LiDAR poses to the world coordinate system according to the gravity. After that, the continuous-time trajectory in the initialization window is fitted, employing the discrete-time states as its control points.

D. Preprocessing

1) Trajectory Fitting with IMU Readings: For the incoming IMU data, we begin with the last estimated state and integrate the IMU readings to propagate state anchors for a specific interval. By this way, the $\mathbb{R}^3$ and $\mathbb{SO}(3)$ B-spline estimation can be converted into a curve fitting problem according to Eq. 5 and Eq. 6. The error terms of such a curve fitting problem are composed of two parts. One is the position-related (orientation-related) term stemming from the trajectory and the integration, and the other is the acceleration-related (angular rate-related) term derived from the trajectory and the raw IMU readings.

2) LiDAR Data Synchronization and De-skewing: A beam of point clouds is scanned point by point by a laser emitter. If the scanning process of a beam of point clouds is accompanied by the movement of the carrier, the sampling pose of each point will be different. Therefore, if a beam of point clouds is processed according to the same time and pose, the point clouds will be distorted. In today's LiDAR, each laser point has its corresponding time stamp, which makes it possible to correct the distortion according to the movement of the carrier, thereby improving the accuracy of registration and mapping.

Assuming that there are one primary LiDAR and several auxiliary LiDARs, we regard the starting and ending time stamps of the incoming scan from the primary LiDAR as a reference, and merge all the cached points from the auxiliary LiDARs whose time stamps are in the reference time interval to obtain the fused point cloud in a chronological order (as shown in "'Association and De-skewing" of Fig. 1(b)). Subsequently, according to the extrinsics among LiDARs, all the auxiliary points are first transformed to the primary LiDAR frame $\mathscr{F}_{l_p}$. To further remove the distortion, we retrieve the sampling pose of each point from the fitted trajectory and transform all points to the world frame $\mathscr{F}_w$. That is, a point $\boldsymbol{p}_t^{l_a}$ with a sampling time t in the auxiliary LiDAR frame $\mathscr{F}_{l_a}$ will be transformed to $\mathscr{F}_w$ by, $\mathring{\boldsymbol{p}}_t^w = \boldsymbol{T}_b^w \boldsymbol{T}_l^b \boldsymbol{T}_{l_a}^l \mathring{\boldsymbol{p}}_t^{l_a}$, where $\mathring{()}$ returns the corresponding homogeneous coordinate; $\boldsymbol{T}_b^w$, $\boldsymbol{T}_l^b$, and $\boldsymbol{T}_{l_a}^l \in \mathbb{SE}(3)$ are the predicted pose of $\mathscr{F}_b$, the offline calibrated extrinsics of the primary LiDAR to the IMU, and the extrinsics of the auxiliary LiDAR to the primary LiDAR, respectively.

3) Visual Feature Extraction and Tracking: For each input image, in order to make the framework robust to illumination changes, we first homogenize its pixels with CLAHE histogram equalization [47]. Since an image contains millions of pixels, it's impractical to incorporate all the dense visual data in estimation. To lower the computation load, we extract its FAST [48] corners and track these feature points from frame to frame via Lucas-Kanade optical flow [49]. In this way, the key points detected in a certain time window can be efficiently and stably associated to avoid complicated feature matching.

E. Front-end: Continuous-time Laser-Visual-Inertial Odometry

In the front-end, we aim to fuse all the sensor data into a unified framework and perform a tightly-coupled estimation of the carrier trajectory, map points as well as sensor states.

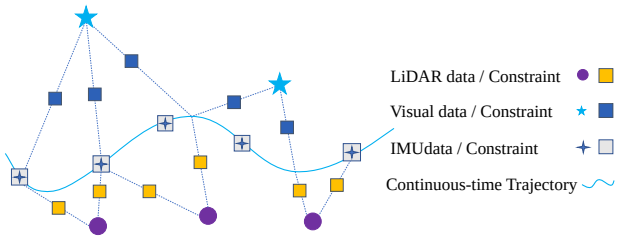

Fig. 2. Continuous-time laser-visual-inertial odometry.

1) Formulation: Denote the LiDAR data, the image features, and the IMU measurements by $\mathcal{L}$, $\mathcal{V}$, and $\mathcal{I}$, respectively. The variables to be jointly estimated in Ct-LVIO are the $\mathbb{R}(3)$ and $\mathbb{SO}(3)$ B-splines ($\mathcal{C}$), the visual map points ($\mathcal{M}$), and the extrinsics along with the IMU biases ($\mathcal{T}$). The objective of Ct-LVIO is to maximize the posterior probability, i.e.,

$$\{\mathcal{C},\mathcal{M},\mathcal{T}\}^* = \arg\max_{\mathcal{C},\mathcal{M},\mathcal{T}} p(\mathcal{C},\mathcal{M},\mathcal{T} \mid \mathcal{L},\mathcal{V},\mathcal{I}). \tag{7}$$

According to Bayes law, $p(\cdot)$ can be reformulated as,

$$p(\mathcal{C},\mathcal{M},\mathcal{T} \mid \mathcal{L},\mathcal{V},\mathcal{I}) = \frac{p(\mathcal{C},\mathcal{M},\mathcal{T}) p(\mathcal{L},\mathcal{V},\mathcal{I} \mid \mathcal{C},\mathcal{M},\mathcal{T})}{p(\mathcal{L},\mathcal{V},\mathcal{I})}. \tag{8}$$

LiDAR data are associated with extrinsics and trajectories, while IMU observations are only associated with the trajectories. Further, via preprocessing and offline calibration, we already have reasonable estimates for the trajectories, visual points, and extrinsics. Hence, they can be considered as independent prior terms in Eq. 8, leading to,

$$\begin{aligned} p(\mathcal{C},\mathcal{M},\mathcal{T} \mid \mathcal{L},\mathcal{V},\mathcal{I}) \propto\ & p(\mathcal{C}) p(\mathcal{M}) p(\mathcal{T}) \\ & p(\mathcal{L} \mid \mathcal{C},\mathcal{T}) p(\mathcal{V} \mid \mathcal{C},\mathcal{M},\mathcal{T}) p(\mathcal{I} \mid \mathcal{C}), \end{aligned} \tag{9}$$

where the first three terms are the prior terms, and the last three are the posteriors of the LiDAR, camera and IMU, respectively. In Maximum A Posterior estimation, these posterior terms are modeled as the corresponding high-dimensional Gaussian distributions characterized by their means and covariances. Thus, the optimization objective of our front-end can be equated with minimizing the sum of the quadratic error terms, i.e.,

$$\begin{aligned} \{\mathcal{C},\mathcal{M},\mathcal{T}\}^* = \arg\min_{\mathcal{C},\mathcal{M},\mathcal{T}} (& {}^L\boldsymbol{e}^T \boldsymbol{Q}_L {}^L\boldsymbol{e} \\ & + {}^V\boldsymbol{e}^T \boldsymbol{Q}_V {}^V\boldsymbol{e} + {}^I\boldsymbol{e}^T \boldsymbol{Q}_I {}^I\boldsymbol{e}), \end{aligned} \tag{10}$$

where ${}^L\boldsymbol{e}$, ${}^V\boldsymbol{e}$, and ${}^I\boldsymbol{e}$ are the laser, visual, and inertial error terms, and $\boldsymbol{Q}_L$, $\boldsymbol{Q}_V$, and $\boldsymbol{Q}_I$ are their corresponding covariance matrices. To perform the joint optimization, we need to construct the concrete forms of ${}^L\boldsymbol{e}$, ${}^V\boldsymbol{e}$, and ${}^I\boldsymbol{e}$ first.

2) Laser Error Term: Different types of LiDAR often have different scanning patterns. Up to now, almost all of the existing laser-visual-inertial SLAM approaches are based on feature points, lines, or planes of a scan for a specific type of LiDAR. This kind of methods has the advantage that the geometric constraints can be conveniently established, bringing fairish results. However, such feature-based methods usually suffer from poor scalability and are sensitive to noise. Therefore, we seek a more general and robust way to perform

point cloud registration. Inspired by the probabilistic submap representation [30], [50], we register a scan to a submap by finding the highest cumulative hit probability of the raw point cloud in the probability map. Accordingly, the optimization objective is,

$$\{\boldsymbol{q}_l^s, \boldsymbol{p}_l^s\}^* = \arg\max_{\boldsymbol{q}_l^s, \boldsymbol{p}_l^s} \sum_i p(R(\boldsymbol{q}_l^s)\boldsymbol{p}_i^l + \boldsymbol{p}_l^s), \tag{11}$$

where $\boldsymbol{p}_i^l$ is the i-th point of the point cloud in LiDAR frame, $\boldsymbol{q}_l^s$ and $\boldsymbol{p}_l^s$ are the orientation and position of the LiDAR in the submap, $R(\cdot)$ converts a quaternion to the corresponding rotation matrix, and $p(\cdot)$ returns the submap probability at the associated voxel. Such a way avoids feature extraction and matching, and the registration efficiency is very high if a reasonable initial value is available. Hence, on the premise of predicting the approximate pose from IMU readings, we can register the synchronized point cloud to the submap to obtain the registered pose. Afterwards, to fuse it with IMU and camera data, we uniformly sample pose anchors between the starting and ending timestamps of the scan to construct laser error terms.

Assume that the poses of the k-th frame and the $k+1$-th frame obtained by the aforementioned scan registration are $(\boldsymbol{q}_k^s,\ \boldsymbol{p}_k^s)$ and $(\boldsymbol{q}_{k+1}^s,\ \boldsymbol{p}_{k+1}^s)$. The anchor pose at $t \in (t_k, t_{k+1})$ can be obtained by position and quaternion interpolation,

$$\boldsymbol{p}_t = \boldsymbol{p}_k + \frac{t - t_k}{t_{k+1} - t_k}(\boldsymbol{p}_{k+1} - \boldsymbol{p}_k), \tag{12}$$

$$\boldsymbol{q}_t = \boldsymbol{q}_k \otimes (\boldsymbol{q}_k^* \otimes \boldsymbol{q}_{k+1})^{\frac{t-t_k}{t_{k+1}-t_k}}. \tag{13}$$

Thus, the error terms between the anchor pose and the trajectory can be defined as,

$${}^L\boldsymbol{e}_{p,t} = \boldsymbol{p}_t - \boldsymbol{p}(t), \tag{14}$$

$${}^L\boldsymbol{e}_{q,t} = \log(\boldsymbol{q}_t{}^* \otimes \boldsymbol{q}(t)). \tag{15}$$

3) Visual Error Term: To keep the computation bounded, we only reserve the visual feature points in a local time window to construct visual error terms. We initialize the depth of a visual feature point in two ways. One is to associate the depth from the point cloud map that has been registered nearby. The other is to estimate the initial depth via triangulation based on multiple observation frames of the point in the past time window. Note that a point is only triangulated when its parallax in keyframes reaches a certain threshold. Assume the k-th feature point $\breve{\boldsymbol{p}}_{i,k}^i$ of the i-th frame is subsequently observed by the j-th frame and its estimated depth is $d_{i,k}$. According to the epipolar geometry, the reprojection error ${}^V\boldsymbol{e}_{i,j,k}$ between the i-th and the j-th frame regarding $(\breve{\boldsymbol{p}}_{i,k}^i, d_{i,k})$ is defined as,

$${}^V\boldsymbol{e}_{i,j,k} = \breve{\boldsymbol{p}}_{i,k}^j - \pi(\boldsymbol{T}_b^c \boldsymbol{T}_b^{w^T}(t_j) \boldsymbol{T}_b^w(t_i)(\boldsymbol{R}_b^{c^T} \mathring{\boldsymbol{p}}_{i,k}^{c_i}), \boldsymbol{K}), \tag{16}$$

where $\boldsymbol{K}$ is the camera intrinsic matrix, $\boldsymbol{T}_b^c(\boldsymbol{R}_b^c)$ is the IMU-to-camera extrinsics which are obtained beforehand by calibration, and π represents the projection of the associated 3D spatial visual point to the 2D image plane.

4) Inertial Error Term: Each IMU reading contains 3-axis acceleration and 3-axis angular velocity. Our goal is to adjust the $\mathbb{R}^3$ and $\mathbb{SO}(3)$ trajectories so that the acceleration and angular velocity calculated from them are close to those measured from IMU. At time t, the carrier pose $(\boldsymbol{p}_b^w(t), \boldsymbol{q}_b^w(t))$ can be directly queried from the trajectories. The acceleration and angular velocity of the carrier at time t can be deduced from the second-order derivative of the $\mathbb{R}^3$ trajectory $(\ddot{\boldsymbol{p}}_b^w(t))$ and the first-order derivative of the $\mathbb{SO}(3)$ one $(\dot{\boldsymbol{q}}_b^w(t))$, respectively. Thus, the loss term between the IMU reading $\boldsymbol{a}_t$ $(\boldsymbol{\omega}_t)$ at time t and the $\mathbb{R}^3$ $(\mathbb{SO}(3))$ trajectory can be defined as,

$${}^I\boldsymbol{e}_{a,t} = \boldsymbol{a}_t - R(\boldsymbol{q}_b^w(t))^T(\ddot{\boldsymbol{p}}_b^w(t) - \boldsymbol{g}^w) - {}^a\boldsymbol{b} \tag{17}$$

$${}^I\boldsymbol{e}_{\omega,t} = \boldsymbol{\omega}_t - R(\boldsymbol{q}_b^w(t))^T(\dot{\boldsymbol{q}}_b^w(t)) - {}^\omega\boldsymbol{b}, \tag{18}$$

where $\boldsymbol{g}^w$ is the gravity vector; ${}^a\boldsymbol{b}$ and ${}^\omega\boldsymbol{b}$ are the biases of the accelerator and gyroscope, respectively.

5) Laser-Visual-Inertial Joint Optimization: In a local time window of the front-end, after the constraints among different sensors are established, we can jointly optimize all the variables resorting to common mathematical tools. Those variables include the extrinsics ($\boldsymbol{R}_c^b$, $\boldsymbol{p}_c^b$, $\boldsymbol{R}_l^b$ and $\boldsymbol{p}_l^b$), the IMU biases (${}^a\boldsymbol{b}$ and ${}^\omega\boldsymbol{b}$), the inverse depths of all the 3D visual points ($\boldsymbol{d}$), and the control points of the trajectories (${}^R\boldsymbol{c}$ and ${}^p\boldsymbol{c}$), resulting in a compact vector,

$$\boldsymbol{x} = [\boldsymbol{R}_c^{b^T}, \boldsymbol{p}_c^{b^T}, \boldsymbol{R}_l^{b^T}, \boldsymbol{p}_l^{b^T}, {}^a\boldsymbol{b}^T, {}^\omega\boldsymbol{b}^T, \boldsymbol{d}^T, {}^R\boldsymbol{c}^T, {}^p\boldsymbol{c}^T]^T.$$

With the error terms defined above, the concrete loss function can be formulated as,

$$F(\boldsymbol{x}) = \sum \|{}^L\boldsymbol{e}_{p,t}\|_{\boldsymbol{Q}_L} + \sum \|{}^L\boldsymbol{e}_{q,t}\|_{\boldsymbol{Q}_L} + \sum_{i,j,k} \|{}^V\boldsymbol{e}_{i,j,k}\|_{\boldsymbol{Q}_V} + \sum \|{}^I\boldsymbol{e}_{a,t}\|_{\boldsymbol{Q}_a} + \sum \|{}^I\boldsymbol{e}_{\omega,t}\|_{\boldsymbol{Q}_\omega}, \tag{19}$$

in which $\|\boldsymbol{e}\|_{\boldsymbol{Q}_\alpha} = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{Q}_\alpha^{-1}\boldsymbol{e}$, $\alpha \in \{L, V, a, \omega\}$, and $\boldsymbol{Q}_L$, $\boldsymbol{Q}_V$, $\boldsymbol{Q}_a$, and $\boldsymbol{Q}_\omega$ are the measuring covariances of the LiDAR, camera, accelerator, and gyroscope, respectively. Accordingly, the optimization objective is defined as,

$$\boldsymbol{x}^\star = \arg\min_{\boldsymbol{x}} F(\boldsymbol{x}). \tag{20}$$

To find the optimal solution, we start from the estimated initial values and resort to the Levenberg-Marquardt algorithm [51], [52] to solve the problem. Specifically, suppose that all the elements of ${}^L\boldsymbol{e}_{p,t}$, ${}^L\boldsymbol{e}_{q,t}$, ${}^V\boldsymbol{e}_{i,j,k}$, ${}^I\boldsymbol{e}_{a,t}$, and ${}^I\boldsymbol{e}_{\omega,t}$ are rearranged into a stacked function vector $\boldsymbol{f}(\boldsymbol{x})$. Denote $\boldsymbol{J}^T\boldsymbol{Q}^{-1}\boldsymbol{J}$ and $\boldsymbol{J}^T\boldsymbol{Q}^{-1}\boldsymbol{f}$ by $\boldsymbol{H}$ and $\boldsymbol{\delta}$ respectively, where $\boldsymbol{J} = \frac{d\boldsymbol{f}(\boldsymbol{x})}{d\boldsymbol{x}}$, $\boldsymbol{Q}$ is the concatenated covariance matrix, and $\boldsymbol{f}$ is the error of the current iteration. The compact variable $\boldsymbol{x}$ can be updated via,

$$\boldsymbol{x} \leftarrow \boldsymbol{x} \ominus (\boldsymbol{H} + \gamma\boldsymbol{I})^{-1}\boldsymbol{\delta}, \tag{21}$$

where γ is the damping coefficient of the current iteration, $\boldsymbol{I}$ is the identity matrix which has the same dimension as $\boldsymbol{H}$, and $\ominus$ means the minus operation on the manifold.

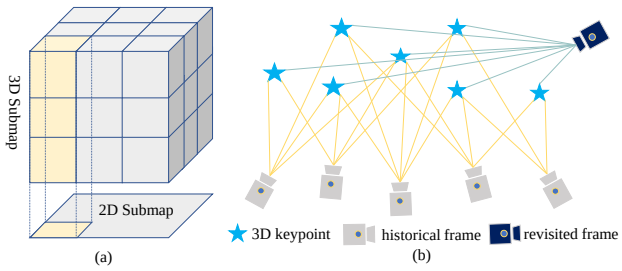

Fig. 3. (a) Submap projection. (b) Visual relocalization.

F. Backend: Pose Graph Adjustment via Laser-Visual Loop Constraints

1) Laser Loop Constraint: The sparseness of the point cloud brings challenges to the algorithm design of laser loop detection. In addition, the establishment of accurate loop constraints is equally important for the backend. Considering that the multi-frame point cloud with pose information can better reflect the location characteristics stably, we project the point cloud of a submap onto the horizontal plane to detect the loop from submap to submap. At the same time, the transformation relationship between submaps can be obtained by extracting and matching feature points from the projected submaps.

When detecting laser loops, whenever a new 3D submap construction is completed, the submap-to-submap loop detection will be launched. That newly finished submap will be matched with all historical submaps one by one to find the possible loop closure.

Specifically, assume that we are going to determine whether there is a loop closure between the 3D submap ${}^3\mathbb{S}_a$ and ${}^3\mathbb{S}_b$ or not. The 2D submaps, ${}^2\mathbb{S}_a$ and ${}^2\mathbb{S}_b$, will be obtained by projecting ${}^3\mathbb{S}_a$ and ${}^3\mathbb{S}_b$ along the gravity, respectively. During projection, the probability of each grid is the sum of the probabilities of all voxels corresponding to it in the direction of gravity (as Fig. 3 (a) illustrates). In this way, each grid reflects the richness of ground objects in the vertical direction at that position.

After that, we extract SURF corners and calculate the corresponding feature descriptors [53] of ${}^2\mathbb{S}_a$ and ${}^2\mathbb{S}_b$. When performing the loop detection, we match the SURF corners of ${}^2\mathbb{S}_a$ with those of ${}^2\mathbb{S}_b$ using their descriptors by FLANN search [54] under Lowe's strategy [55]. If the number of successfully matched points is larger than an acceptable threshold (5 in our setting), a similarity transformation will be further estimated between these matched points. When the scale value s of the estimated transformation is close to 1 ($s \in [0.9, 1.1]$), it is regarded that there is a loop between ${}^3\mathbb{S}_a$ and ${}^3\mathbb{S}_b$.

To further establish the scan-to-submap loop constraint, we first calculate the initial value of the point cloud under the target submap where the loop occurs. Assuming that the poses of ${}^3\mathbb{S}_a$ and ${}^3\mathbb{S}_b$ are $\boldsymbol{T}_{s_a}$ and $\boldsymbol{T}_{s_b}$, respectively, and that the pose of a point cloud in ${}^3\mathbb{S}_b$ is $\boldsymbol{T}_l^{s_b}$, the pose of the point cloud in ${}^3\mathbb{S}_a$ can be obtained as $\boldsymbol{T}_l^{s_a}$ according to the 3D coordinate transformation rule. Moreover, since the scale-determined similarity transformation between ${}^2\mathbb{S}_a$ and ${}^2\mathbb{S}_b$ is only 3-DoF, the translation value z in $\boldsymbol{T}_l^{s_a}$ is left undetermined. To obtain a proper estimate of z, we resort to branch-and-bound searching as the practice of [50].

2) Visual Loop Constraint: Compared with laser loop detection, the visual one is relatively mature. We resort to DBoW2 [18] to detect visual loops. Since the error and noise of visual positioning are larger than those of laser, we first roughly locate the loop pose from the visual features and subsequently refine it by matching the point cloud to the submap.

Specifically, we set two relative position (rotation) thresholds, coarse and fine, for keyframe selection. The coarse threshold is used for loop detection, and the fine one is set for caching keyframes for relocalization. When a loop is detected, we take out several keyframes near the loop frame from the cached keyframes. Next, one-to-one feature point matching is carried out among these keyframes, and the common-view feature points are triangulated. Afterwards, Perspective-n-Point [56] relocalization can be performed by using the feature points observed in the loop frame. To further improve the repositioning accuracy, after the successful repositioning of Perspective-n-Point, all the observations in the local window will be further constructed as a bundle adjustment problem [57] (as Fig. 3(b) shows). If the solving of the bundle adjustment converges, it is considered that the visual loop is successfully found and a reasonable repositioning pose is obtained. At last, we convert the relocated pose into the scan-to-submap pose and employ the scan-to-submap registration to refine the positioning.

IV. Experiment

A. Datasets, Metrics, and Implementation

Datasets. *Public Dataset.* The public dataset VIRAL [58] was taken for experimental verification, whose acquisition platform was a DJI M600 UAV, which was equipped with two OS1-16-gen-1 LiDARs (the horizontally mounted one was utilized), two uEye-1221-LE cameras (the left one was used), and a VectorNav-VN100 IMU.

Self-collected Dataset. To gather multi-sensor data for experiments, we developed a handheld device as shown in Fig. 1(a), which includes a ROBOSENSE 16-beam LiDAR and a ZED binocular camera (its left eye was used), in which a consumer-grade IMU is embedded. To test the performance of our framework in various scenes, we collected rich data in structured scenes (around buildings) and unstructured scenes (around rivers and bushes), HD-1∼HD-4 as Table I shown. Besides, four sequences when the carrier was in fast motion were also gathered from two places (HD-5∼HD-8). During acquisition, these series were accompanied by continuous intense motion (their highest angular velocities were over $200^\circ/s$).

Metrics. *Absolute Positioning Error (APE).* VIRAL's absolute position of the carrier was provided by a Leica tracker. During evaluation, we first aligned the estimated trajectory with the ground truth by Umeyama algorithm [59], and then calculated the average position deviation between the two trajectories as APE.

Relative Revisiting Error (RRE). Due to the obstruction of tall buildings and trees in the gathering environment of

TABLE I
DETAILS OF HD DATASET. "TRAJ. LEN.", "LIN. VEL.", AND "ANG. VEL." ARE ABBREVIATIONS OF TRAJECTORY LENGTH, LINEAR VELOCITY, AND ANGULAR VELOCITY, RESPECTIVELY.

Name	Traj. Len. (m)	Lin. Vel. (m/s)	Ang. Vel. $(^\circ/s)$	Area $(10^4 m^2)$	Scans
HD-1	3365.25	4.91	139.23	12.62	6279
HD-2	4574.29	4.61	195.38	24.34	8516
HD-3	3285.27	5.13	101.98	17.02	6159
HD-4	1223.89	2.21	168.45	1.78	4582
HD-5	137.65	2.13	250.38	3.80×10^{-4}	713
HD-6	110.57	1.88	228.61	8.79×10^{-4}	512
HD-7	84.65	2.22	201.10	3.49×10^{-4}	402
HD-8	101.25	2.36	238.92	5.81×10^{-4}	468

the self-collected dataset, the GNSS signal was extremely unstable. Therefore, we regarded the relative pose as the ground truth to evaluate the accuracy of the algorithm on our HD dataset. Specifically, to automatically generate the ground truth, we first selected out the point clouds around a revisited location, resulting in historical point clouds $\mathbb{P}_h$ and revisiting ones $\mathbb{P}_r$. After that, we conducted Normal Distribution Transform (NDT) [45] registration between $\mathbb{P}_h$ and $\mathbb{P}_r$ and manually checked whether the registration succeeded or not. If a pair of point clouds was successfully registered, the 6-DoF relative pose of the revisiting point cloud in the historical one would be regarded as the ground truth. By repeating the above procedures over all the revisited locations and all the data sequences, we established our HD dataset with 6-DoF ground truth.

Implementation. All modules in Ct-LVI were implemented in C++. With the help of Ceres-Solver[2], we constructed and solved the front-end laser-visual-inertial joint optimization problem and the backend pose graph optimization problem. Message communication among processes was fulfilled by the popular Robot Operating System (ROS)[3].

The noise statistics of IMU were measured by IMU-utils[4]. The extrinsics among the LiDAR, camera, and IMU were calibrated by LVI-ExC [60]. The number of scans contained in a submap was empirically set to 100. The threshold of well-matched pairs of feature points for judging that there was a loop closure between two submaps was assigned as 5. A submap's resolution (voxel size) was set to $0.2m$. In comparative experiments, for all the competing rivals (R^{2}LIVE [9], R^{3}LIVE [10], LVI-SAM [11], D-LIOM [30], Fast-LIO2 [26], and LIO-Mapping (LIOM for short) [28]), involved in the comparisons, we adopted their corresponding open-source implementations, and the parameters of these competitors were set in accordance with their original papers except for the adaption of the sensor related configurations.

All experiments were carried out on a notebook computer with the configuration of "Intel(R) Core(TM) i7-8750H CPU @ 2.20 GHz $\times$ 2" and 16GB RAM.

[2]http://ceres-solver.org/
[3]http://wiki.ros.org/
[4]https://github.com/gaowenliang/imu_utils

TABLE II
APES (m) OF LASER-VISUAL-INERTIAL ODOMETRY FRAMEWORKS ON VIRAL. "W-AVG" IS THE WEIGHTED AVERAGE ERROR OF THE RELEVANT SEQUENCES.

	eee1	eee2	eee3	nya1	nya2	nya3	sbs1	sbs2	sbs3	w-avg
R^{2}LIVE	1.23	**0.13**	0.25	0.23	0.61	0.15	2.81	2.88	0.33	0.94
R^{3}LIVE	**0.11**	0.32	0.32	**0.14**	**0.13**	**0.11**	0.26	0.77	**0.12**	0.24
Ct-LVIO	0.16	0.18	**0.14**	**0.14**	**0.13**	0.15	**0.15**	**0.16**	0.19	**0.15**

B. Results of Laser-Visual-Inertial Odometry

We first investigate the performance of the proposed LiDAR-Visual-Inertial Odometry (Ct-LVIO) in the indoor structured environment on VIRAL. On nine data series covering three scenes, the APEs of Ct-LVIO and its counterparts were evaluated and recorded in Table II. From Table II, in terms of APE, compared with the existing state-of-the-art laser-visual-inertial odometry approaches, R^{2}LIVE [9] and R^{3}LIVE [10], our Ct-LVIO has achieved the best results in most sequences. Notably, its "w-avg" error on VIRAL measured by APE is $9cm$ lower than R^{3}LIVE. From the perspective of stability, R^{2}LIVE achieves an accuracy of $0.13m$ in "eee2", but it also produces a huge error of $1.23m$ in the same scenario. Likewise, R^{3}LIVE [10] fluctuates from $0.12m$ to $0.77m$ in "sbs". By contrast, our Ct-LVIO is much more robust, producing more consistent results in different scenes with various moving trajectories. Besides, one phenomenon that Ct-LVIO performs less effectively on a few sequences needs to be further discussed. The underlying reason may lie in the manner of scan-to-scan association. The VIRAL dataset is collected from low-speed moving drones and its collection scenarios are small structured scenes such as indoor halls or enclosed courtyards. Under such conditions, R^{2}LIVE and R^{3}LIVE associate point clouds using laser geometric features, making it easier to establish precise geometric constraints. Relatively speaking, our Ct-LVIO associates point clouds using probability submaps, which may drop some geometric detail due to the voxelized submap representation.

To compare the performance of Ct-LVIO with state-of-the-art approaches more intuitively, we also provide detailed qualitative experimental results. First, we align their positioning trajectories on "eee1", "nya2", and "sbs3" with the ground-truth ones and draw them in Fig. 4. It can be seen that R^{2}LIVE produces large deviations on all the three sequences. Likewise, R^{3}LIVE only performs well on "eee1", but deviates largely from the ground-truth on "nya2" and "sbs2". By contrast, our Ct-LVIO obtains the best estimation results, which are in good agreement with the actual motion trajectories on all the sequences. In addition, we also generated the corresponding point cloud maps of the three sequences under the trajectories estimated by Ct-LVIO. As shown in Fig. 5, the maps constructed by Ct-LVIO can accurately reconstruct the overall structure as well as local details of these scenes, which verifies the high accuracy of Ct-LVIO's trajectory estimation.

C. Mapping Improvement with Laser-visual Loop Constraints

To corroborate the effectiveness of the laser-visual fused loop detection strategy, the global mapping results are eval-

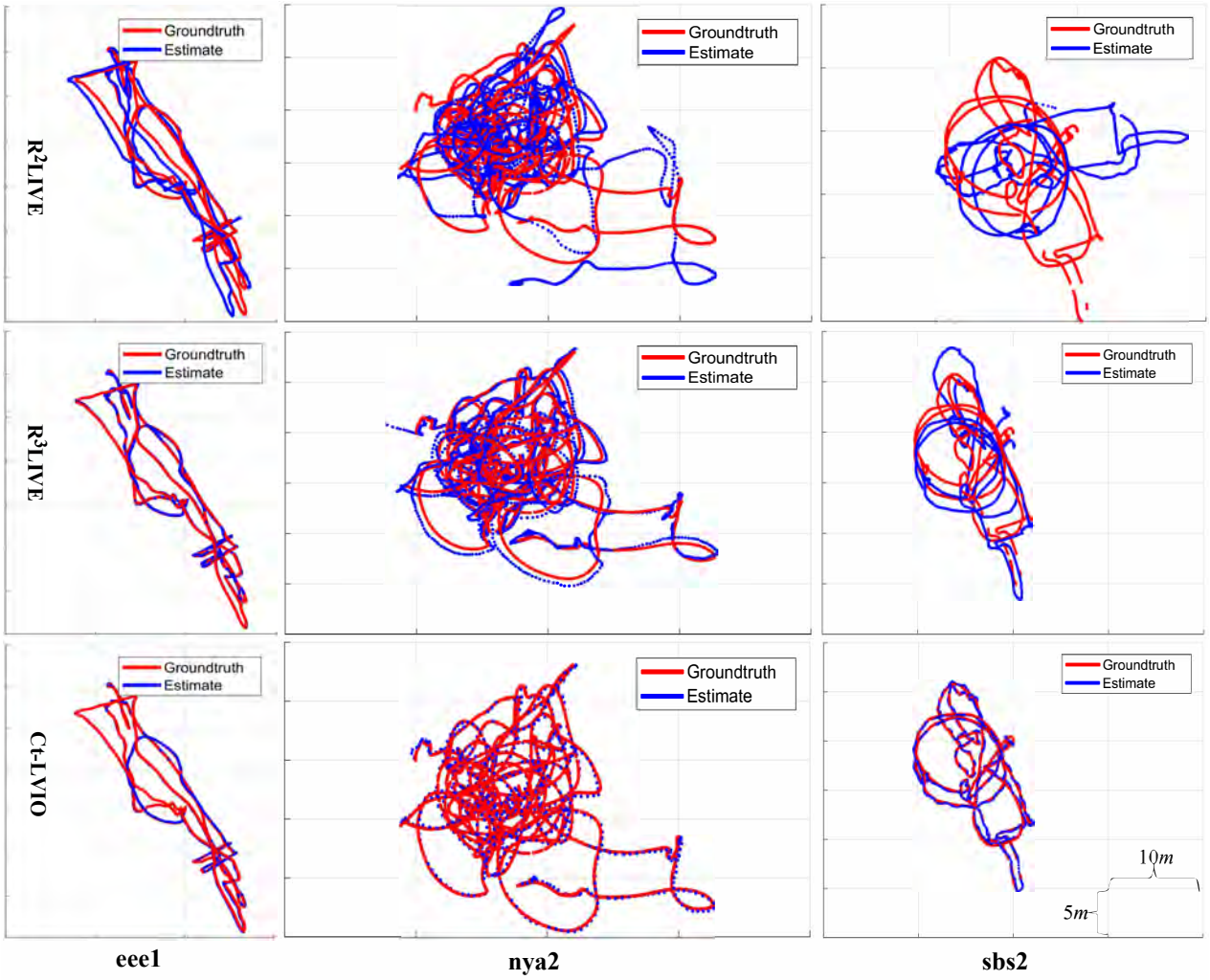

Fig. 4. Estimated trajectories. Subfigures from top to bottom plot the trajectories estimated by R^{2}LIVE [9], R^{3}LIVE [10], and our Ct-LVIO respectively.

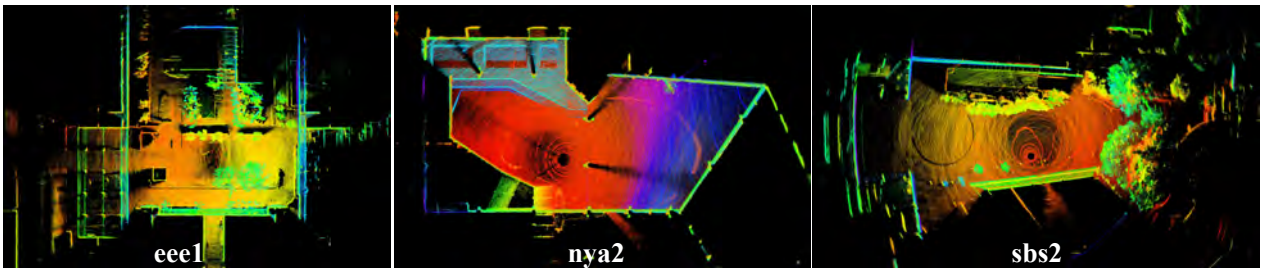

Fig. 5. Point cloud maps built by Ct-LVIO in which the points are colored by their z values gradually from red to blue.

uated both quantitatively and qualitatively. We collected two types of outdoor data. One is the mixed data of structured and unstructured scenes (HD-1, HD-2, and HD-3) gathered along campus roads, and the other is the data collected in an unstructured jungle (HD-4). The former can test the performance of Ct-LVI in large-scale scenes, and the latter can assess Ct-LVI in unstructured scenes.

The top row of Fig. 6 shows the overall mapping results of the full framework Ct-LVI (with Ct-LVIO in the front-end and laser-visual loop constraint construction in the backend) when the carrier works outdoors. It can be seen that Ct-LVI can build high-precision maps with global consistency, whether in hybrid scenes or completely unstructured scenes, which shows its adaptability to complex environments.

Under the same outdoor sequences, we also evaluated the revisiting errors of competing odometry approaches (R^{2}LIVE [9] and R^{3}LIVE [10]), and LVI-SAM [11] which is the only existing LVI-SLAM framework with a loop closure detection at its backend. The obtained results are listed in Table III and the positioning trajectories of the compared methods are also drawn in the bottom row of Fig. 6. It can be seen that when only the front-end odometry is carried out,

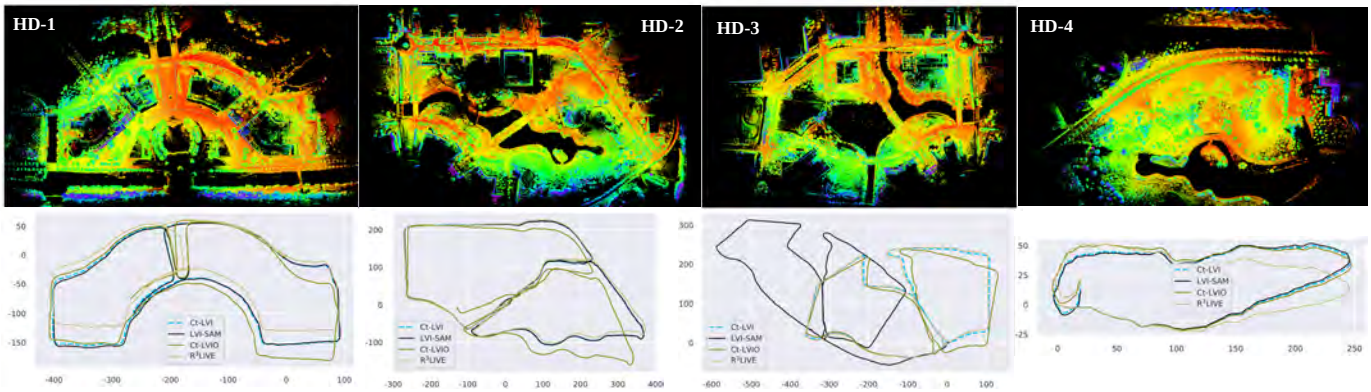

Fig. 6. Mapping results built by Ct-LVI (the top row) and the estimated trajectories of competing laer-visual-inertial odometry/SLAM frameworks (the bottom row) in large-scale structured scenes (HD-1, HD-2, and HD-3) and an unstructured area (HD-4).

TABLE III
RREs (m) OF COMPETING LASER-VISUAL-INERTIAL ODOMETRY/SLAM FRAMEWORKS ON THE OUTDOOR DATASET. ✗ MEANS "FAILED". CT-LVI$_{vl}$/CT-LVI$_{ll}$ DENOTES CT-LVI WITH VISUAL-ONLY/LIDAR-ONLY LOOP CONSTRAINTS AT THE BACKEND, RESPECTIVELY.

	HD-1	HD-2	HD-3	HD-4
R^{2}LIVE [9]	✗	✗	✗	✗
R^{3}LIVE [10]	11.82	70.71	89.98	12.14
Ct-LVIO	21.45	25.54	12.87	7.43
LVI-SAM [11]	0.46	0.35	✗	2.45
Ct-LVI$_{vl}$	0.42	0.36	12.44	7.32
Ct-LVI$_{ll}$	0.42	0.38	0.26	0.39
Ct-LVI	**0.24**	**0.18**	**0.21**	**0.35**

our Ct-LVIO and R^{3}LIVE have large cumulative drifts when long-term mapping in large-scale scenes, let alone the quick divergences of R^{2}LIVE on these challenging sequences. When the backend laser-visual loop detection is turned on, Ct-LVI successfully reduced the revisiting error to the decimeter level, ensuring the global consistency of mapping. As for LVI-SAM, although its positioning results on HD-1 and HD-2 are close to Ct-LVI, its result on HD-4 is one order lower than ours and it encounters a divergence on HD-3, showing lower stability.

Additionally, we also perform ablation studies on the visual and laser loop constraints at the backend. The results for Ct-LVI$_{vl}$ (Ct-LVIO + visual loop constraints) as well as Ct-LVI$_{ll}$ (Ct-LVIO + laser loop constraints) are provided in Table III. It can be seen that when only visual loop constraints are constructed, Ct-LVI$_{vl}$ successfully eliminates the cumulative error on HD-1 and HD-2, but exhibits significant drifts on HD-3 and HD-4. Comparatively, the laser loop constraints are successfully established by Ct-LVI$_{ll}$ on all the sequences, showing greater stability. From the results of Ct-LVI, it can be observed that when both the visual and laser loop constraints take effect, the system can achieve the best performance to construct global consistency maps with high precision.

D. Performance under Fast Motion

In order to evaluate the effectiveness of IMU fusion, we evaluated Ct-LVIO and its competing methods on the collected fast-moving dataset. The relative revisiting errors of

TABLE IV
RREs (m) UNDER FAST CARRIER MOTION. ✗ MEANS "FAILED".

	HD-5	HD-6	HD-7	HD-8
R^{2}LIVE [9]	✗	✗	✗	✗
R^{3}LIVE [10]	✗	✗	✗	✗
LVI-SAM [11]	✗	2.18	✗	0.86
Ct-LVIO	**0.29**	**0.30**	**0.20**	**0.31**

the compared methods on the four fast-moving sequences are presented in Table IV. The results clearly indicate that Ct-LVIO maintains a high level of positioning accuracy even in scenarios involving fast carrier motion, thereby demonstrating its successful fusion of IMU data. Furthermore, it is noteworthy that the laser feature-based approaches (R^{2}LIVE, R^{3}LIVE, and LVI-SAM) are highly susceptible to failure when the carrier undergoes rapid rotation. Conversely, the proposed Ct-LVIO consistently achieves successful localization. This can be attributed to the difficulty in extracting stable features during rapid carrier rotation, which subsequently impedes the establishment of accurate scan-to-scan correspondences. In contrast, the probabilistic map-based point cloud alignment employed in Ct-LVIO exhibits greater resilience to outliers in fast-moving scenarios, thus ensuring robust state estimation and precise reconstruction.

To intuitively showcase the positioning and mapping effects of Ct-LVIO when the carrier is moving rapidly, we plot the estimated trajectories of the carrier, the 6-DoF errors of the estimated poses against the ground truth, the built maps by Ct-LVIO on HD-5∼HD-8, and the corresponding gyroscope reading profiles during the data acquisitions in Fig. 7. From the profiles of the gyroscope readings and the moving trajectories of the carrier, the complexity of the motion can be readily observed. Nevertheless, the relatively low magnitudes of the 6-DoF errors in the estimated poses provide evidence of Ct-LVIO's ability to achieve highly accurate localization even in such challenging cases. Furthermore, the point cloud maps constructed reveal the capacity of Ct-LVIO to accurately capture the overall structures of buildings as well as the finer details such as windows, columns, and trunks. This observation serves as confirmation that Ct-LVIO is capable

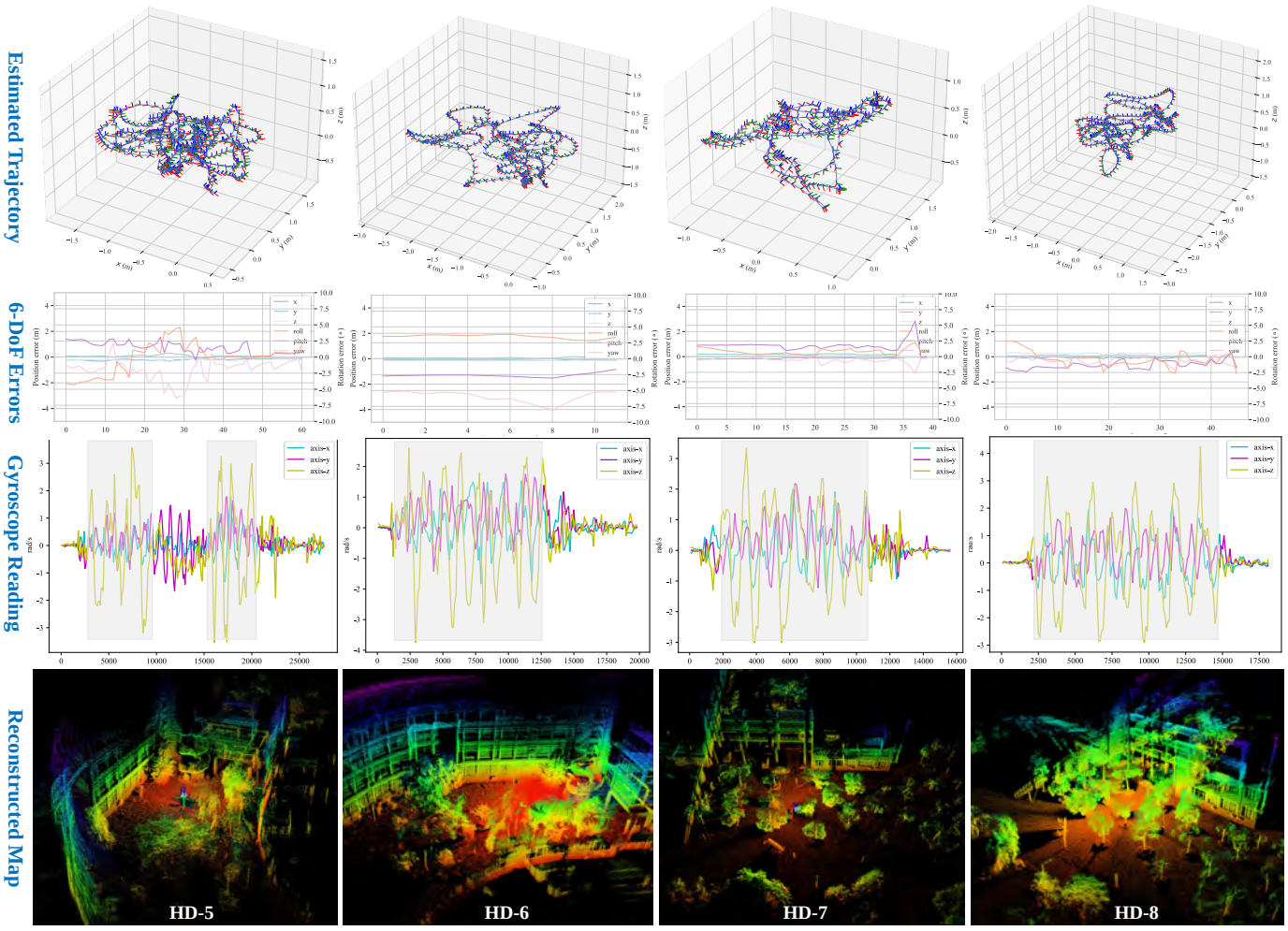

Fig. 7. Results under fast carrier motion. The first row: the estimated trajectories by Ct-LVIO when the handheld device moves intensely. The second row: the 6-DoF errors of the estimated poses against the ground truth. The third row: the corresponding profiles of the raw gyroscope readings, where the segments with large angular velocities are highlighted. The last row: the point cloud maps built by Ct-LVIO in which the points are colored by their z values gradually from red to blue.

of producing high-quality scene reconstructions even in the presence of fast carrier motion.

E. Sensor Degeneration Study

TABLE V
APEs (m) WHEN THE CAMERA/LIDAR DEGENERATES. CT-LIO IS THE ABBREVIATION OF CONTINUOUS-TIME LIDAR-INERTIAL ODOMETRY. CT-LVIO-BL DENOTES RUNNING CT-LVIO WHILE THE LIDAR IS OCCASIONALLY BLOCKED.

	eee1	eee2	eee3	nya1	nya2	nya3	sbs1	sbs2	sbs3
LIOM [28]	1.06	0.72	1.03	2.24	1.97	3.00	1.67	1.81	2.00
D-LIOM [30]	0.23	0.24	0.11	0.14	0.14	0.15	0.16	0.15	0.19
Fast-LIO2 [26]	0.13	0.12	0.16	0.12	0.14	0.14	0.14	0.14	0.13
Ct-LIO	0.16	0.11	0.22	0.19	0.17	0.14	0.47	0.15	0.21
Ct-LVIO-BL	0.15	0.13	0.26	0.17	0.16	0.14	0.15	0.17	0.21

One of the advantages of multi-sensor fusion is that the system can still work when one sensor fails occasionally. In this subsection, we evaluate the performance of Ct-LVIO when the LiDAR's/camera's data is lost.

Camera Degeneration: Regarding the scenario of camera degradation, we evaluate the ability of our system under such a case by conducting experiments under extreme conditions where the camera is completely blocked throughout the SLAM cycle, rendering image data entirely unavailable. When the functionality of the camera deteriorates, the front-end of the system will automatically transition to a laser-inertial odometry approach (termed as Ct-LIO). In the real case, camera degradation can be judged by the tracking results of the image. Specifically, we consider the camera to be degraded when there are insufficient points successfully tracked or insufficient points successfully triangulated. Under such a situation, the visual constraints will no longer take effect.

To investigate the system performance when the camera deteriorates, we evaluate Ct-LIO and three other laser-inertial odometry frameworks (LIOM [28], D-LIOM [30], and Fast-LIO2 [26]), using the VIRAL dataset [58] as a benchmark. The obtained results are presented in Table V. Although our system was not specifically designed for laser-inertial odometry, it can be seen from Table V that Ct-LIO can still run reliably and achieve high levels of positioning accuracy. Also, Ct-

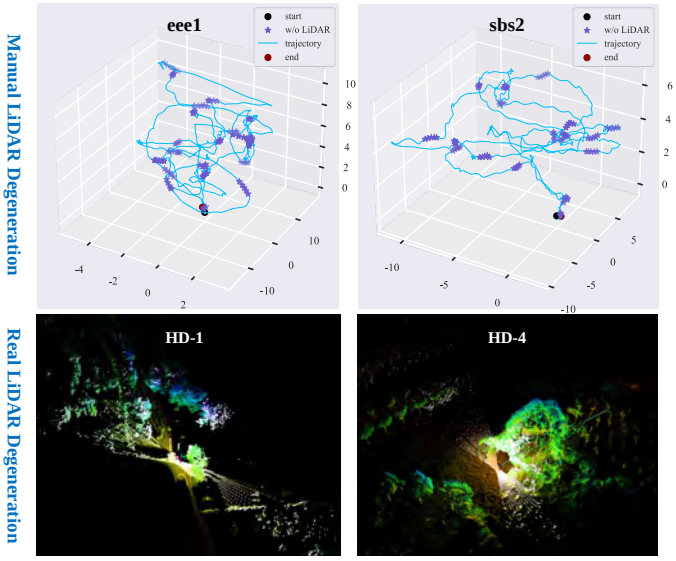

Fig. 8. LiDAR degeneration cases. Top: the carrier's moving trajectory (blue curves) and the positions where the LiDAR is manually blocked (purple pentagrams, "w/o" is the abbreviation of "without"). Bottom: two typical real-world degeneration cases encountered amid the collection of HD-1 and HD-4 where the white points stand for the incoming point cloud while the colored ones represent the previously built map.

LIO's performance is competing even compared with those pure laser-inertial odometry frameworks.

LiDAR Degeneration: As our system design prioritises the LiDAR as the principal sensor, we anticipate the visual-inertial odometry to assume responsibility when the LiDAR degrades momentarily. To substantiate this capability, we examined Ct-LVIO on VIRAL's sequences and manually blocked the LiDAR to simulate scenarios of LiDAR degradation. As depicted in the top row of Fig. 8, we demonstrate the distributions of manually blocked LiDAR segments on two sequences, namely eee1 and sbs2. Specifically, the purple pentagrams denote the locations where the LiDAR point clouds were deliberately discarded. The system front-end which runs under occasionally blocked LiDAR is named Ct-LVIO-BL. The localization accuracies of Ct-LVIO-BL are enumerated in Table V. It can be discerned that Ct-LVIO-BL still achieves satisfactory localization accuracy despite the artificial LiDAR degradations, demonstrating robustness in such extreme cases.

It is worth underscoring that, in addition to the simulation test of manually blocking the LiDAR, we also have several real-world cases of LiDAR degradation in the dataset actually collected. As exhibited in the bottom row of Fig. 8, two degradation cases encountered in HD-1 and HD-4 acquisitions are illustrated wherein the LiDAR scanning direction is approximately perpendicular to the ground, causing most nearby points to lie in the same plane. Conversely, distant points are sparse and principally leafages, rendering it difficult to extract meaningful features from them and resulting in a lack of constraints on the carrier pose. Nevertheless, as analyzed in Sec. IV-C, the mapping results with strong global consistency on HD-1 and HD-4 shown in Fig. 6 and the corresponding low revisiting errors listed in Table III also corroborate that our Ct-LVI has the faculty to address real-world cases of degradation.

TABLE VI
TIME COSTS (MS).

Front-end		Backend	
Preprocess	Optimize	Build laser constraint	Build visual constraint
20	75	54	80

F. Time Cost

In this part, we investigate the time efficiency of our framework. As shown in Table VI, we respectively list the time consumption of each frame of data (point cloud or image) processed by each main module in the front-end and backend of the system.

It can be seen that the frame rate of the front-end is about 10 frames per second, which ensures the real time processing efficiency for most of the existing LiDARs (for example, the frame rates of Velodyne, Robosense, Ouster, and Livox are all 10 frames). This should be attributed to the fact that the registration strategy based on probability maps avoids time-consuming feature extraction and matching, and the direct pose constraint also saves the time cost of continuous-time trajectory optimization. For the image data, although most cameras can provide frame rates higher than 10, to ensure sufficient parallax, the front-end actually only requires a lower image frame rate. In our implementation, we track the image keypoints at the raw frequency but set the triangulation frequency for the feature points to one-third of the raw frequency to avoid unnecessary computation. In addition, since the joint optimization of the front-end is triggered by the incoming point cloud and the processing time of each image is mainly in the preprocessing, the front-end can also process image data with high efficiency.

At the backend, the framework takes about 54 milliseconds to establish a laser loop constraint, while it takes about 80 milliseconds to completely construct a visual loop. Thus, the processing efficiency of the backend of Ct-LVI is comparable to that of its front-end. Besides, most of the data sent to the backend will be quickly screened when there is no candidate loop closure. Therefore, the processing efficiency of the backend is actually much higher, which is of great significance for eliminating accumulated errors in time.

V. CONCLUSION

In this article, we propose a continuous-time laser-visual-inertial SLAM framework Ct-LVI. Its front-end Ct-LVIO performs tightly coupled state estimation by integrating the loss terms of LiDAR, camera, and IMU with the continuous-time trajectory representation, supporting multi-sensor inputs without time synchronization as well as multi-LiDAR inputs in arbitrary scanning patterns. Ct-LVI's backend conducts pose graph optimization to eliminate accumulated drifts via laser-visual fused loop detection and constraint construction, ensuring a long-term mapping consistency. The effectiveness of Ct-LVI is corroborated on both the public dataset and the self-collected challenging dataset. Compared with its state-of-the-art counterparts, Ct-LVI produces more consistent maps in large-scale outdoor scenes and performs much more robustly

when the carrier works with fast motion. In future work, we will devote our efforts to further improve the scalability of our framework, e.g., to make it support multi-camera inputs.

REFERENCES

[1] X. Shao, L. Zhang, T. Zhang, Y. Shen, and Y. Zhou, "Mofisslam: A multi-object semantic slam system with front-view, inertial, and surround-view sensors for indoor parking," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 32, no. 7, pp. 4788–4803, 2022.

[2] A. Zhu, Y. Xiao, C. Liu, and Z. Cao, "Robust LiDAR-camera alignment with modality adapted local-to-global representation," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 33, no. 1, pp. 59–73, 2023.

[3] Y. Wang, Y. Qiu, P. Cheng, and J. Zhang, "Hybrid CNN-Transformer features for visual place recognition," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 33, no. 3, pp. 1109–1122, 2023.

[4] Z. Zhang, J. Sun, Y. Dai, B. Fan, and M. He, "VRNet: Learning the rectified virtual corresponding points for 3D point cloud registration," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 32, no. 8, pp. 4997–5010, 2022.

[5] S. Ren, Y. Zeng, J. Hou, and X. Chen, "CorrI2P: Deep image-to-point cloud registration via dense correspondence," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 33, no. 3, pp. 1198–1208, 2023.

[6] J. Zhang and S.Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, pp. 1242–1264, 2018.

[7] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: LiDAR-Inertial-Camera odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Macau, China, Nov. 2019, pp. 5848–5854.

[8] 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," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, USA, Oct. 2020, pp. 5112–5119.

[9] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R^2LIVE: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robot. Autom. Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.

[10] J. Lin and F. Zhang, "R^3LIVE: A robust, real-time, RGB-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, USA, May 2022, pp. 10 672–10 678.

[11] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, Xi'an, China, May 2021, pp. 5692–5698.

[12] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, USA, Sept. 2014, pp. 4973–4980.

[13] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, USA, May 2015, pp. 2174–2181.

[14] F. Tschopp, M. Riner, M. Fehr, L. Bernreiter, F. Furrer, T. Novkovic, A. Pfrunder, C. Cadena, R. Siegwart, and J. Nieto, "VersaVIS-An open versatile multi-camera visual-inertial sensor suite," *Sensors*, vol. 20, no. 5, pp. 1439: 1–9, 2020.

[15] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, May 2014, pp. 431–437.

[16] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Proc. Robot. Sci. Syst. Conf.*, California, USA, Jul. 2014, pp. 1–9.

[17] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[18] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, 2012.

[19] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.

[20] Y. Wang, Z. Sun, C. Xu, S. E. Sarma, J. Yang, and H. Kong, "LiDAR Iris for loop-closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, USA, Oct. 2020, pp. 5769–5775.

[21] G. Kim and A. Kim, "Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Madrid, Spain, Oct. 2018, pp. 4802–4809.

[22] S. Chen, B. Zhou, C. Jiang, W. Xue, and Q. Li, "A LiDAR/visual SLAM backend with loop closure detection and graph optimization," *Remote Sens.*, vol. 13, no. 14, pp. 2720–2748, 2021.

[23] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 3565–3572.

[24] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A lidar-inertial state estimator for robust and efficient navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Paris, France, May 2020, pp. 8899–8906.

[25] W. Xu and F. Zhang, "FAST-LIO: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robot. Autom. Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[26] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[27] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-LIO: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intell. Systems*, vol. 5, no. 7, pp. 1–20, 2023.

[28] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D lidar inertial odometry and mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, Montreal, Canada, May 2019, pp. 3144–3150.

[29] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, USA, Oct. 2020, pp. 5135–5142.

[30] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, "D-LIOM: Tightly-coupled direct lidar-inertial odometry and mapping," *IEEE Trans. Multimedia*, vol. 25, pp. 3905–3920, 2023.

[31] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, 2008.

[32] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 3281–3288.

[33] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "FAST-LIVO: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Kyoto, Japan, Oct. 2022, pp. 4003–4009.

[34] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sept. 2011, pp. 3297–3304.

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

[36] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, Portland, ON, USA, Jun. 2013, pp. 1360–1367.

[37] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, 2018.

[38] J. Mo and J. Sattar, "Continuous-time spline visual-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, USA, May 2022, pp. 9492–9498.

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

[40] 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.*, Las Vegas, USA, Oct. 2020.

[41] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative B-Splines on Lie groups," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, Seattle, WA, USA, Jun. 2020, pp. 11 145–11 153.

[42] D. Hug and M. Chli, "HyperSLAM: A generic and modular approach to sensor fusion and simultaneous localization and mapping in continuous-time," in *Proc. IEEE Int. Conf. 3D Vision*, Fukuoka, Japan, Nov. 2020, pp. 978–986.

[43] C. de Boor, *A Practical Guide to Spline*. New York, NY, USA: Springer, 1978.

[44] M. J. Kim, M. S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *Proc. 22nd Annual Conf. Comput. Graph. Interact. Tech.*, New York, NY, USA, Aug. 1995, pp. 369–376.
[45] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, USA, Oct. 2003, pp. 2743–2748.
[46] 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, 2017.
[47] S. Pizer, R. Johnston, J. Ericksen, B. Yankaskas, and K. Muller, "Contrast-limited adaptive histogram equalization: Speed and effectiveness," in *Proc. Conf. Visual. Biom. Comput.*, Atlanta, GA, USA, May 1990, pp. 337–345.
[48] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Euro. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 430–443.
[49] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, BC, Canada, Aug. 1981, pp. 674–679.
[50] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LiDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Stockholm, Sweden, May 2016, pp. 1271–1278.
[51] K. Levenberg, "A method for the solution of certain problems in least square," *Quarterly Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
[52] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameter," *J. Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
[53] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Euro. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 404–417.
[54] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vision Theory Appl.*, Lisboa, Potugal, Feb. 2009, pp. 331–340.
[55] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Jan. 2004.
[56] L. Vincent, M.-N. Francesc, and F. Pasca, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 5, pp. 155–166, 2009.
[57] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, Corfu, Greece, Sept. 1999, pp. 298–372.
[58] T. M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "NTU VIRAL: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint," *Int. J. Robot. Research*, vol. 41, no. 3, pp. 270–280, 2021.
[59] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, 1991.
[60] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, "LVI-ExC: A target-free LiDAR-visual-inertial extrinsic calibration framework," in *Proc. ACM Int. Conf. Multimedia*, Lisboa, Portugal, Oct. 2022, pp. 3319–3327.

Zhong Wang received the B.S. and M.S. degrees from the School of Surveying and Geo-Informatics, Tongji University, Shanghai, China, in 2016 and 2019, respectively. Starting from 2020, he is pursuing his Ph.D. degree at the School of Software Engineering, Tongji University, Shanghai, China. His research interests include motion planning of mobile robot, SLAM and computer vision.

Lin Zhang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2006, respectively. He received the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2011. From March 2011 to August 2011, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University. In Aug. 2011, he joined the School of Software Engineering, Tongji University, Shanghai, China, where he is currently a Full Professor. His current research interests include environment perception of intelligent vehicle, pattern recognition, computer vision, and perceptual image/video quality assessment. He serves as an Associate Editor for IEEE Robotics and Automation Letters, and Journal of Visual Communication and Image Representation. He was awarded as a Young Scholar of Changjiang Scholars Program, Ministry of Education, China.

Shengjie Zhao (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1988, the M.S. degree in electrical and computer engineering from the China Aerospace Institute, Beijing, China, in 1991, and the Ph.D. degree in electrical and computer engineering from Texas AM University, College Station, TX, USA, in 2004. He is currently the Dean of the School of Software Engineering and a Professor with the School of Software Engineering and the School of Electronics and Information Engineering, Tongji University, Shanghai, China. In previous postings, he conducted research at Lucent Technologies, Whippany, NJ, USA, and the China Aerospace Science and Industry Corporation, Beijing. He is a fellow of the Thousand Talents Program of China and an Academician of the International Eurasian Academy of Sciences. His research interests include artificial intelligence, big data, wireless communications, image processing, and signal processing.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently a Full Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, computer vision, and machine learning. He serves as an Associate Editor for Neurocomputing, Journal of Visual Communication and Image Representation, and Signal Processing: Image Communication. He is a Co-Chair of the Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He is a Senior Member of the International Society for Optical Engineering (SPIE).