Received: 11 November 2022 | Revised: 19 December 2022 | Accepted: 29 December 2022

IET Radar, Sonar & Navigation

DOI: 10.1049/rsn2.12376

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

FT-LVIO: Fully Tightly coupled LiDAR-Visual-Inertial odometry

Zhuo Zhang | **Zheng Yao** | **Mingquan Lu**

Department of Electronic Engineering, Tsinghua University, Beijing, China

Correspondence

Zheng Yao, Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China.
Email: yaozheng@tsinghua.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Number: 42274018

Abstract

In this paper, we propose a fully tightly-coupled multi-sensor fusion framework termed FT-LVIO, that fuses measurements from a light detection and ranging (LiDAR), a monocular camera and an inertial measurement unit (IMU) simultaneously to achieve robust and accurate state estimation in real time. FT-LVIO is built atop the framework of an error-state-iterated Kalman filter. To take full advantage of the complimentary characteristics of individual sensors, LiDAR point clouds are undistorted by IMU prediction to the nearest camera exposure time and the filter is updated with measurements from all sensors. In addition, an efficient sampling method for the LiDAR point-to-plane measurements is proposed, which can help select the measurements providing sufficient constraints to the pose estimation and facilitate a low-drift odometry. Extensive experiments are performed on both the public NTU dataset and the private handheld dataset, and the results show that the proposed FT-LVIO outperforms the state-of-the-art LiDAR-inertial, visual-inertial and LiDAR-visual-inertial methods in both accuracy and robustness. Furthermore, FT-LVIO can survive in the challenging staircase environment.

KEYWORDS

Kalman filters, navigation, odometry, sensor fusion

1 | INTRODUCTION

Ego-motion estimation is essential for mobile robots and vehicles especially when they are performing tasks autonomously in unfamiliar environments. The past 2 decades have witnessed many excellent solutions based on a single sensor, such as a camera [1], a light detection and ranging (LiDAR) [2] and an inertial measurement unit (IMU) [3]. Cameras are relatively cheap and can provide dense color and texture information of the environments at a moderate rate, but they are sensitive to illumination levels; thus, vision-based methods tend to fail in texture-less, exceptionally bright or dark scenes. LiDAR sensors can provide accurate range measurements at a low rate regardless of light conditions and are preferred in most scenarios for their robustness and capacity to build a dense map. However, LiDAR can suffer from point cloud sparsity; thus, LiDAR-based methods may degrade in structure-less or open-field environments. In contrast with cameras and LiDAR sensors, which collect the environment information, IMUs measure the angular velocity and acceleration of ego motion at a high rate, so they are barely affected by environments and can capture aggressive motions. IMU-based methods usually rely on the integration of raw, noisy and biased measurements and can present considerable drifts after a short time. To overcome the weakness of a single sensor and improve system robustness, multi-sensor fusion-based methods recently have attracted much attention and predictably will arouse more attention in the future since sensor costs keep decreasing. Therefore, this paper focuses on methods fusing at least two types of sensors.

In general, multi-sensor fusion methods can be grouped into two categories: loosely coupled methods and tightly coupled methods. Loosely coupled methods fuse odometry results of all individual sensors or odometry results of some sensors and specific measurements of others. Tightly coupled methods, instead, directly fuse the measurements of all individual sensors. Based on this categorization, we will briefly review the representative work on visual-inertial odometry (VIO), LiDAR-inertial odometry (LIO) and LiDAR-visual-inertial odometry (LVIO).

MSF-EKF SLAM [4] is a loosely coupled VIO, which uses an extended Kalman filter (EKF) to fuse the IMU measurements with visual odometry results. Also based on EKF,

MSCKF [5] augments the filter state with several previous camera poses and updates with multi-view observations of the same visual features, forming a tightly coupled VIO. In addition to the filter-based methods, optimization is another way to tightly couple IMUs and cameras. It usually estimates the states of multiple frames by iteratively minimizing the total visual and IMU measurement errors, which can help reduce the linearized errors of the system compared with the filter-based methods. OKVIS [6] is a representative optimization-based VIO, which maintains a sliding window to limit computation. VINS-Mono [7] further extends OKVIS with robust initialization, relocalization and loop closure and becomes one of the most popular VIO methods.

LOAM [2] is a representative loosely coupled LIO. It can not only run as a LiDAR odometry (LO) but also accept IMU inputs, which help to undistort skewed point clouds and initialize the pose estimation in scan matching. LIO-SAM [8] further introduces a factor graph with IMU preintegration, LiDAR odometry and loop closure factors to correct the IMU bias and reduce the drifts. Many tightly coupled LIO systems have emerged recently for their higher robustness and accuracy. LIOM [9] adopts the optimization framework of ref. [7] as a processing front end and introduces the rotation constraint at the back end to reduce the height drifts. LINS [10] designs an egocentric error-state iterated Kalman filter (ESIKF) to reduce filter divergence in a long run. Also based on ESIKF, FAST-LIO [11] proposes a new formula of the Kalman gain to reduce computation complexity and achieves 50 Hz odometry. FAST-LIO2 [12] further accelerates FAST-LIO to 100 Hz by incrementally building the k-d tree of the map using their proposed ikd-Tree structure.

In [13], a loosely-coupled LVIO with a sequential multilayer processing pipeline is proposed. A high-frequency IMU first provides motion prediction to a loosely coupled VIO and then the VIO result is loosely coupled with LiDAR measurements, achieving a coarse-to-fine motion estimation. LVI-SAM [14] combines VIO in [7] and LIO in [8] to form a tightly coupled LVIO. VIO provides an initial guess for scan matching in LIO and LIO output helps VIO to initialize the subsystem and retrieve feature depth. These two subsystems can function independently when failure is detected in one of them and they both provide odometry factors to a factor graph for accuracy. R2LIVE [15] uses ESIKF to fuse IMU data, respectively, with camera and LiDAR data in a tightly coupled manner and a local factor graph to refine the keyframe poses and visual landmark positions. FAST-LIVO [16] shares a similar framework with R2LIVE, while its visual measurements use direct photometric errors instead of feature-based reprojection errors, which save the feature extraction time and are more robust in texture-less environments. Based on the MSCKF framework, LIC-Fusion [17] estimates not only keyframe poses, visual and LiDAR feature positions but also IMU delays and extrinsics with respect to the camera and LiDAR. Strictly speaking, the aforementioned systems are not fully tightly coupled since they all essentially consist of two sub-systems (VIO and LIO); thus, they only achieve the tight couple of two sensors at a time. This design scheme does not take full advantage of the complimentary characteristics of all sensors and is prone to fail if no special care is taken to identify the integrity of each sub-system.

In this paper, we present a fully tightly coupled LiDAR-visual-inertial odometry algorithm based on the ESIKF framework of FAST-LIO2 [12] to achieve accurate and robust ego-motion estimation. The main contributions of our work are:

- We present a fully tightly coupled LiDAR-visual-inertial odometry framework (termed FT-LVIO), which simultaneously fuses measurements of three complimentary sensors and can achieve accurate ego-motion estimation.
- We propose an efficient sampling method for the LiDAR point-to-plane measurements, which helps select the measurements providing sufficient constraints to the pose estimation and facilitate a low-drift odometry.
- We perform extensive experiments on both public and private datasets to validate the proposed FT-LVIO. The results show that FT-LVIO outperforms the state-of-the-art methods even in face of sensor degradation.
- We conduct an ablation study to show the effects of the proposed sampling method and the introduction of the vision.

The remaining content of the paper is organized as follows. In Section 2, we first give an overview of the system and define the notations used throughout the paper. Then, each system module is introduced in detail and the proposed sampling method is specifically discussed. Section 3 presents the experiment results on public NTU and private handheld dataset and also the results of our ablation study. Section 4 is the final conclusion.

2 | FULLY TIGHTLY COUPLED LiDAR-VISUAL-INERTIAL ODOMETRY

2.1 | System overview

The framework of FT-LVIO is shown in Figure 1. The feature handler first extracts surf points from LiDAR point clouds and tracks corner points in camera images, respectively, at their input frequency. To tightly fuse the data of three sensors simultaneously, the synchronization handler associates each LiDAR feature frame with the nearest image feature frame and transforms all of the LiDAR feature points to the image time along with IMU-state propagation. The measurement handler selects valid LiDAR point-to-plane and visual point-to-pixel measurements and updates the ESIKF alongside the IMU prior to produce an accurate state estimation. The map handler finally builds and modifies the LiDAR and visual maps, which serve as the landmarks of the next round of measurements to reduce drifts.

Now, we define notations used throughout the paper. We denote ${}^{I}(\cdot), {}^{C}(\cdot), {}^{L}(\cdot)$ respectively as the local frame of the IMU, camera and LiDAR. Global frame ${}^{G}(\cdot)$ is the first IMU local frame. The total state vector of the system is:

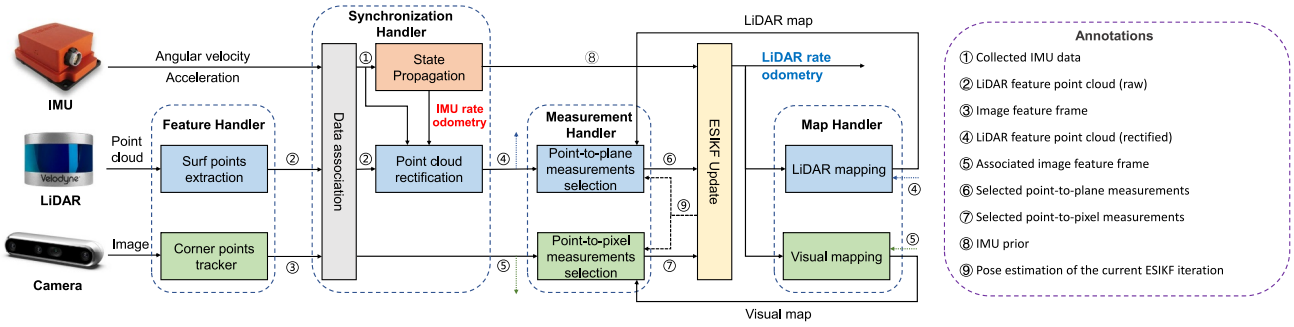


FIGURE 1 The overview of the FT-LVIO system. Each circled number near the arrow represents a specific data flow, which is annotated in detail at the right side of the figure.

$$\mathbf{x} = \left[{}^{G}_{I}\mathbf{R}^{T}, {}^{G}_{I}\mathbf{t}^{T}, {}^{G}\mathbf{v}^{T}, \mathbf{b}_a^{T}, \mathbf{b}_g^{T}, {}^{G}\mathbf{g}^{T} \right]^{T} \tag{1}$$

where ${}^{G}_{I}\mathbf{R}$ and ${}^{G}_{I}\mathbf{t}$ are the rotation matrix and translation vector from IMU frame to global frame, ${}^{G}\mathbf{v}$ is the IMU velocity in the global frame, $\mathbf{b}_a$ and $\mathbf{b}_g$ are the biases of IMU accelerometer and gyroscope, and ${}^{G}\mathbf{g}$ is the gravity vector in the global frame. Extrinsic between the LiDAR and the IMU is ${}^{I}_{L}\mathbf{T} = [{}^{I}_{L}\mathbf{R},\ {}^{I}_{C}\mathbf{t}]$ and between the camera and the IMU is ${}^{I}_{C}\mathbf{T} = [{}^{I}_{C}\mathbf{R},\ {}^{I}_{C}\mathbf{t}]$, which are assumed pre-calibrated and constant. Moreover, We define the jth timestamp with certain semantics z as t_j^z. The IMU local frame and state vector at t_j^z, for example, are represented by ${}^{z}_{j}(\cdot)$ and $\mathbf{x}_j^z$.

2.2 | Feature handler

The feature handler receives the incoming data from the LiDAR and the camera. Then it extracts different features from the LiDAR point clouds and camera images respectively, which will be used in the subsequent modules to construct measurements with the corresponding maps for pose estimation.

2.2.1 | Surf points extraction

From a raw LiDAR point cloud, we first extract surf points with high local smoothness [2] as LiDAR features and then downsample them with a uniform sampling filter [18] to ensure an even distribution. Note that this is done before point cloud undistortion (or rather, on the raw distorted point clouds) since we assume that ego-motion has a similar effect on neighboring points and will not change the local smoothness significantly.

2.2.2 | Corner points extraction

From an input image, we extract GFTT corner points [19] as visual features and track them in the following images using Kanade-Lucas-Tomasi optical-flow [20]. To ensure the track accuracy, the visual tracker works at the raw image rate (30 Hz) instead of the rate after synchronization (10 Hz).

2.3 | Synchronization handler

The goal of synchronization is to retrieve simultaneous data from three different sensors so that they can be fused. Considering hardware synchronization support is unavailable in most cases, we choose to align the data by their timestamps. Since image warp is inaccurate when pixel depth is unknown, we choose the image time as ESIKF update time and transform LiDAR and IMU data to that time. We denote the timestamp of the nth LiDAR (feature) point cloud as t_n^l, the kth image (feature frame) as t_k^c. The synchronization scheme is shown in Figure 2.

2.3.1 | Data association

When the (n+1)-th LiDAR feature cloud $\mathcal{L}_{n+1}$ with timestamp t_{n+1}^l is available, we associate it with the nearest image feature frame $\hat{\mathcal{C}}_{n+1}$ (namely $\mathcal{C}_{k+3}$ in Figure 2) whose timestamp is denoted as t_{n+1}^u (namely t_{k+3}^c in Figure 2) since the next ESIKF update will take place at this time. Image feature frames within $\left(t_n^u, t_{n+1}^u\right)$ will be discarded. Then, we collect all IMU data within $\left[t_{n+1}^s, t_{n+1}^e\right]$ where $t_{n+1}^s = \min\left\{t_n^l, t_n^u\right\}$ and $t_{n+1}^e = \max\left\{t_{n+1}^l, t_{n+1}^u\right\}$. It will be seen in Section 2.3.2 and 2.3.3 that these IMU data are necessary to propagate the state from t_n^u to t_{n+1}^u and transform all the feature points in $\mathcal{L}_{n+1}$ to t_{n+1}^u.

2.3.2 | State propagation

After data association, we first synchronize IMU data to t_{n+1}^u. We propagate the state from the last ESIKF update time t_n^u to t_{n+1}^u by IMU data collected before according to the following continuous kinematic model [11]:

$$\begin{aligned} {}^{G}_{I}\dot{\mathbf{R}} &= {}^{G}_{I}\mathbf{R}\left\lfloor \boldsymbol{\omega}_m - \mathbf{b}_g - \mathbf{n}_g \right\rfloor_{\times}, {}^{G}_{I}\dot{\mathbf{t}} = {}^{G}\mathbf{v}, \\ {}^{G}\dot{\mathbf{v}} &= {}^{G}_{I}\mathbf{R}(\boldsymbol{a}_m - \mathbf{b}_a - \mathbf{n}_a) + {}^{G}\mathbf{g}, \\ {}^{G}\dot{\mathbf{g}} &= \mathbf{0}, \dot{\mathbf{b}}_a = \mathbf{n}_{ba}, \dot{\mathbf{b}}_g = \mathbf{n}_{bg} \end{aligned} \tag{2}$$

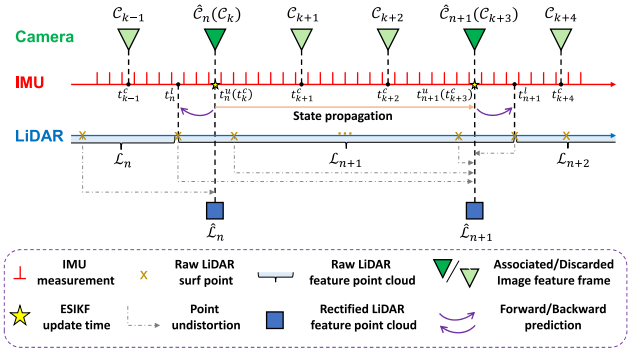


FIGURE 2 The illustration of the scheme of the synchronization handler. First, we associate each light detection and ranging (LiDAR) feature point cloud to the nearest image feature frame. Then to synchronize data from all sensors, we transform all LiDAR feature points and propagate the system state with inertial measurement unit (IMU) data to the associated image time, at which the error-state iterated Kalman filter (ESIKF) will conduct update and produce accurate odometry approximately at the LiDAR rate.

where $\lfloor\cdot\rfloor_\times$ represents the skew-symmetric matrix of the vector, $\boldsymbol{\omega}_m$, $\boldsymbol{a}_m$ are raw IMU accelerometer and gyroscope readings, $\mathbf{n}_a$, $\mathbf{n}_g$ are the Gaussian white noise of IMU measurements and $\mathbf{n}_{ba}$, $\mathbf{n}_{bg}$ are random walk noise of IMU biases.

During the propagation, we will produce the odometry at the IMU rate and finally obtain an Gaussian state prior $\hat{\mathbf{x}}^u_{n+1} \sim \mathcal{N}\left(\mathbf{x}^u_{n+1}, \hat{\mathbf{P}}^u_{n+1}\right)$ at t^u_{n+1}. Details of the discrete implementation of propagation can be seen in [11, 12].

2.3.3 | Point cloud undistortion

According to the LiDAR working principle, points in $\mathcal{L}_{n+1}$ are sampled at different time within $\left[t^l_n, t^l_{n+1}\right]$ and represented, respectively, in the local LiDAR frame of their sampling time. To synchronize them to t^u_{n+1}, we apply the following transformation to each raw point ${}^{L^{\mathcal{L}}_i}\mathbf{p}^l_i$ sampled at time $t^{\mathcal{L}}_i$:

$$ {}^{L^u_{n+1}}\mathbf{p}^l_i = {}^L_I\mathbf{T} \cdot {}^G_{I^u_{n+1}}\mathbf{T}^{-1} \cdot {}^G_{I^{\mathcal{L}}_i}\mathbf{T} \cdot {}^I_L\mathbf{T} \cdot {}^{L^{\mathcal{L}}_i}\bar{\mathbf{p}}^l_i \tag{3} $$

where ${}^G_{I^u_{n+1}}\mathbf{T}$, ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ are IMU poses at time t^u_{n+1} and $t^{\mathcal{L}}_i$, ${}^{L^{\mathcal{L}}_i}\bar{\mathbf{p}}^l_i$ is the homogeneous coordinate of ${}^{L^{\mathcal{L}}_i}\mathbf{p}^l_i$, and ${}^{L^u_{n+1}}\mathbf{p}^l_i$ is the synchronized point represented in local LiDAR frame of t^u_{n+1}. In Equation (3), we use ${}^G_{I^u_{n+1}}\hat{\mathbf{T}}$ retrieved from $\hat{\mathbf{x}}^u_{n+1}$ to approximate ${}^G_{I^u_{n+1}}\mathbf{T}$; thus, we only need to calculate ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ for each point:

- If $t^{\mathcal{L}}_i$ is within $\left[t^u_n, t^u_{n+1}\right]$, ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ can be linearly interpolated as

$$ {}^G_{I^{\mathcal{L}}_i}\mathbf{T} = {}^G_{I_l}\mathbf{T} \cdot \mathrm{Exp}(\lfloor s\delta\xi \rfloor_\times) \tag{4} $$

where, $s = \frac{t^{\mathcal{L}}_i - t_l}{t_r - t_l}$, $\delta\boldsymbol{\xi} = \mathrm{Log}\left({}^G_{I_l}\mathbf{T} - {}^1{}^G_{I_r}\mathbf{T}\right)$, t_l, t_r are the nearest IMU data time to $t^{\mathcal{L}}_i$ following $t_l \le t^{\mathcal{L}}_i \le t_r$, and ${}^G_{I_l}\mathbf{T}$, ${}^G_{I_r}\mathbf{T}$ are the corresponding odometry results at t_l, t_r produced during the state propagation.

- If $t^{\mathcal{L}}_i$ is earlier than t^u_n, ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ is the pose earlier than the latest updated pose ${}^G_{I^u_n}\mathbf{T}$. We can make a backward prediction of ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ by inversely propagating the ${}^G_{I^u_n}\mathbf{T}$ from t^u_n to $t^{\mathcal{L}}_i$ using the collected IMU data within $\left[t^s_{n+1}, t^u_n\right]$ according to Equation (2).
- If $t^{\mathcal{L}}_i$ is later than t^u_{n+1}, ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ is the pose later than the latest propagated pose ${}^G_{I^u_{n+1}}\mathbf{T}$. We can make a forward prediction of ${}^G_{I^{\mathcal{L}}_i}\mathbf{T}$ by further propagating ${}^G_{I^u_{n+1}}\hat{\mathbf{T}}$ to $t^{\mathcal{L}}_i$ using the collected IMU data within $\left[t^u_{n+1}, t^e_{n+1}\right]$ according to Equation (2).

Note that the 'propagation' mentioned above is independent of the filter and the filter state keeps fixed at $\hat{\mathbf{x}}^u_{n+1}$ during the whole undistortion process. After the undistortion of all points, we can develop the synchronized LiDAR feature point cloud $\hat{\mathcal{L}}_{n+1}$.

2.4 | Measurement handler

Based on the temporary state estimation ${}^{(\varkappa)}\tilde{\mathbf{x}}^u_{n+1}$ from the κ-th iteration of ESIKF update, the measurement handler selects valid LiDAR and visual measurements for the next iterated update to improve the accuracy and robustness of the state estimation.

2.4.1 | Point-to-plane measurements selection

Similar to [2], for a surf point ${}^{L^u_{n+1}}\mathbf{p}^l_i$ in $\hat{\mathcal{L}}_{n+1}$, we transform it to the global frame by pose in ${}^{(\varkappa)}\tilde{\mathbf{x}}^u_{n+1}$, searching for the nearest several points in the LiDAR map and fitting a plane. If the fitted plane is flat enough and the distance from the transformed point to the fitted plane is lower than a threshold, we can form a preliminary point-to-plane measurement:

$$ \begin{aligned} \mathbf{r}^l_i\left({}^G_{I^u_{n+1}}\mathbf{R}, {}^G_{I^u_{n+1}}\mathbf{t}\right) = {}^G\mathbf{n}^T_i \Big(& {}^G_{I^u_{n+1}}\mathbf{R}\left({}^I_L\mathbf{R}\,{}^{L^u_{n+1}}\mathbf{p}^l_i + {}^I_L\mathbf{t}\right) \\ & + {}^G_{I^u_{n+1}}\mathbf{t} - {}^G\mathbf{q}^l_i \Big) \end{aligned} \tag{5} $$

where ${}^G\mathbf{n}_i$ and ${}^G\mathbf{q}^l_i$ are the normal and the in-plane point of the fitted plane.

In practice, there are usually over 1000 preliminary point-to-plane measurements in outdoor scenarios, and we find that pose estimation is prone to be stuck in the local minima if too many measurements are fed to the ESIKF update. It means that some measurements are not constraining enough for pose estimation and they are more likely to introduce unwanted local minimum points than to improve the accuracy if utilized in the pose estimation. Inspired by [21, 22], we propose an efficient sampling method to select the most constraining point-to-plane measurements to facilitate a global optimal state estimation.

Essentially, Equation (5) is a concrete form of registering the source point cloud to the target point cloud using point-to-plane ICP [23]; thus, it can be written in the general form:

$$\mathbf{r}_i^l({}_s^t\mathbf{R}, {}_s^t\mathbf{t}) = {}^t\mathbf{n}_i^T({}_s^t\mathbf{R}{}^s\mathbf{p}_i^l + {}_s^t\mathbf{t} - {}^t\mathbf{q}_i^l) \tag{6}$$

By the perturbation of ${}_s^t\mathbf{R}, {}_s^t\mathbf{t}$ with $\boldsymbol{\delta\varphi} \in \mathfrak{so}(3), {}_s^t\mathbf{t}$ in (6), we have:

$$\begin{aligned}
&\mathbf{r}_i^l\left({}_s^t\mathbf{R} \cdot \mathrm{Exp}(\lfloor\boldsymbol{\delta\varphi}\rfloor_\times),\ {}_s^t\mathbf{t} + \boldsymbol{\delta t}\right) \\
&= {}^t\mathbf{n}_i^T\left({}_s^t\mathbf{R}(\mathbf{I} + \lfloor\boldsymbol{\delta\varphi}\rfloor_\times){}^s\mathbf{p}_i^l + {}_s^t\mathbf{t} + \boldsymbol{\delta t} - {}^t\mathbf{q}_i^l\right) \\
&= {}^t\mathbf{n}_i^T({}_s^t\mathbf{R}{}^s\mathbf{p}_i^l + {}_s^t\mathbf{t} - {}^t\mathbf{q}_i^l) + {}^t\mathbf{n}_i^T\boldsymbol{\delta t} + ({}^s\mathbf{p}_i^l \times {}^s\mathbf{n}_i)^T\boldsymbol{\delta\varphi} \\
&= \mathbf{r}_l^i({}_s^t\mathbf{R}, {}_s^t\mathbf{t}) + \mathbf{J}_\mathbf{t}^i\boldsymbol{\delta t} + \mathbf{J}_{\boldsymbol{\varphi}}^i\boldsymbol{\delta\varphi}
\end{aligned} \tag{7}$$

where ${}^s\mathbf{n}_i = {}^s\mathbf{R}_t{}^t\mathbf{n}_i + {}^s\mathbf{t}_t$ and is the target plane normal represented in the source frame.

From Equation (7), we can see ${}^t\mathbf{n}_i^T$ and $({}^s\mathbf{p}_i^l \times {}^s\mathbf{n}_i)^T$ are the Jacobians of the point-to-plane measurement with respect to the pose perturbation. Their magnitude of each component implies approximately how much the measurement residual will change if the corresponding pose component has a unit increment; hence, they can be seen as an indication of how much constraint a measurement can provide to each pose component. To get a subset of preliminary measurements, which in total can provide sufficient constraints for pose estimation, in each pose component, we sample those measurements providing the most constraints for this component and the details of our method are presented in Algorithm 1:

Algorithm 1 Sampling method of preliminary point-to-plane measurements

Input: Preliminary point-to-plane measurements $\mathbf{r}^l$; current LiDAR attitude estimation ${}^G_{L^u_{n+1}}\widetilde{\mathbf{R}}$;

Output: Sampled point-to-plane measurements $\bar{\mathbf{r}}^l$;

1 Initialize all elements of the array $C[6]$ with empty lists;
2 **for** *each measurement* $\mathbf{r}_i^l$ *in* $\mathbf{r}^l$ **do**
3 Compute $\mathbf{n}_i \leftarrow {}^G_{L^u_{n+1}}\widetilde{\mathbf{R}}^T{}^G\mathbf{n}_i$;
4 Compute $\mathbf{m}_i \leftarrow {}^{L^u_{n+1}}\mathbf{p}_i^l \times \mathbf{n}_i$;
5 Append $|(\mathbf{n}_i)_x|, |(\mathbf{n}_i)_y|, |(\mathbf{n}_i)_z|, |(\mathbf{m}_i)_x|, |(\mathbf{m}_i)_y|, |(\mathbf{m}_i)_z|$ respectively to $C[0], C[1], C[2], C[3], C[4], C[5]$;
6 **end**
7 **for** *each list* $C[k]$ *in* C **do**
8 Sort the items of $C[k]$ in desending order;
9 $cnt \leftarrow 0$;
10 **for** *each item* v_j *in* $C[k]$ **do**
11 **if** *the measurement* $\mathbf{r}_j^l$ *corresponding to* v_j *is not in* $\bar{\mathbf{r}}^l$ **then**
12 Add $\mathbf{r}_j^l$ to $\bar{\mathbf{r}}^l$;
13 $cnt \leftarrow cnt + 1$;
14 **if** $cnt = MaxSamples$ **then**
15 **goto** line 19;
16 **end**
17 **end**
18 **end**
19 **end**

Here, we have some comments on this algorithm:

- To ensure the robustness of our odometry in structureless environments, we perform the sampling only when the number of preliminary point-to-plane measurements is greater than a certain threshold (e.g., 600) and we will obtain $6 \cdot MaxSamples$ sampled measurements in total.
- In line 5 of the algorithm, we choose the normal in the local frame ($\mathbf{n}_i$) instead of the global frame (${}^G\mathbf{n}_i$) as the index of translation constraint as if we were registering to a local map according to Equation (7). Such an egocentric choice coincides with [10, 22] and is shown to be more robust in practice. Also, we here adopt the six indexes as in [21] instead of the nine indexes in [22], which considers the flatness of planes, because we find that these six indexes are robust and sufficient to indicate constraints. However, we do not project the index vectors $\mathbf{n}_i$, $\mathbf{m}_i$ to the eigenvectors of the information matrix as in [21] but directly perform greedy sampling on the Jacobian matrix like [22] since the former practice shows no gains in our experiments.
- Our sampling is performed on the measurements instead of on the raw point cloud as [22]. The normals are directly retrieved from the measurements instead of from time-consuming normal estimation of the raw point cloud. Thus, there is little computation in our method compared to [21, 22] and the main cost lies in the sort of six lists, which however is rather efficient in modern C++.
- Figure 3 illustrates the sampled measurements in both bird's-eye-view and side view. It can be seen that the sampled measurements distributed sparsely all around, which can provide sufficient constraints for the pose estimation and meanwhile reduce the possibility of falling into the local minima point.

2.4.2 | Point-to-pixel measurements selection

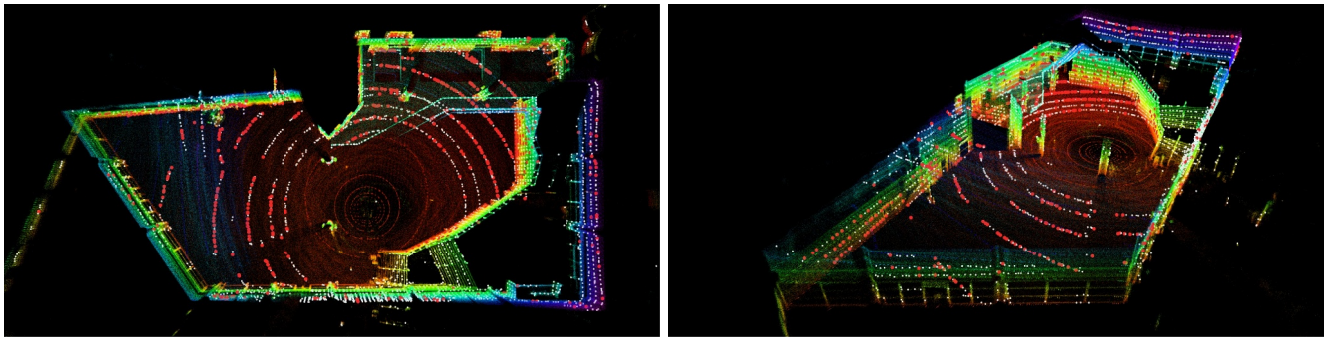
For a tracked corner point ${}^{\mathcal{C}}\mathbf{p}_i^c(u, v)$ in $\hat{\mathcal{C}}_{n+1}$, we first retrieve its corresponding 3D landmark point ${}^G\mathbf{p}_i^c$ in the visual map if it has been triangulated in the map handler. Then, we project ${}^G\mathbf{p}_i^c$ to $\hat{\mathcal{C}}_{n+1}$ by the pose in ${}^{(x)}\tilde{\mathbf{x}}^u_{n+1}$, obtaining a preliminary point-to-pixel measurement:

$$\mathbf{r}_i^c\left({}^G_{I^u_{n+1}}\mathbf{R}, {}^G_{I^u_{n+1}}\mathbf{t}\right) = \boldsymbol{\pi}\left({}^C_I\mathbf{R}{}^G_{I^u_{n+1}}\mathbf{R}^T\left({}^G\mathbf{p}_i^c - {}^G_{I^u_{n+1}}\mathbf{t}\right) + {}^C_I\mathbf{t}\right) - {}^{\mathcal{C}}\mathbf{p}_i^c \tag{8}$$

where $\boldsymbol{\pi}(\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$ represents the camera model. To reject outliers, we only select the measurements whose residuals are smaller than a certain threshold (e.g. 3 pixels) and add them to $\bar{\mathbf{r}}^c$.

2.5 | ESIKF update

In the aforementioned modules, we have obtained the state prior $\hat{\mathbf{x}}^u_{n+1}$ and the selected measurements $\bar{\mathbf{r}}^l, \bar{\mathbf{r}}^c$. Assuming each



(a) bird's-eye view (b) side view

FIGURE 3 An illustration of the sampled light detection and ranging (LiDAR) point-to-plane measurements. The colorful point cloud represents the accumulated LiDAR map. Small white points are the feature points constructing the preliminary point-to-plane measurements of the current frame. Big red points correspond to the measurements sampled according to the proposed method and will be used for the error-state iterated Kalman filter (ESIKF) update.

LiDAR and visual measurement is affected, respectively, by independent Gaussian white noise with zero mean and covariance matrix $\boldsymbol{\Sigma}_i^l, \boldsymbol{\Sigma}_j^c$, the maximum a posteriori estimation of the state $\mathbf{x}_{n+1}^u$ is:

$$\arg\min_{\mathbf{x}_{n+1}^u} \left\{ \|\mathbf{x}_{n+1}^u - \hat{\mathbf{x}}_{n+1}^u\|_{\hat{\mathbf{P}}_{n+1}^u}^2 + \sum_i \|\bar{\mathbf{r}}_i^l(\mathbf{x}_{n+1}^u)\|_{\boldsymbol{\Sigma}_i^l}^2 + \sum_j \|\bar{\mathbf{r}}_j^c(\mathbf{x}_{n+1}^u)\|_{\boldsymbol{\Sigma}_j^c}^2 \right\} \tag{9}$$

where $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}$. Note that measurement constraints from all of three sensors are included in Equation (9), meaning that our odometry system simultaneously uses the data from three sensors to estimate the system state at time t_{n+1}^u and therefore achieves a fully tightly couple of the sensors.

According to the Bayesian estimation theory, the solution of Equation (9) is the Kalman update for linear measurements. We iteratively linearize Equation (9) with respect to the estimated state and perform Kalman update as in [12] to reduce the linearized errors of the measurement models and improve the estimation accuracy. After each iteration, the current state estimation ${}^{(\kappa)}\tilde{\mathbf{x}}_{n+1}^u$ is fed back to the measurement handler to select new measurements. The iterated update is stopped if the state increment between two iterations is small enough or the maximum iteration time is reached.

2.6 | Map handler

The map handler is in charge of building and managing the LiDAR and the visual map based on the latest feature frames $\hat{\mathcal{L}}_{n+1}, \hat{\mathcal{C}}_{n+1}$ and the pose estimation $\mathbf{T}_{n+1}^u$.

2.6.1 | LiDAR mapping

We use ikd-Tree [12] to efficiently build and manage the LiDAR map. When $\hat{\mathcal{L}}_{n+1}$ arrives, we transform all of its surf points with the estimated pose $\mathbf{T}_{n+1}^u$ to the global frame and add them to the ikd-Tree, which will update itself incrementally and support the efficient nearest searching operations from the measurement handler.

2.6.2 | Visual mapping

The visual map consists of the 3D points corresponding to the visual features tracked in the latest frame. As in [7], we maintain a sliding window recording the past few image feature frames along with global poses. When a new frame $\hat{\mathcal{C}}_{n+1}$ arrives, we first push it into the sliding window along with its pose and discard the oldest frame. For each visual map point ${}^G\mathbf{p}_i^c$ observed in $\hat{\mathcal{C}}_{n+1}$, a feature-only optimization is performed by minimizing the total reprojection errors Equation (7) on all its observed frames in the window to improve mapping accuracy. Points with large average reprojection errors (e.g. 5 pixels) are deemed as outliers and deleted from the visual map. Also, points not tracked in $\hat{\mathcal{C}}_{n+1}$ will be removed since it cannot serve as the landmark any more. For features observed at least twice but not corresponding to any map point, we triangulate them to generate new visual map points.

3 | EXPERIMENTS

In this section, we evaluate the performance of FT-LVIO on both public and private datasets. Also, an ablation study is conducted to demonstrate the effects of the point cloud undistortion, our proposed sampling method and the introduction of the vision. In the end, we analyze the running time of the whole system.

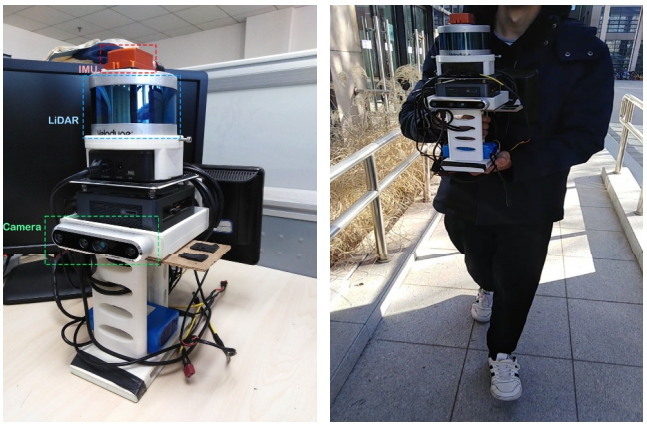
3.1 | NTU VIRAL dataset

First, we evaluate FT-LVIO on the nine sequences of the public NTU VIRAL dataset [24], which is collected on the NTU campus by an aerial platform equipped with multiple

sensors. We compare our method with various state-of-the-art multi-sensor odometry or SLAM systems, including Visual-Inertial (VI) systems like VINS-Mono, VINS-Fusion [25], LiDAR-Inertial (LI) systems like LIO-SAM, FAST-LIO2, LiDAR-Visual (LV) systems like DVL-SLAM [26] and LiDAR-Visual-Inertial (LVI) systems like R2LIVE, VIRAL-SLAM [27] and FAST-LIVO. For all the sequences, our FT-LVIO uses the data from the left camera (10 Hz), the horizontal LiDAR (10 Hz) and the main IMU (385 Hz). Since the camera data have the same rate with the LiDAR data but are unsynchronized with them, we set a maximum time offset of 0.04s (slightly lower than half of the data period) in the synchronization handler. It means that if a LiDAR feature frame is associated with an image feature frame with a timestamp difference over 0.04s, we reject this association and only use IMU and LiDAR data in the pose estimation for this frame. This can help prevent from suffering from errors caused by bad and uncertain synchronization. Besides, we set the *MaxSamples* of Algorithm 1 to 100.

Absolute Trajectory Errors of the aforementioned methods are presented in Table 1, where the results for VINS-Mono, VINS-Fusion, LIO-SAM and VIRAL-SLAM are directly from [27], and the results for DVL-SLAM, FAST-LIO2, R2LIVE and FAST-LIVO are from [16]. We can see that our FT-LVIO outperforms other methods on all sequences. VIRAL-SLAM, as another system with a fully tightly coupled framework, produces the second best results. The gap between the performance of FT-LVIO and VIRAL-SLAM can be mainly attributed into two factors: (1) The LiDAR observations of VIRAL-SLAM are constructed with a local map formed by some nearest frames instead of with a global map like FT-LVIO do; thus, VIRAL-SLAM is likely to produce a larger drift compared to FT-LVIO; (2) VIRAL-SLAM synchronizes data from all sensors to the LiDAR time and uses the estimated pixel velocity to compensate the time delay for the image while FT-LVIO chooses to transform LiDAR points to the image time for synchronization. The accuracy of the former is usually less than that of the latter since the former is based on the velocity estimation from the past image frames, while the latter directly uses the estimated pose from timely IMU data.

It is worth noting that even though FAST-LIVO and R2LIVE fuse the data from three kinds of sensors, their performance is apparently worse than LIO-SAM, which only fuses the data from the LiDAR and the IMU. The reason here is twofold: (1) FAST-LIVO and R2LIVE are just odometry systems and they are lack of the loop closure module as LIO-SAM has to reduce the long-run drifts; (2) FAST-LIVO and R2LIVE do not realize a fully tightly couple of all the sensors since their systems essentially consist of an LIO subsystem and a VIO subsystem. Such framework leaves them susceptible to the degeneration of a certain subsystem. And on this dataset, the VIO subsystem is prone to degeneration because the UAV sometimes performs fast motion and blurs the images; thus, the LIO subsystem of FAST-

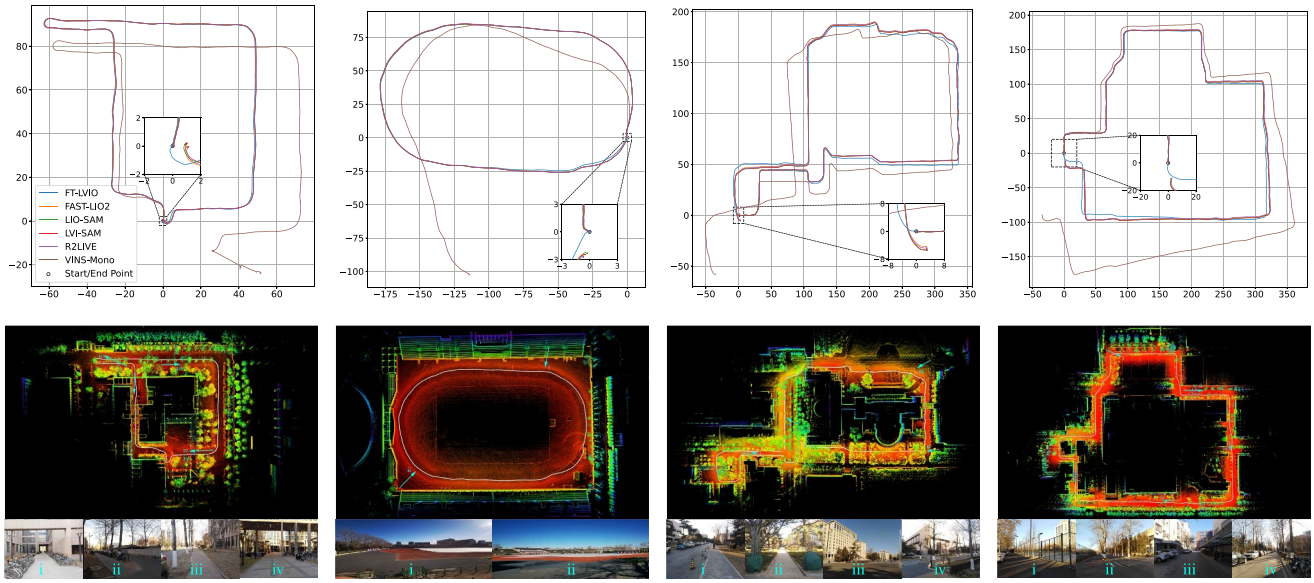


(a) Handheld device (b) Data-gathering process

FIGURE 4 The illustration of our handheld device and data-gathering process.

TABLE 1 Absolute Trajectory Errors (ATE) (units: meters) of various methods over the NTU VIRAL Dataset. The best result for each sequence is highlighted in **bold**, and the second best is underlined. The last line calculates the average ATE for each method on all the sequences

	VI Systems		LI Systems		LV Systems	LVI Systems			
Sequence	**VINS-mono**	**VINS-fusion**	**LIO-SAM**	**FAST-LIO2**	**DVL-SLAM**	**R2LIVE**	**VIRAL-SLAM**	**FAST-LIVO**	**FT-LVIO**
eee_01	0.568	0.306	0.075	0.540	2.880	0.450	0.060	0.280	**0.031**
eee_02	0.443	0.266	0.069	0.220	1.650	0.210	0.058	0.170	**0.020**
eee_03	0.886	0.383	0.101	0.250	3.080	0.970	0.037	0.230	**0.026**
nya_01	0.830	0.237	0.076	0.240	2.090	0.190	0.051	0.190	**0.026**
nya_02	0.422	0.297	0.090	0.210	1.450	0.630	0.043	0.180	**0.029**
nya_03	0.501	0.368	0.137	0.230	1.820	0.310	0.032	0.190	**0.029**
sbs_01	3.739	0.372	0.089	0.250	1.080	0.560	0.048	0.290	**0.026**
sbs_02	0.890	0.369	0.083	0.260	2.310	0.240	0.062	0.220	**0.025**
sbs_03	0.802	0.276	0.140	0.240	2.230	0.440	0.054	0.220	**0.026**
Average	1.009	0.319	0.096	0.271	2.066	0.444	0.049	0.219	**0.026**



(a) *WQ* (b) *EP* (c) *MB* (d) *RM*

FIGURE 5 We evaluate the performance of various methods on 4 outdoor sequences *WQ*, *EP*. *MB* and *RM* of our private handheld dataset. The top line shows the estimated trajectories of all the methods in bird's-eye-view. The trajectory direction is clockwise for *WQ*, *RM* and counterclockwise for *EP*, *MB*. The bottom line displays the mapping results (rendered by elevation) of our proposed FT-LVIO and the corresponding real scenarios. The white lines are the estimated trajectories and the cyan arrows represent the shooting locations and directions of the associated images.

LIVO and R2LIVE can be affected by the bad estimation results from the VIO subsystem, making the gain of fusing camera data not clear.

Counterintuitively, the LV system DVL-SLAM produces the worst results on most sequences even compared to the VI systems VINS-Mono and VINS-Fusion. The reason is due to that DVL-SLAM is more like a visual SLAM system because LiDAR data are just used to provide sparse depth information to the images. It cannot handle the fast motion of the UAV well without inertial sensors because the severely distorted point clouds will introduce significant depth errors to the image pixels.

3.2 | Private handheld dataset

In this section, we use our handheld device shown in Figure 4a to gather dataset for further evaluation. Our handheld device consists of a Velodyne VLP-16 LiDAR, an Intel RealSense D455 camera and a Xsens MTi-300 IMU. We use RealSense RGB frames with the resolution of 1280 × 720 as the image inputs of the system. The data rate of the LiDAR, camera and IMU is 10, 30 and 200 Hz, respectively. All of 5 sequences in our dataset are collected by an operator holding the device at chest height (Figure 4b) and walking around the campus of Tsinghua University, Beijing, China. Due to the lack of a high-precision RTK or motion capture suite for ground truth, all sequences start and end at the same position and we calculate the start-to-end translation errors to represent the odometry drifts. We compare FT-LVIO to open-sourced LI systems (LIO-SAM, FAST-LIO2), VI systems (VINS-Mono) and LVI systems (LVI-SAM, R2LIVE). For FT-LVIO, we set *Max-Samples* of Algorithms 1–5 on outdoor sequences *WQ*, *EP*, *MB*, *RM* and deactivate the sampling on indoor sequence *SC* due to point cloud sparsity. Also, the loop closure of LIO-SAM, LVI-SAM and VINS-Mono is switched off for fairness. All the methods are implemented in C++ and tested on a laptop with Intel i7-11800H CPU and 16 GB RAM in Ubuntu Linux. Furthermore, the results on *EP*, *MB* and *SC* can be found in the video attachment[1].

3.2.1 | Outdoor short-distance experiments

In this group of experiments, we show the robustness of our proposed FT-LVIO after a short journey with sensor degradation. We consider two sequences *WQ* (∼389m) and *EP* (∼495m). *WQ* is collected in the vicinity of the Weiqing Building. The camera suffers from degradation when we walk in and out of a narrow corridor (indicated by the cyan arrow (ii) in Figure 5a) due to the sharp change of illumination. *EP* is collected by walking around the open East Playground, where the track lines and fences exhibit similar visual features and the LiDAR sometimes degrades in certain directions due to point cloud sparsity.

From the results shown in Table 2, we can see that benefit from the fully tightly couple of all the sensors, our FT-LVIO almost closes the loops with a drift of only 0.174 and 0.054 m on these two sequences, demonstrating its accuracy and robustness in spite of sensor degradation. VINS-Mono drifts significantly because the camera sometimes suffers from degradation due to the lack of visual features or the

[1] https://cloud.tsinghua.edu.cn/f/41d3adf10311491d8bdf/

TABLE 2 Start-to-end errors (units: meters) over the private handheld dataset

	VI Systems	LI Systems		LVI Systems		
Sequence	**VINS-mono**	**LIO-SAM**	**FAST-LIO2**	**LVI-SAM**	**R2LIVE**	**FT-LVIO**
WQ	56.764	0.986	1.029	1.289	1.143	**0.174**
EP	153.241	2.375	2.708	2.542	2.835	**0.054**
MB	72.664	7.499	5.682	7.737	6.089	**0.085**
RM	101.307	12.076	11.428	12.270	11.301	**0.051**
SC	7.053	Fail	1.574	1.341	1.751	**0.128**
Average	78.206	5.734	4.484	5.036	4.624	**0.098**

Note: The best result for each sequence is highlighted in bold, and the second best is underlined.

presence of many similar features. Two LI systems, LIO-SAM and FAST-LIO2 also present considerable drifts. Interestingly, both LVI-SAM and R2LIVE produce larger errors than their LIO subsystems namely LIO-SAM and FAST-LIO2, meaning that their system performances deteriorate with the introduction of the camera. This can be attributed to their two-subsystem framework.

3.2.2 | Outdoor long-distance experiments

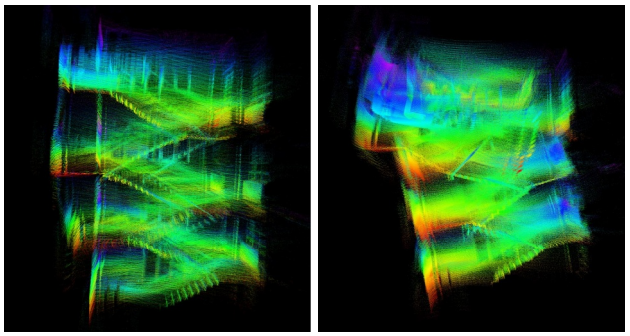
In this group of experiments, we show that our FT-LVIO is low-drift after long journey in environments containing abundant facades and vegetations. We consider two sequences *MB* (~1054m) and *RM* (~1130m), which are collected near the Tsinghua Main Building and the RHOM Building, respectively. As shown in Table 2, FT-LVIO presents a drift of only 0.085 m on *MB*, while the second least drift produced by FAST-LIO2 is as large as 5.682 m. LVI-SAM and R2LIVE again produce larger errors than their LIO subsystems. Performance gap is more prominent on *RM*, where FT-LVIO successfully closes the loop even after a long journey, while all of the other methods produce drifts over 10 m Figure 5 shows the estimated trajectories of different methods and the mapping results of FT-LVIO, from which we can further see how FT-LVIO outperforms other methods on various outdoor scenarios.

3.2.3 | Indoor degradation experiment

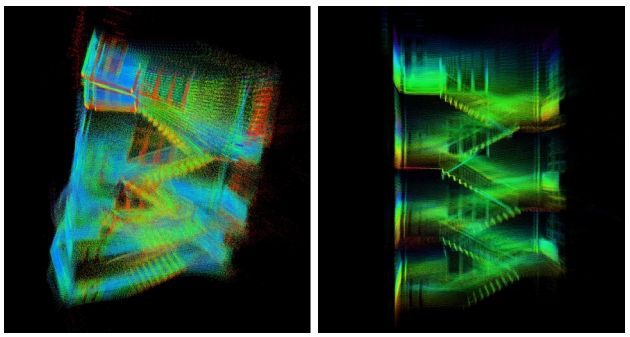
In this experiment, we challenge the difficult staircase scenario in the Weiqing Building of Tsinghua University (Figure 6a). The main challenges are: (1) the surrounding walls are nearly pure white with few visual features; (2) the LiDAR point clouds are sparse especially in the vertical direction, and the body of our operator can also block some points horizontally; (3) the windows on the wall can further reduce the density and accuracy of the point cloud; (4) the whole trajectory contains many sharp turns of 180°, which can cause the loss of visual features and the severe distortion of the point clouds.

The dataset *SC* (~93m) is gathered by walking down the stairs from the 12th floor to the 8th floor and then walking

(a) Staircase environment of the Weiqing Building, where the narrow corridors, white walls and large windows can cause severe sensor degradation.



(b) FAST-LIO2 (c) R2LIVE



(d) LVI-SAM (e) FT-LVIO

FIGURE 6 The illustration of the challenging staircase environment and the corresponding mapping results of some methods.

upstairs to the origin. The odometry and mapping results of different methods are shown in Table 2 and Figure 6, respectively. We can see that VINS-Mono has a poor performance in this visual-unfriendly environment as expected. LIO-SAM fails on this dataset since its IMU data are just used to undistort point clouds and provide preintegration factors to the backend factor graph but are not used in the pose estimation of the odometry. Such a loose couple of IMU and LiDAR data is not beneficial for the system to survive in the environment where the LiDAR degrades significantly. In contrast, FAST-LIO2 achieves a tight couple of LiDAR and IMU and does not fail on this dataset, but it still presents considerable drifts. With the aid of vision, LVI-SAM does not fail like LIO-SAM, while its odometry and mapping results are far from satisfactory. Similarly, R2LIVE survives but its performance does not improve with the introduction of the vision compared with FAST-LIO2. Making full advantage of complimentary characteristics of individual sensors by a fully tightly coupled framework, our FT-LVIO has a drift of only 0.128 m and builds a consistent map of the staircase. This experiment confirms the robustness of FT-LVIO in the scenario of severe sensor degradation.

3.3 | Ablation study

In this study, we explore the effects of the point cloud undistortion (U), the proposed LiDAR point-to-plane measurement sampling method (S0) and the introduction of the vision on the performance of FT-LVIO. We denote the subsystem without S0 and all the visual-related modules (V, i.e. the green modules in the Figure 1) as the backbone (B), which is actually an LIO system like FAST-LIO2. We consider: (1) removing U from B (B - U); (2) adding S0 and V, respectively, to B (B + S0, B + V); (3) adding S0 and V simultaneously to B (B + S0 + V). Besides the aforementioned NTU and private datasets, we also conduct the ablation study on KITTI dataset [28] to make the results more convincing. Since IMU data are unavailable on the KITTI dataset, we use the constant motion model for state propagation. And we set *MaxSamples* of S0 to 100 for KITTI. The total results of the study are summarized in Table 3.

3.3.1 | Effects of the undistortion

In Table 3, the fourth column shows the results of the backbone as a baseline. The third column lists the results after disabling the point cloud undistortion for NTU and private datasets, and the corresponding results for KITTI dataset are omitted since KITTI directly provides the undistorted point clouds. It is clear that the errors increase over two times on both datasets without undistortion, which highlights the adverse effects that the distorted point clouds have for the system accuracy. Hence, the point cloud undistortion is essential to high-accuracy odometry on platforms moving unsteadily like UAVs and handheld devices in these datasets.

3.3.2 | Effects of the vision

The results after adding visual modules to the backbone correspond to the eighth column in Table 3. It can be seen that the NTU and the private dataset each have two sequences obtaining the improved results with the vision, but the improvements are slight. Things are better for the KITTI dataset since there are 7 out of 11 sequences that are improved and the average ATE is reduced by around 0.18 m due to the vision. Furthermore, we can find that sequences of long distance benefit most from the vision. For the KITTI dataset, the apparently improved sequences (00, 02, 05, 08, 09) are all with a length of over 1.7 km. And for the private dataset, the results are also improved only on two long-distance sequences *RM* and *MB*. The reason for this phenomenon is due to that the LiDAR measurements greatly outnumber the visual measurements and thus dominate the pose estimation. For short-distance sequences, LiDAR odometry is usually robust in the whole process and can solely provide an accurate pose estimation (just like on NTU dataset), making the introduction of the vision unnecessary. However, for long-distance sequences, LiDAR odometry is liable to drift slowly if not aided with the vision. So, the introduction of the vision can somewhat help reduce the drift rate but its function is limited since the whole pose estimation is still dominated by LiDAR.

3.3.3 | Effects of the sampling

The fifth column in Table 3 shows the results after performing point-to-plane measurements sampled with the proposed method (S0) on the basis of the backbone system. For comparison, we also implement other two sampling methods in [21, 22] and name them S1 and S2, respectively. The results for S1 and S2 are listed in the sixth and the seventh columns. Note that the original methods in [21, 22] require the normal estimation for the source point cloud, while this operation is absent in our system, so we use the normals in the point-to-plane measurements as a replacement in the implementation of S1 and S2. Besides, we make sure that the sampling number is equal for all the sampling methods.

In summary, there are, respectively, 5 out of 9 sequences on the NTU dataset, 10 out of 11 sequences on the KITTI dataset and 3 out of 4 sequences on the private dataset benefiting from our sampling method. Moreover, the average error is reduced, respectively, by 0.001 , 0.441 and 1.848 m. This result confirms that the proposed methods can certainly improve the odometry accuracy even without the vision, and the reason is due to that the sampled point-to-plane measurements contain fewer local minima points while maintaining good constraints for the pose estimation. Similar to our method, S2 also results into the improvement of average accuracy on all the datasets, while its contribution is rather limited. As for S1, it mainly produce gain on the KITTI dataset and the gain is smaller than that of our method.

TABLE 3 Results of the ablation study. The results are presented with Absolute Trajectory Errors (ATE) (units: meters) for NTU and KITTI datasets and with start-to-end errors (units: meters) for the private dataset

		Undistortion	-	Sampling			Vision	Sampling + Vision		
Dataset	**Sequence**	**B - U**	**B**	**B + S0**	**B + S1**	**B + S2**	**B + V**	**B + S0 + V**	**B + S1 + V**	**B + S2 + V**
NTU	eee_01	0.073	**0.028**	0.032	**0.033**	0.033	0.029	0.031	**0.033**	0.032
	eee_02	0.060	0.021	0.021	0.027	0.022	0.022	**0.020**	0.027	0.021
	eee_03	0.096	0.027	0.028	0.028	0.026	0.027	0.026	0.027	**0.025**
	nya_01	0.056	0.032	**0.025**	0.032	0.026	0.032	**0.025**	0.033	0.026
	nya_02	0.078	0.032	**0.029**	0.034	0.030	0.031	**0.029**	0.032	0.031
	nya_03	0.071	0.032	0.029	0.032	0.029	0.032	**0.028**	0.032	**0.028**
	sbs_01	0.068	0.029	0.028	0.030	0.026	0.028	0.026	0.027	**0.025**
	sbs_02	0.059	0.026	0.025	0.026	**0.024**	0.026	0.025	0.026	**0.024**
	sbs_03	0.061	0.026	0.026	0.026	**0.025**	0.026	0.026	0.026	**0.025**
	Average	0.070	0.028	0.027	0.030	0.027	0.028	**0.026**	0.029	**0.026**
KITTI	00	-	3.823	3.382	3.104	2.585	3.020	**2.532**	3.119	2.827
	01	-	18.922	15.803	15.850	18.763	18.888	15.767	**15.615**	18.644
	02	-	10.014	9.954	10.619	11.316	**9.403**	9.668	11.277	11.449
	03	-	0.978	**0.879**	0.954	0.967	0.975	0.919	0.914	0.928
	04	-	0.464	0.410	**0.397**	0.403	0.467	0.421	0.435	0.410
	05	-	1.681	1.267	1.395	1.341	1.324	1.256	1.387	**1.231**
	06	-	0.876	0.781	**0.729**	0.785	0.887	0.786	0.741	0.785
	07	-	0.471	0.477	**0.439**	0.895	0.818	0.464	0.501	0.495
	08	-	4.179	3.874	3.916	3.817	3.730	**3.425**	4.044	4.213
	09	-	1.786	**1.582**	1.602	2.186	1.627	1.723	2.213	1.933
	10	-	1.744	1.679	1.990	1.774	1.832	1.721	1.634	**1.601**
	Average	-	4.085	3.644	3.727	4.075	3.906	**3.516**	3.807	4.047
Private	*WQ*	1.987	0.747	0.179	0.829	**0.168**	0.753	0.174	0.387	0.661
	EP	5.711	2.467	1.602	**0.013**	2.431	2.549	0.054	0.694	2.613
	MB	17.770	4.846	5.725	5.790	6.846	4.739	**0.085**	6.188	3.363
	RM	15.487	10.215	3.376	14.767	7.927	10.186	**0.051**	11.967	7.647
	Average	10.239	4.569	2.721	5.350	4.343	4.557	**0.091**	4.809	3.571

Note: The best result for each sequence is highlighted in bold.

3.3.4 | Combined effects

When simultaneously utilizing the vision and the proposed sampling method, it forms the complete FT-LVIO system and the corresponding results are presented in the ninth column of Table 3. It is clearly seen that the results on all the sequences except eee_01 and sbs_03 are improved compared to that of the backbone system and the decrement of average error is, respectively, 0.002 , 0.569 and 4.478 m for three datasets. It means that the combined effects of the vision and the proposed sampling method are stronger than their individual effects, which can be attributed to the balance and the complementation of different sensors. However, we must admit that the results of B + S0 on the KITTI sequence 03, 08, 09 and 10 are clearly better than that of B + S0 + V, which warn us that there may exist better methods to fuse LiDAR and camera data. Also, Table 3 lists the results of combining S1 and S2 with the vision in the last two columns. Except from S1 on the KITTI dataset, the introduction of the vision brings about gains for these two sampling methods, while the final average errors of them are still larger than those of our method on all the datasets. So far, we can conclude that the proposed sampling method is more efficient and robust than other two methods regardless of the vision.

3.4 | Running time analysis

Table 4 shows the average per-frame time consumption of each FT-LVIO module on the aforementioned datasets.

TABLE 4 Average per-frame time consumption of each system module (units: milliseconds)

Module	Submodule	NTU	KITTI	Private
Feature handler	Surf points extraction	3.42	25.06	4.04
	Corner points tracker	7.37	9.92	18.29
Synchronization handler	Data association	0.33	0.00	0.01
	State propagation	0.33	0.00	0.20
	Point cloud undistortion	1.98	3.12	2.29
Measurement handler	Point-to-plane measurements selection (Sampling)	21.65 (1.08)	26.37 (1.10)	20.48 (0.99)
	Point-to-pixel measurements selection	0.02	0.02	0.04
ESIKF update		0.56	0.28	0.33
Map handler	LiDAR mapping	0.60	1.90	1.40
	Visual mapping	1.27	1.07	2.18
Others		0.12	0.09	0.26
Total time		**30.26**	**57.92**	**31.24**

Considering NTU and KITTI are both outdoor datasets, we do not include the indoor sequence *SC* in the computation of the running time for our private dataset. Besides, the time of the corner points tracker is excluded from the total time for the tracker running in a separate process.

It can be seen that the main time cost lies in the surf point extraction and point-to-plane measurement selection since the former needs point-by-point smoothness calculation and the latter requires frequent nearest search operations. The proposed sampling method consumes around 1 ms on average, exhibiting its efficiency as expected. Also, the corner points tracker can perform the real-time visual feature track even facing the 30 Hz large resolution (1280 × 720) images of our private dataset. So in general, FT-LVIO is able to perform the whole state estimation in real time with respect to the 10 Hz LiDAR input in outdoor environments either for the 16-channel LiDAR of NTU and private datasets (consuming around 30 ms pre frame) or for the 64-channel LIDAR of KITTI dataset (consuming around 60 ms per frame).

4 | CONCLUSION

Ego-motion estimation is an essential task for autonomous robots, and multi-sensor fusion-based methods recently have attracted lots of attention due to their accuracy and robustness. In this paper, we propose a fully tightly coupled multi-sensor fusion odometry framework termed FT-LVIO within the framework of an error-state iterated Kalman filter. To take full advantage of the complimentary characteristics of individual sensors, measurements from the LiDAR, the monocular camera and the IMU are synchronized to the same time and update the filter simultaneously to achieve an accurate pose estimation. Moreover, an efficient sampling method for LiDAR point-to-plane measurements is proposed to select those measurements, providing the most constraints for a global optimal state estimation.

According to the extensive experiments on both the public NTU dataset and private handheld dataset, the proposed FT-LVIO is able to achieve real-time state estimation and outperforms the state-of-the-art methods. It also shows robustness and accuracy in scenarios with severe sensor degradation. Furthermore, our ablation study demonstrates that the direct fusion of visual measurements and preliminary LiDAR point-to-plane measurements helps little in improving performance compared to the raw LIO system since the latter are great in number and dominate the state estimation. Meanwhile, the study confirms that the proposed sampling method is beneficial for the system to avoid local minima points and result in an accurate state estimation, and the performance can be apparently improved if the visual measurements are fused with the sampled LiDAR measurements as our FT-LVIO do.

Currently, the sampling number of the LiDAR measurements is pre-determined by a series of tuning attempts and fixed throughout the whole sequence, which is not elegant and may not be optimal for the specific sequence. In the future, we will explore the adaptive method to determine the sampling number automatically according to the environment.

AUTHOR CONTRIBUTIONS

Zhuo Zhang: Software; Validation; Visualization; Writing – original draft. **Zheng Yao** (Corresponding Author): Conceptualization; Funding acquisition; Methodology; Project administration; Writing – review and editing. **Mingquan Lu**: Writing – review and editing.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (NSFC), under Grant 42274018, and the Beijing National Research Center for Information Science and Technology under Grant BNR2021RC01015.

CONFLICT OF INTEREST

We declare that we have no conflict of interest.

DATA AVAILABILITY STATEMENT

Data are available on request from the corresponding author.

ORCID

Zheng Yao https://orcid.org/0000-0002-7657-644X

REFERENCES

1. Mur Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. IEEE Trans. Robot. 31(5), 1147–1163 (2015). https://doi.org/10.1109/tro.2015.2463671
2. Zhang, J., Singh, S.: LOAM: lidar odometry and mapping in real-time. In: Robotics: Science and Systems (2014)
3. Liu, W., et al.: TLIO: tight learned inertial odometry. IEEE Rob. Autom. Lett. 5(4), 5653–5660 (2020). https://doi.org/10.1109/lra.2020.3007421
4. Lynen, S., et al.: A robust and modular multi-sensor fusion approach applied to mav navigation. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3923–3929 (2013)
5. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: Proceedings 2007 IEEE International Conference on Robotics and Automation, pp. 3565–3572 (2007)
6. Leutenegger, S., et al.: Keyframe-based visual-inertial slam using nonlinear optimization. In: Robotics: Science and Systems (2013)
7. Qin, T., Li, P., Shen, S.: VINS-Mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. 34(4), 1004–1020 (2018). https://doi.org/10.1109/tro.2018.2853729
8. Shan, T., et al.: LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5135–5142 (2020)
9. Ye, H., Chen, Y., Liu, M.: Tightly coupled 3d lidar inertial odometry and mapping. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 3144–3150 (2019)
10. Qin, C., et al.: LINS: a lidar-inertial state estimator for robust and efficient navigation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 8899–8906 (2020)
11. Xu, W., Zhang, F.: FAST-LIO: a fast, robust lidar-inertial odometry package by tightly-coupled iterated Kalman filter. IEEE Rob. Autom. Lett. 6(2), 3317–3324 (2021). https://doi.org/10.1109/lra.2021.3064227
12. Xu, W., et al.: FAST-LIO2: Fast Direct Lidar-Inertial Odometry. ArXiv (2021). abs/2107.06829
13. Zhang, J., Singh, S.: Laser-visual-inertial odometry and mapping with high robustness and low drift. J. Field Robot. 35(8), 1242–1264 (2018). https://doi.org/10.1002/rob.21809
14. Shan, T., et al.: LVI-SAM: tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 5692–5698 (2021)
15. Lin, J., et al.: R^2LIVE: a robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping. IEEE Rob. Autom. Lett. 6(4), 7469–7476 (2021). https://doi.org/10.1109/lra.2021.3095515
16. Zheng, C., et al.: FAST-LIVO: Fast and Tightly-Coupled Sparse-Direct Lidar-Inertial-Visual Odometry. ArXiv (2022). abs/2203.00893
17. Zuo, X., et al.: LIC-Fusion: lidar-inertial-camera odometry. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5848–5854 (2019)
18. Rusu, R.B., Cousins, S.: 3d is here: point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4 (2011)
19. Shi, J., Tomasi, C.: Good features to track. In: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
20. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI (1981)
21. Gelfand, N., et al.: Geometrically stable sampling for the icp algorithm. In: Fourth International Conference on 3-D Digital Imaging and Modeling, 2003 3DIM 2003 Proceedings, pp. 260–267 (2003)
22. Deschaud, J.E.: IMLS-SLAM: scan-to-model matching based on 3d data. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2480–2485 (2018)
23. Low, K.L.: Linear Least-Squares Optimization for Point-To-Plane Icp Surface Registration (2004)
24. Nguyen, T.M., et al.: NTU VIRAL: a visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. Int. J. Robot Res. 41(3), 270–280 (2022). https://doi.org/10.1177/02783649211052312
25. Qin, T., et al.: A General Optimization-Based Framework for Local Odometry Estimation with Multiple Sensors. ArXiv (2019). abs/1901.03638
26. Shin, Y.S., Park, Y.S., Kim, A.: DVL-SLAM: sparse depth enhanced direct visual-lidar slam. Aut. Robots 44(2), 115–130 (2020). https://doi.org/10.1007/s10514-019-09881-0
27. Nguyen, T.M., et al.: VIRAL SLAM: Tightly Coupled Camera-Imu-Uwb-Lidar Slam. ArXiv (2021). **abs/2105**.03296
28. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361 (2012)

How to cite this article: Zhang, Z., Yao, Z., Lu, M.: FT-LVIO: Fully Tightly-coupled LiDAR-Visual-Inertial odometry. IET Radar Sonar Navig. 1–13 (2023). https://doi.org/10.1049/rsn2.12376