

LVIO-Fusion: Tightly-Coupled LiDAR-Visual-Inertial Odometry and Mapping in Degenerate Environments

Hongkai Zhang¹, Liang Du¹, Sheng Bao¹, *Member, IEEE*, Jianjun Yuan¹, *Senior Member, IEEE*, and Shugen Ma², *Fellow, IEEE*

Abstract—In this letter, we present an innovative, tightly-coupled LiDAR-Visual-Inertial Odometry and Mapping framework, termed LVIO-Fusion, which achieves robust and precise state estimation and map construction in environments characterized by LiDAR-degenerated and texture-less. The LVIO-Fusion is comprised of two subsystems: a LiDAR-inertial odometry (LIO) and visual-inertial odometry (VIO). The LIO subsystem employs a dynamic voxel mapping method by utilizing a Hash table and octrees for building and updating point cloud map directly and efficiently. The map points are directly projected to images to build the optical flow matching, which couples LiDAR and camera measurements in a deeper level. Then, the VIO subsystem proposes a coarse-to-fine state estimation that minimizes the frame-to-frame reprojection errors and photometric errors to align images. Furthermore, the VIO subsystem leverages online photometric calibration to acquire the true radiometric information from the environment, consequently refining the precision of the system. We extensively evaluate our method in challenge environments. The results validate the high accuracy and robustness of our method when compared to other state-of-the-art methods in LiDAR-degenerated and texture-less scenarios.

Index Terms—LiDAR-degenerated, mapping, SLAM, state estimation, texture-less.

I. INTRODUCTION

SIMULTANEOUS localization and mapping (SLAM) is a fundamental capability of mobile robots. Due to its real-time capacity to estimate six degrees of freedom (DoF) egomotion, SLAM is indispensable in various application, which includes robotic navigation, autonomous driving cars, Unmanned

Aerial Vehicles, etc. Existing SLAM methods have been demonstrated satisfactory performance in environments that have abundant textures or geometric features by employing single sensors such as camera [1], [2], [3] or light detection and ranging (LiDAR) sensor [4], [5], [6]. Visual SLAM is the technology that minimizes reprojection errors of tracked feature points or photometric errors of pixels to estimate camera motion. Currently, visual SLAM can attain satisfactory localization accuracy in scenarios with ample texture variation and stable, abundant lighting. By triangulating disparities from multi-view images, visual SLAM can recover depth information of pixels points, and reconstruct environmental maps. However, because of the calibration errors in camera parameters and the baseline of multi-view images, it is easy to cause errors in the depth estimation of map points. Additionally, because of the observation noise, field of view occlusion and exposure effects of the camera, visual SLAM is less effective in constructing precise outdoor maps. Furthermore, the performance of visual SLAM is susceptible to be affected in low-texture, high-speed motion, and fluctuating lighting conditions.

Compared to visual sensors, LiDAR sensors provide accurate and lighting-invariant perception of the environment, which make LiDAR SLAM achieve higher accuracy and robustness in state estimation and map construction. However, LiDAR SLAM easily degenerates in those scenes with less geometric features, such as tunnels, pipelines and corridors.

Above all, the SLAM method based on a single sensor (LiDAR or camera) is difficult to achieve high robustness of localization and mapping in degenerate scenarios. Therefore, fusing the observations of the LiDAR and camera can fully utilize their respective advantages and compensate for their shortcomings, which can solve the degeneracy problem. Inspired by this, we proposed LVIO-Fusion and its contributions are summarized as follows:

- We propose a LiDAR-visual-inertial odometry and mapping framework that consists of a direct LiDAR-inertial odometry (LIO) and a direct visual-inertial odometry (VIO). The proposed system can capture subtle geometric and radiometric changes to overcome sensor-degenerated issues in geometry-less and texture-less environments.
- To lower the computation load caused by a large number of LiDAR points, the LIO subsystem employs a dynamic voxel mapping technique for point registration and mapping efficiently.
- To achieve precise and robust state estimation in geometrically sparse and texture-less environments, we directly

Manuscript received 10 October 2023; accepted 8 February 2024. Date of publication 28 February 2024; date of current version 11 March 2024. This letter was recommended for publication by Associate Editor M. Hanheide and Editor S. Behnke upon evaluation of the reviewers' comments. (*Corresponding author: Sheng Bao.*)

Hongkai Zhang, Sheng Bao, and Jianjun Yuan are with Shanghai Robotics Institute, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China, and also with the Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China (e-mail: baos@shu.edu.cn).

Liang Du is with Shanghai Robotics Institute, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China.

Shugen Ma is with Robotics and Autonomous Systems Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong 511453, China.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3371383>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3371383

project map points to images to build optical flow matching, which integrates LiDAR and camera measurements at a deeper level. Besides, the VIO subsystem leverages online photometric calibration to get true radiometric information, further enhancing robustness in scenes with changing illumination.

II. RELATED WORK

In this section, we review previous works closely related to our system, including LiDAR SLAM, visual SLAM and LiDAR-visual SLAM.

A. LiDAR SLAM

LiDAR SLAM is performed by using scan-matching methods such as Iterative Closest Point (ICP) [7] to find the translation between two consecutive frames. To promote the performance of scan-matching, the LiDAR sensor is commonly incorporated with other sensors (IMU, encoder) to provide localization and mapping. In LOAM [4], the IMU is used to deskew the scan and provide a prior translation for scan-matching. However, the LOAM system is classified as a loosely-coupled framework since the IMU measurement and LiDAR odometry are not fused in a joint optimization. Compared to loosely-coupled system, the tightly-coupled system typically has higher accuracy and gets a growing attention. Based on graph optimization, LIOM [8] jointly optimizes measurements from IMU and LiDAR in a sliding window to obtain state estimation. LIO-SAM [5] presents a tightly-coupled smooth and mapping framework, which fuses IMU pre-integration, LiDAR odometry, loop closure factor and GPS odometry in a graph factor, to achieve higher accurate trajectory. Compared to the tightly coupled SLAM framework based on the factor graph optimization method, the filter-based framework is typically more efficient. LINS [9] presents a tightly coupled framework based on an iterated Kalman filter method to obtain 6 DoF ego-motion. To reduce the computation complexity caused by the dimension of the measurements, FAST-LIO [10] proposes a new formula that uses dimension of the state to compute the Kalman gain. However, the above LiDAR SLAM methods are based on feature points which neglect subtle geometric change. FAST-LIO2 [6] proposes an incremental k-dimensional Tree (ikd-Tree) that performs efficient nearest neighbor search to further reduce the computation load, and directly uses raw LiDAR points to perform scan-matching.

B. Visual SLAM

The previous works of visual SLAM can be categorized into feature-based methods and direct methods. The feature-based visual SLAM detects features, builds feature correspondences across frames, and minimizes reprojection errors to estimate relative camera motion. MonoSLAM [11] is the first system that using an Extended Kalman Filter (EKF) to solve monocular SLAM problem. By using Shi Tomasi feature points, the system establishes data association to estimate camera state and create sparse map. However, the system only optimizes the current state and landmark points while marginalizes previous camera states, which results in accumulation of linearization errors. In contrast, keyframe-based systems estimate the camera state by using selected frames which performs more costly but accurate Bundle Adjustment (BA) optimization. Following this idea, PTAM [12] is the most representative system that splits

state tracking and mapping in parallel threads and performs in real-time successfully in small environments. Building on these ideas, ORB-SLAM proposed in [2] is more complete, and achieves short-term, mid-term and long-term data association by extracting ORB features. Based on ORB-SLAM, Mur-Artal presents ORB-SLAM2 [13] that utilizes scale information provided by stereo or RGB-D cameras. In RGB-D or stereo mode, the system can direct obtain depth information and achieves more robust in state tracking. Its further work ORB-SLAM3 [3], a tightly-coupled visual-inertial SLAM system jointly optimizes camera and IMU measurements in a sliding window that performs higher accuracy in localization. Unlike the ORB series methods, VINS-Mono [1] establishes data associations for corner points by using the optical flow tracking method instead of visual descriptors. Then, VINS-Mono employs BA optimization method to minimize the reprojection errors.

The direct visual SLAM skips feature extraction process, and minimizes photometric errors between successive frames to estimate camera state. The early direct VO method like DTAM [14], estimates camera states by aligning the whole image with the dense depth map. However, creating a dense depth map from monocular images requires much computation, and is typically based on GPU to run in real-time. LSD-SLAM proposed in [15] presents a large-scale direct monocular SLAM method which tracks camera pose by minimizing photometric errors directly, and performs graph optimization to correct accumulate errors. It should be noted that LSD-SLAM requires strong assumption of photometric invariance for stale operation. The direct and sparse formulation for monocular visual odometry proposed in [16] further takes full advantage of photometric camera calibration, which includes known exposure times, gamma correction and lens attenuation. Compared to previous direct visual SLAM systems, the system improves the accuracy and robustness based on the photometric camera calibration.

Previous LiDAR and visual SLAM works have presented advantages and characteristics of both feature-based and direct methods. The direct LiDAR SLAM method like FAST-LIO2 can capture subtle geometric change in geometry-less environments, and the direct visual SLAM typically presents higher accuracy and robustness than feature-based visual SLAM in low-texture environments [17], [18], [19]. Therefore, to overcome difficulties on state estimation in low-texture scenes, LVIO-Fusion directly uses the LiDAR and camera measurements for state estimation.

C. LiDAR-Visual SLAM

Compared to LIO system and VIO system, LVIO system [20], [21], [22] typically has higher accuracy and robustness by coupling LiDAR, camera and IMU measurements in sensor-degenerated tasks. A LiDAR-visual-inertial odometry and mapping proposed in [23] employs a loosely coupled framework to estimate state from coarse to fine. Combining the IMU prediction, the system performs VO to obtain a coarse ego-motion estimation. The LO system further refines state estimation and registers the current scan to the global map. Compared to the loosely coupled system, the tightly-coupled LVIO framework typically performs better which has been demonstrated in recent works. LIC-Fusion [24] presents a tightly-coupled system that fuses LiDAR, inertial, and camera measurements based on MSCKF framework [25]. The system tracks sparse LiDAR feature points and visual features to provide the accurate motion

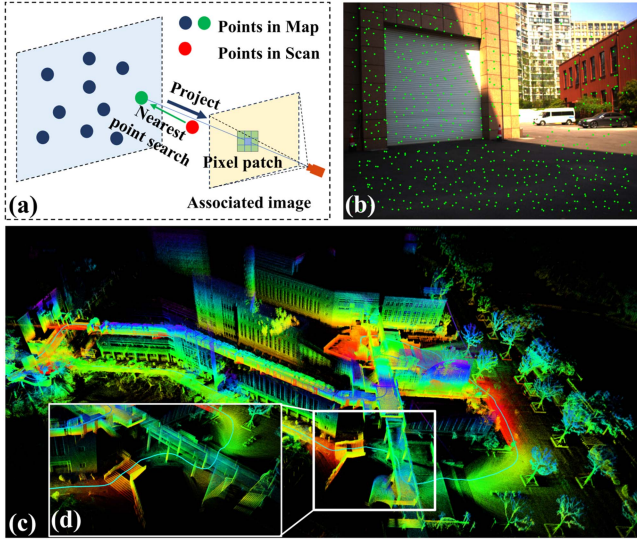


Fig. 1. (a) Process of projecting map points onto the image. The LIO subsystem uses point-to-plane matching method to register the current scan to the global map, and projects map points to the current image. The VIO subsystem utilizes the projected points and Lucas Kanade (LK) optical flow algorithm to establish associations between two consecutive images, aligning them to refine the state estimation. (b) Green points are the projected points. (c) Based on this key innovation, Our LVIO-Fusion is able to reconstruct a dense, clear 3D map of the challenging scenarios that contain illumination change, altitude change, and scale change. The blue path is the travelling trajectory, and the details of the 3D map is shown in (d).

tracking in different environments. Its further work LIC-Fusion 2.0 [26] introduces a novel planar feature tracking method and a new outlier rejection method for high quality data association. LVI-SAM proposed in [27] shows a LVIO framework that combines a visual-inertial system (VIS) and LiDAR-inertial system (LIS). The LIS provides depth information for VIS to facilitate initialization. The LIS uses the state of the VIS as initial guesses for scan-matching. Similar to LVI-SAM, mVIL-Fusion [21] proposed a novel double-sliding-window-based optimization that contains scan-to-scan translation and scan-to-map rotation to enhance the accuracy of the state estimation. R2LIVE proposed in [28] fuses the multi-measurements in an error state iterated Kalman filter method. The LIO and VIO subsystems of R2LIVE can run independently when one of them is failed. However, the above LVIO frameworks use features to estimate state while lose many subtle features, which are not suitable for degenerate environments, such as geometry-less or texture-less scenarios.

III. SYSTEM OVERVIEW

Unlike the above feature-based LVIO systems, the proposed LVIO-Fusion is built by a LIO subsystem and a VIO subsystem which are both based on direct method. An overview of our system is shown in Fig. 2. The LIO subsystem utilizes IMU pre-integration to deskew point clouds, and employs a dynamic voxel mapping method to reconstruct the dense map. The LVIO-Fusion utilizes the state output by LIO subsystem as the initial values for VIO subsystem, and leverage online photometric calibration to obtain true radiance information. Then, the local map is projected to the current image to obtain tracked points directly, as shown in Fig. 1(a). The VIO subsystem utilizes the LK method to track projected points for data association, and

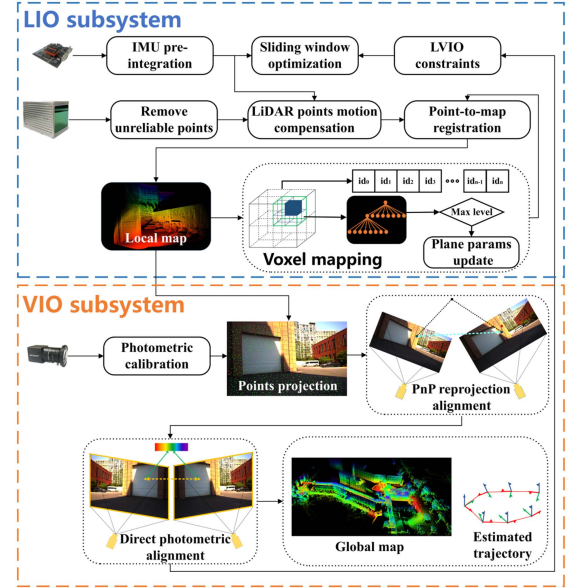


Fig. 2. System overview of LVIO-Fusion.

obtains the rough state estimation by minimizing the frame-to-frame reprojection errors. Furthermore, the two consecutive image frames are aligned by minimizing frame-to-frame photometric errors to obtain a fine state estimation, and update the global point cloud map. Then, the LVIO constraints are added to sliding window factor-graph optimization to correct IMU bias. To reduce computation load, the resolution of projected map points is set to 0.05m. Additionally, to improve the quality of tracking points, map points are not projected within the edge region of the image, as shown in Fig. 1(b).

IV. PRELIMINARIES

A. Notations

In this section, we define frames and notations of this letter. We denote $^W(\cdot)$ as the world frame, $^L(\cdot)$ as the LiDAR frame, $^C(\cdot)$ as the camera frame, $\mathbf{x}_k$ as the state $\mathbf{x}$ at the end time of k -th LiDAR scan, $\mathbf{b}_g$ and $\mathbf{b}_a$ as the bias of gyroscope and accelerometer.

B. State Vector

In our system, we define the state $\mathbf{x}$ as:

$$\mathbf{x} = [{}^W\mathbf{R}^T \quad {}^W\mathbf{p}^T \quad {}^W\mathbf{v}^T \quad \mathbf{b}_a^T \quad \mathbf{b}_g^T] \quad (1)$$

where ${}^W\mathbf{R}$, ${}^W\mathbf{p}$, ${}^W\mathbf{v}$ denote the rotation, position and velocity of the state in the world frame, $\mathbf{b}_a$, $\mathbf{b}_g$ are the accelerometer bias and gyroscope bias. We denote the IMU frame as the body frame and the first IMU frame as the world frame. Here, we calibrated the extrinsic and intrinsic parameters of the LiDAR, camera, and IMU, and performed time synchronization in an offline manner [29].

C. Camera Photometric Calibration

The camera records the intensity of reflected light through the CMOS sensor and presents the radiance of the world in the form

of image pixels. The total amount of power received at CMOS sensor location $\mathbf{q}$ is called the irradiance $\mathbf{C}_i(\mathbf{q})$. Direct method estimates the camera pose by using the translation of image pixels intensity. The direct method relies on the assumption of photometric invariance, that is, the intensity of the pixels representing the same spatial point in two images does not change. Typically, the intensity reflected in the pixel values of images recorded at different spatial positions within the same scene point is consistent. However, for most cameras, there will be a vignetting effect when the pixel intensity is towards the image edges. This so-called vignetting effect may be due to partial blocking of light rays by the lens barrel or induced by the geometry of the lens. Moreover, the received irradiance is accumulated over exposure time. To overcome the vignetting effect, the irradiance value $\mathbf{C}_i(\mathbf{q})$ at location $\mathbf{q}$ in frame k can be obtained by multiplying the radiance c_i of the scene point with a vignetting factor $V(\mathbf{q})$: $V(\mathbf{q}) \in [0, 1]$. The vignetting factor depends on the location of the lens region.

$$\mathbf{C}_i(\mathbf{q}) = V(\mathbf{q})c_i. \quad (2)$$

The pixel intensity $\mathbf{I}_i(\mathbf{q})$ is then computed by the camera response function $\mathbf{O}_i$:

$$\mathbf{I}_i(\mathbf{q}) = \mathbf{O}_i(\sigma_i V(\mathbf{q})c_i), \quad \mathbf{I}_i \in [0, 255], \quad (3)$$

where the σ is the exposure time. If the exposure time is unknown, it is estimated based on [18]. Otherwise, it is obtained from the camera's API. With the photometric calibration model, the observed pixel radiance can be computed as:

$$c_i = \frac{\mathbf{I}_i^{-1}(\mathbf{q})}{\sigma_i V(\mathbf{q})} \quad (4)$$

After photometric calibration, the pixel radiance value c_i is used for frame-to-frame optical flow tracking and minimizing the photometric residuals, which are elaborated detailed in V. C and V. D.

V. STATE ESTIMATION

A. IMU Integration

To overcome the problem of trajectory drift during intense motion, The system utilizes IMU-integration referring from [20] between two consecutive frames (LiDAR frames or visual frames), to obtain high-frequency state output. Here, we assume that the IMU bias remains constant during the time interval of two consecutive frames. Denote j as the index of the IMU measurement, the state update can be computed as:

$$\mathbf{x}_j = \mathbf{x}_{j-1} \boxplus \Delta t \mathbf{f}_j \quad (5)$$

where $\boxplus$ is the operator on the manifold [20]. The IMU-integration function $\mathbf{f}_j$ is denoted as follows:

$$\mathbf{f}_j = \begin{bmatrix} \omega_r - \mathbf{b}_g - \mathbf{n}_g \\ \mathbf{w} \mathbf{v} + \frac{1}{2} (\mathbf{w} \mathbf{R}_j (\mathbf{a}_r - \mathbf{b}_a - \mathbf{n}_a) + \mathbf{w} \mathbf{g}) \Delta t \\ \mathbf{w} \mathbf{R}_j (\mathbf{a}_r - \mathbf{b}_a - \mathbf{n}_a) + \mathbf{w} \mathbf{g} \\ \mathbf{b}_g^* \\ \mathbf{b}_a^* \end{bmatrix} \quad (6)$$

where ω_r and $\mathbf{a}_r$ denote the raw IMU angular velocity and acceleration velocity, $\mathbf{b}_g$ and $\mathbf{b}_a$ are the IMU bias, $\mathbf{n}_g$ and $\mathbf{n}_a$ are the measurement noise of ω_r and $\mathbf{a}_r$, and the Δt denotes the

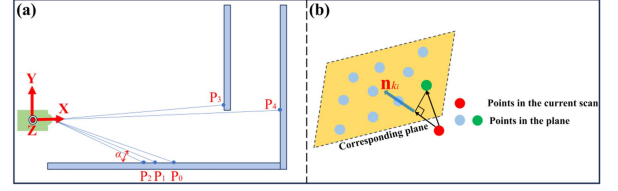


Fig. 3. (a) Illustration of the incident angle, hidden point, denoted as α , and $\mathbf{P}_4$. (b) Illumination of the point-to-plane association.

IMU sample period. The above data are the measurements of the IMU index j . $\mathbf{b}_g$ and $\mathbf{b}_a$ are jointly optimized alongside the LiDAR odometry constraints and visual odometry constraints in the factor graph, to obtain the update bias $\mathbf{b}_g^*$ and $\mathbf{b}_a^*$.

B. LiDAR Point-to-Plane Residual

Considering the measuring mechanism of LiDAR (reflection angle, reflection energy), we employ a selection strategy to extract reliable LiDAR points. The incident angle is the angle between LiDAR ray and the local plane which is computed as

$$\alpha(\mathbf{P}_1) = \cos^{-1} \left(\frac{(\mathbf{P}_0 - \mathbf{P}_2) \cdot \mathbf{P}_1}{|\mathbf{P}_0 - \mathbf{P}_2| |\mathbf{P}_1|} \right) \quad (7)$$

To improve the accuracy of scan matching and provide reliable depth values for VIO subsystem, the following LiDAR points are removed: 1). The intensity values of LiDAR points are too large or too small. Since the excessive intensity typically causes signal distortion, while insufficient intensity causes decrease signal-to-noise ratio, both of which can reduce the ranging accuracy of LiDAR points. 2). The incident angle value is near to π or 0. Because the laser spot of this LiDAR point is larger, the measured range of the LiDAR point is the average distance of all ranges within the spot, rather than the distance of a specific point. This phenomenon can reduce the ranging accuracy of LiDAR points. 3). The LiDAR points are hidden behind an object. Because the occluded LiDAR points can cause incorrect depth projection, which affects the accuracy of visual odometry. The LiDAR point $\mathbf{P}_4$ in Fig. 3(a) is a hidden point if $|D(\mathbf{P}_4) - D(\mathbf{P}_3)| \geq 0.1D(\mathbf{P}_4)$, and $D(\mathbf{P}_4) \geq D(\mathbf{P}_3)$, where the D denotes the distance of the LiDAR point.

To eliminate point cloud distortion caused by motion, we utilize a linear interpolation method based on IMU-integration. Denote $(\mathbf{R}_k, \mathbf{T}_k)$ as the state at the end time of the current LiDAR frame k , $(\mathbf{R}_{k-1}, \mathbf{T}_{k-1})$ as the state at the end time of the previous frame, and $(\mathbf{R}_k^{k-1}, \mathbf{T}_k^{k-1})$ as the relative rotation and translation between the previous and the current frame which is computed by IMU-integration, $\Delta \mathbf{x} = \sum_{s=j}^{j+n} \Delta t \mathbf{f}_s$. The Δt is the IMU sample period, $s = j$ represents the IMU index at the end time of the previous LiDAR frame, and $s = j + n$ represents the IMU index at the end time of the current LiDAR frame.

$$\mathbf{R}_k = \mathbf{R}_{k-1} \mathbf{R}_k^{k-1}, \quad \mathbf{T}_k = \mathbf{R}_{k-1} \mathbf{T}_k^{k-1} + \mathbf{T}_{k-1} \quad (8)$$

The linearly interpolated pose at point i is then computed as:

$$\mathbf{R}_i^{k-1} = \mathbf{I} + \hat{\omega}_i \sin(s\theta) + \hat{\omega}_i^2 (1 - \cos(s\theta)), \quad \mathbf{T}_i^{k-1} = s \mathbf{T}_k^{k-1} \quad (9)$$

where $\hat{\omega}_i$ is the skew symmetric matrix of ω_i , which is the unit vector of the rotation axis of $\mathbf{R}_i^{k-1}$, θ is the angular change between the previous and the current frame, s is computed as

$s = (t - t_{k-1}) / (t_k - t_{k-1})$, and t is the timestamp of $\mathbf{P}_{k_i}$, $t \in [t_{k-1}, t_k]$. Therefore, we can obtain the rotation and translation of $\mathbf{P}_{k_i}$ at the time t :

$$\mathbf{R}_{k_i} = \mathbf{R}_{k-1} \mathbf{R}_i^{k-1}, \quad \mathbf{T}_{k_i} = \mathbf{R}_{k-1} \mathbf{T}_i^{k-1} + \mathbf{T}_{k-1} \quad (10)$$

With $\mathbf{R}_{k_i}$ and $\mathbf{T}_{k_i}$ estimated by IMU integration, the LiDAR point is removed skewness, and can be projected to the global map before minimizing the point-to-plane residual.

$${}^W \mathbf{P}_{k_i} = \mathbf{R}_{k_i} \mathbf{P}_{k_i} + \mathbf{T}_{k_i} \quad (11)$$

To capture subtle geometric features during motion, the feature extraction module is omitted and all LiDAR points are involved in scan matching. The LiDAR subsystem utilizing point-to-plane association registers points in a new frame to the local map (last 50 LiDAR frames) to performs state estimation. For a point ${}^W \mathbf{P}_{k_i}$ of the current frame, 10 nearest points that composed as a planar are associated in the local map, as shown in Fig. 3(b). Referring to [6], if the minimum eigenvalue of the covariance matrix formed by 10 nearest points is less than 0.01, then the set of these 10 points constitutes a plane. To boost the searching speed, we apply a dynamic voxel map for point-to-map data association, which is elaborated detailed in Section VI-A. The point-to-plane residual is then computed as

$$\mathbf{e}_{k_i} = \mathbf{n}_{k_i}^T ({}^W \mathbf{P}_{k_i} - \mathbf{q}_{k_i}) \quad (12)$$

where $\mathbf{n}_{k_i}^T$ is the normal vector of the corresponding plane, and $\mathbf{q}_{k_i}$ is a point lying on the plane. Therefore, the cost function of point-to-plane errors can be computed as

$$\mathbf{f}_l = \operatorname{argmin} \frac{1}{2} \sum_{i=0}^n \|\mathbf{e}_{k_i}^2\| \quad (13)$$

Then the Levenberg-Marquardt (L-M) method is applied to minimize the point-to-plane residuals, and the computed state transformation $\Delta \mathbf{x}_L$ can be added to $\mathbf{x}_{k-1}$ as below:

$$\mathbf{x}_k = \mathbf{x}_{k-1} \boxplus \Delta \mathbf{x}_L \quad (14)$$

C. Frame-to-Frame Visual Reprojection Residual

Assuming that LiDAR map points $\mathbf{M}_k = \{\mathbf{P}_{k_0}, \dots, \mathbf{P}_{k_n}\}$ in the current local map are projected to the previous image frame, denoted as $\gamma_{k-1} = \{\gamma_{k-1_0}, \dots, \gamma_{k-1_n}\}$. We utilize the Lucas-Kanade optical flow to extract corresponding pixels of γ_{k-1} in the current image frame, denoted as $\gamma_k = \{\gamma_{k_0}, \dots, \gamma_{k_n}\}$. For each map point, it needs to be projected onto the current camera frame. We take the i -th point $\mathbf{P}_{k_i}$ as example:

$${}^C \mathbf{P}_{k_i} = ({}^W \mathbf{R}_I \cdot {}^I \mathbf{R}_C)^T \cdot {}^W \mathbf{P}_{k_i} - ({}^W \mathbf{R}_I \cdot {}^I \mathbf{R}_C)^T \cdot {}^W \mathbf{P}_I - {}^I \mathbf{R}_C^T \cdot {}^I \mathbf{P}_C, \quad (15)$$

where ${}^C \mathbf{P}_{k_i}$ is the projected map point on the current camera frame, ${}^I \mathbf{R}_C$ and ${}^I \mathbf{P}_C$ are the extrinsic params between IMU and camera.

During the iteration, the pixel position of the projected map point ${}^C \mathbf{P}_{k_i}$ needs to continually approach the pixel point γ_{k_i} . Then, the reprojection error can be calculated as:

$$\mathbf{e}(\gamma_{k_i}, \mathbf{x}_k, {}^C \mathbf{P}_{k_i}) = \gamma_{k_i} - \zeta({}^C \mathbf{P}_{k_i}, \mathbf{x}_k), \quad (16)$$

where $\mathbf{x}_k$ is the state estimated by the IMU integration or the LIO system (when the timestamps of the camera is aligned with

the timestamps of the LiDAR) at the current time, and ζ is the projection function.

$$\zeta_m({}^C \mathbf{P}_{k_i}, \mathbf{x}_k) = \left[f_{m,u} \frac{\mathbf{P}_{x_{k,i}}}{\mathbf{P}_{z_{k,i}}} + c_{m,u}, f_{m,v} \frac{\mathbf{P}_{y_{k,i}}}{\mathbf{P}_{z_{k,i}}} + c_{m,v} \right], \quad (17)$$

where $(f_{m,u}, f_{m,v})$ and $(c_{m,u}, c_{m,v})$ are the local length and principle point associated with the camera. The cost function of reprojection error is then computed as

$$\mathbf{f}_r = \operatorname{argmin} \frac{1}{2} \sum_{i=0}^n \left\| \mathbf{e}(\gamma_{k_i}, \mathbf{x}_k, {}^C \mathbf{P}_{k_i}) \right\|^2 \quad (18)$$

After minimizing the frame-to-frame reprojection errors, the state transformation $\Delta \mathbf{x}_r$ can be added to the $\mathbf{x}_k$ as below:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k \boxplus \Delta \mathbf{x}_r \quad (19)$$

Here, we remove the projected points if their reprojection errors are too large.

D. Frame-to-Frame Visual Radiance Residual

The rough state estimation has been obtained by minimizing frame-to-frame reprojection errors. With the rough state $\hat{\mathbf{x}}_k$, the current LiDAR points $\mathbf{M}'_k$ of the local map can be projected to the current image within the image FoV to locate their pixels. Here, each image has been photometrically corrected by computing the observed pixel radiance, noted as Section IV-C. Take the projected point $\mathbf{M}'_k$ of the current image as an example, its frame-to-frame radiance errors is computed as:

$$\mathcal{F}(\rho_{k_i}, \hat{\mathbf{x}}_k, \mathbf{M}'_k) = \rho_{k_i} - \varphi_{k_i} \quad (20)$$

where ρ_{k_i} is the average radiance of the current image's pixel patch, φ_{k_i} is the average radiance of the previous image's pixel patch. To fully utilize image information, we use the photometric information of image patches instead of individual pixels. Here, we set 9 pixel points as a pixel patch, as shown in Fig. 1(a). The center position of the pixel patch is located by the projection point of $\mathbf{M}'_k$.

By minimizing the radiance errors, the state transformation $\Delta \mathbf{x}_p$ can be added to $\hat{\mathbf{x}}_k$ as follow:

$$\mathbf{f}_p = \operatorname{argmin} \frac{1}{2} \sum_{i=0}^n \left\| \mathcal{F}(\rho_{k_i}, \hat{\mathbf{x}}_k, \mathbf{M}'_k) \right\|^2 \quad (21)$$

$$\mathbf{x}_k = \hat{\mathbf{x}}_k \boxplus \Delta \mathbf{x}_p \quad (22)$$

The $\mathbf{f}_p$ is the cost function that minimizes photometric errors. This optimization problem can be solved by the L-M optimization method. The computed state is used to register the current scan into the global map.

VI. MAP UPDATE AND OPTIMIZATION

A. Map Update

Inspired by FASTER-LIO [30] that using incremental voxels as point cloud spatial data structure, we apply a dynamic voxel structure for point-to-map inquiring, inserting and updating. The dynamic voxel map is organized by a Hash table and an octree. Specifically, the space of local map is first cut into coarse voxels in the world frame. Then, voxels that contain map points are indexed into a Hash table. If the map points within a voxel form a plane, the plane parameters of these map points



Fig. 4. Datasets are collected on two platforms. (a) A handheld device. (b) A mobile vehicle.

are stored which are used for point-to-plane scan matching. Otherwise, the current voxel is still further divided into eight sub-voxels based on the octree, and the plane parameters are computed in each sub-voxel until the maximum level is reached. Additionally, the voxels that not contain map points are labeled as inactive voxels, otherwise are labeled as active voxels. The inactive voxels do not participate in the point-to-plane matching process.

When a new LiDAR scan is registered into the global map, the plane parameters of voxels need to be updated dynamically. However, as new points are continuously received, the processing time for updating plane parameters is also growing. To solve this issue, we have recorded the changes in the plane normal vectors of the current voxel as new points are added. Fortunately, the plane normal vector tends to be stable when the number of new points reaches 20, and the current plane normal vector is stored. Considering the efficiency, the latest 10 points are used to compute new plane normal vector when new points are received. If the latest plane vector has large difference with the previous stored normal vector, the voxel map need to be reconstructed following the above steps.

B. Factor Graph Optimization

As mentioned in Section V-A, the accuracy of IMU integration is affected by IMU bias which are corrected by other constraints. To further improve the robustness and accuracy of the IMU-integration, we leverage a factor graph optimization for optimizing the IMU bias within a sliding window of image keyframes. Our factor graph optimization framework is similar to LVI-SAM, but incorporates LiDAR-visual odometry constraints which is fused in the measurement level. We use the iSAM2 [31] algorithm for building sliding window factor graph to optimize IMU bias.

VII. EXPERIMENT AND RESULT

In this section, we validate the advantages of our proposed system against other state-of-art systems in extensive challenge scenarios. Our handheld data collection device is shown in Fig. 4(a), includes a HIKROBOT MV-CA016-10UC camera, a LIVOX AVIA LiDAR, a Xsens MIT-600 Inertial Measurement Unit (IMU), and a computation platform (equipped with Intel I5-10210U CPU, 64G RAM). The mobile vehicle is additionally equipped with an real time kinematic (RTK) device. To evaluate the precision of LVIO-Fusion, we use the real-time kinematic (RTK) to obtain the ground-truth trajectory. For the GPS denied environment, we employ relocalization techniques to assess the precision of LVIO-Fusion.

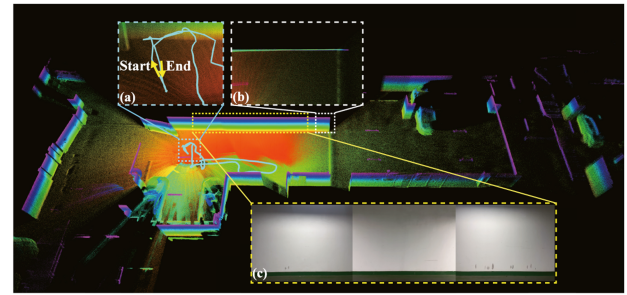


Fig. 5. Experiments in LiDAR-degenerated and low-texture environments. (a) Without loop closure detection, LVIO-Fusion can loop by itself. (b) The point cloud is clear and free from any overlapping. (c) The white wall that is both LiDAR-degenerated and low texture.

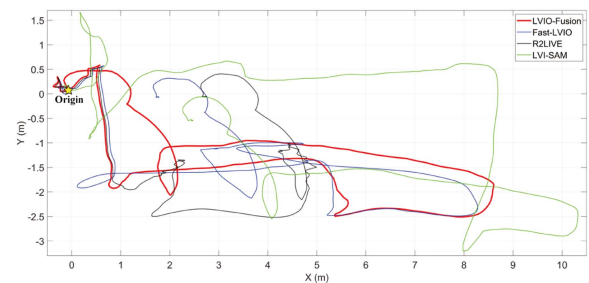


Fig. 6. Comparison of the estimated trajectories in Experiment-1.

A. Experiment-1: Evaluation of Accuracy and Robustness in Degenerate Environments

In this experiment, we test the robustness and precision of LVIO-Fusion in both geometry-less and texture-less scenarios. We choose a white wall, which is a single plane with very limited geometric and texture features, as the validation scene for our algorithm, as shown in Fig. 5(c). We displace the handle platform a distance while orienting it towards the white wall, subsequently returning it to its initial position. We present a performance comparison between our system and the current state-of-the-art LiDAR-Visual-Inertial (LVI) SLAM methods. The estimated trajectories of our system, FAST-LVIO [20], R2LIVE [28], LVI-SAM [27] are shown in Fig. 6. Since the white wall only contains a single plane and has no obvious texture features, the R2LIVE and LVI-SAM systems that are based on geometric and visual features both exhibit noticeable trajectory drift. The FAST-LVIO system directly employs LiDAR points and radiance information for state estimation. However, FAST-LVIO only capitalizes on the photometric information of the sparse point cloud map (low density) in alignment with the image, discarding a substantial amount of image information. Moreover, FAST-LVIO lacks a photometric calibration process that results in divergence in image photometric aligning and trajectory drift.

Our proposed system first minimizes the frame-to-frame reprojection errors to obtain a coarse state estimation, then refines this estimation by minimizing the photometric errors of image blocks from frame to frame. In such degenerate environments, employing the coarse-to-fine state estimation can suppress the divergence of photometric alignment. Moreover, utilizing image blocks, rather than individual points, to minimize photometric errors enables the plentiful exploitation of image information,

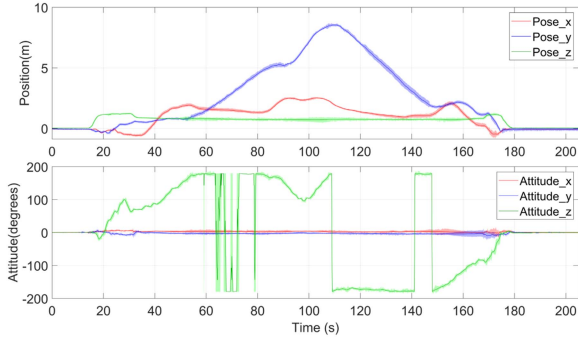


Fig. 7. Estimated results (position and attitude) of all tests in Experiment-1. We test our system ten times using the collected data from Experiment-1. We compute the mean, the maximum, and minimum values of the estimated position and attitude from ten tests. Solid line: mean value. Shaded part: the range between the maximum and minimum values.

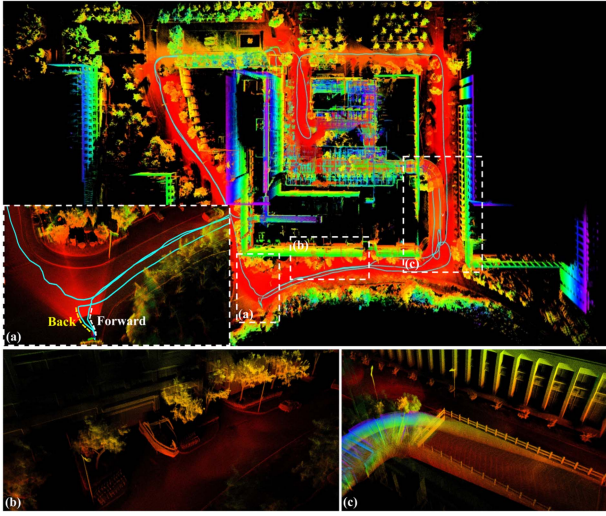


Fig. 8. Reconstructed 3D maps in Experiment-2. (a) shows LVIO can return back to the origin without loop closure detection. Some details of the map are shown in (b) and (c).

thus returns to the origin point successfully, as shown in Fig. 5(a). The estimated results of all tests in Experiment-1 is shown in Fig. 7. The shaded parts are nearly aligned with the mean value, which reveals that our system operates with stability under such degenerate conditions. Our proposed system performs well in the degenerate environment and returns to the origin point with drift 0.0514m in translation and 2.75 degrees in rotation.

B. Experiment-2: Evaluation of Accuracy in Outdoor and Indoor Environments

In this experiment, we challenge our proposed system with an extremely complex environment that includes illumination change from outdoor to indoor and indoor to outdoor, altitude change from ground to roof, and scale change from open to narrow. We collect data around the buildings of Shanghai University for four times. The data collection paths are slightly different, but they all started and ended at the same location. The bird-view of the reconstructed 3D map is shown in Fig. 8. The point cloud 3D map is distinctly perceptible and devoid of any repetition

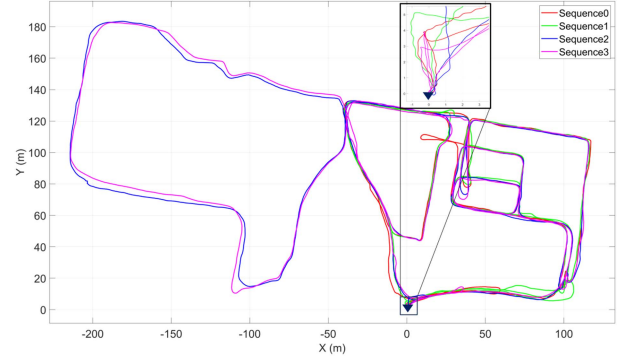


Fig. 9. Four traveling end-to-end trajectories in Experiment-2. The length of four traveling trajectories are 1465 m, 1531 m, 1892 m, and 1963 m respectively.

TABLE I
COMPARISON OF END-TO-END TRANSLATION ERRORS IN EXPERIMENT-2

Algorithm	Type	End-to-End Translation Errors [m]			
		Seq.0 1465m	Seq.1 1531m	Seq.2 1892m	Seq.3 1963m
LIO-SAM[5]	LIO	3.473	4.398	5.560	4.357
FAST-LIO2[6]	LIO	0.155	0.129	0.996	1.229
VINS-Mono[1]	VIO	5.821	18.798	21.003	14.421
LVI-SAM[27]	LVIO	1.080	2.317	3.457	3.526
R2LIVE[28]	LVIO	0.159	0.131	1.005	1.159
FAST-LIVO[20]	LVIO	0.123	0.141	0.865	0.904
Ours	LVIO	0.082	0.098	0.420	0.217

TABLE II
RELATIVE TRANSLATION ERRORS IN EXPERIMENT-3

Relative translation errors (m) w.r.t RTK in Experiment-3					
Algorithm	200 s	400 s	600 s	800 s	1000 s
Ours	0.295	0.274	0.264	0.367	0.385
LIO-SAM[5]	1.049	2.913	2.253	3.1188	5.197
FAST-LIO2[6]	0.221	0.352	1.039	1.565	1.393
VINS-Mono[1]	0.453	3.766	19.385	24.871	15.069
LVI-SAM[27]	1.334	1.515	2.037	2.655	3.129
R2LIVE[28]	0.351	0.873	0.984	1.211	1.189
FAST-LIVO[20]	0.201	0.488	0.859	1.147	0.826

or ghosting. The four traveling trajectories are shown in Fig. 9. In the absence of loop closure detection technology, these four trajectories can form closed loops with minimal drift which demonstrates that the proposed algorithm has high robustness and precision in such complicated environments. Furthermore, the end-to-end translation errors of four trajectories are shown in Table I. LVIO-Fusion demonstrates superior performance compared to existing LIO, VIO, and LVIO methods.

C. Experiment-3: Evaluation of Accuracy by Using RTK

In this experiment, we collect a sequence outdoors which includes a RTK measurements as the ground-truth. We compare the traveling trajectories estimated by LVIO-Fusion, LIO-SAM, FAST-LIO2, VINS-Mono, LVI-SAM, R2LIVE, FAST-LVIO to the ground-truth trajectory, as shown in Fig. 10. The result shows that the trajectory generated by LVIO-Fusion is nearest to the ground-truth trajectory. Moreover, we provide the relative translation errors with respect to RTK measurements at the sampling time, as shown in Table II. The results indicate that

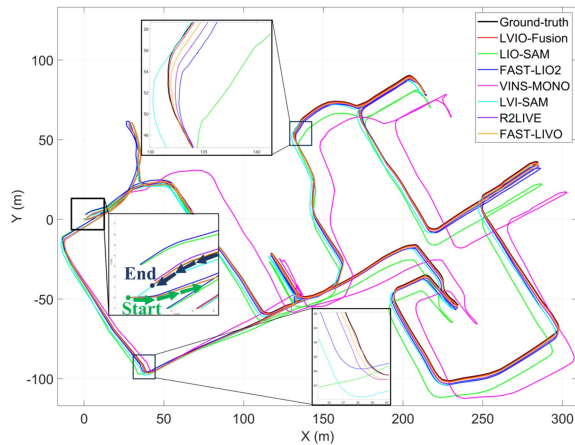


Fig. 10. Comparison of estimated trajectories with RTK ground-truth in Experiment-3.

LVIO-Fusion maintained high accuracy throughout the entire traveling time, with the least relative translation errors.

VIII. CONCLUSION

This letter presented a tightly-coupled LiDAR-inertial-visual framework, which couples the LiDAR and camera at the measurement level. The proposed system directly projects map points to image to establish data association, and employs a coarse-to-fine state estimation by minimizing the frame-to-frame reprojection errors and photometric errors. In all tests, our system achieves high localization accuracy and robustness in extremely challenging scenarios, even with sensor degenerations. We also provide various comparisons with other state-of-the-art LIO, VIO, LVIO SLAM works, and the results show that our system achieve better accuracy and robustness. In our future work, we will further explore the mapping relationship between point clouds and images to fully utilize the high-resolution image information.

REFERENCES

- [1] 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, Aug. 2018.
- [2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
- [4] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," *Robot. Sci. Syst.*, vol. 2, no. 9, pp. 1–9, 2014.
- [5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.
- [6] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [7] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3450–3466, Jul. 2022.
- [8] S. Zhao, Z. Fang, H. Li, and S. Scherer, "A robust laser-inertial odometry and mapping method for large-scale highway environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1285–1292.
- [9] 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. Automat.*, 2020, pp. 8899–8906.
- [10] W. Xu and F. Zhang, "FAST-LIO: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.
- [11] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
- [12] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. IEEE/ACM 6th Int. Symp. Mixed Augmented Reality*, 2007, pp. 225–234.
- [13] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
- [14] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2320–2327.
- [15] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 834–849.
- [16] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
- [17] S. Zhong and P. Chirarattananon, "Direct visual-inertial ego-motion estimation via iterated extended Kalman filter," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1476–1483, Apr. 2020.
- [18] D. Luo, Y. Zhuang, and S. Wang, "Hybrid sparse monocular visual odometry with online photometric calibration," *Int. J. Robot. Res.*, vol. 41, no. 11–12, pp. 993–1021, 2022.
- [19] J. Mo, M. J. Islam, and J. Sattar, "Fast direct stereo visual SLAM," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 778–785, Apr. 2021.
- [20] 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.*, 2022, pp. 4003–4009.
- [21] Y. Wang and H. Ma, "mVIL-fusion: Monocular visual-inertial-LiDAR simultaneous localization and mapping in challenging environments," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 504–511, Feb. 2023.
- [22] X. Zhong, Y. Li, S. Zhu, W. Chen, X. Li, and J. Gu, "LVIO-SAM: A multi-sensor fusion odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2021, pp. 440–445.
- [23] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, 2018.
- [24] 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.*, 2019, pp. 5848–5854.
- [25] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3565–3572.
- [26] X. Zuo et al., "LIC-fusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5112–5119.
- [27] 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. Automat.*, 2021, pp. 5692–5698.
- [28] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R² LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7469–7476, Oct. 2021.
- [29] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution LiDAR and camera in targetless environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7517–7524, Oct. 2021.
- [30] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-LIO: Lightweight tightly coupled LiDAR-inertial odometry using parallel sparse incremental voxels," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.
- [31] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.