# Point-Line LIVO Using Patch-Based Gradient Optimization for Degenerate Scenes

Tong Shi, Kun Qian, *Member, IEEE*, Yixin Fang, Yun Zhang, and Hai Yu

***Abstract*—Simultaneous localization and mapping based on 3-D light detection and ranging (LiDAR) tends to degenerate in structural-less environments, leading to a distinct reduction in localization accuracy and mapping precision. This article proposes a point-line LiDAR-visual-inertial odometry (PL-LIVO) based on the system implementation of FAST-LIVO for robust localization in LiDAR-degenerate scenes. The key idea is integrating both points and lines into the proposed direct visual odometry subsystem (PL-DVO). By minimizing the patch-based gradient residuals for state optimization, PL-DVO provides additional constraints complementary to LiDAR. Furthermore, a LiDAR map assisted visual features depth extraction (LM-VDE) method is proposed to recover 3-D positions of visual features by mapping them onto the 3-D planes of the LiDAR map. This method is independent of the single scan's density and notable for superior generalization across various LiDAR sensors. Extensive experiments on both public datasets and our datasets demonstrate that PL-LIVO ensures robust localization and outperforms other state-of-the-art systems in LiDAR degenerate scenes.**

***Index Terms*—SLAM, localization, mapping, lidar-inertial-visual odometry, patch-based gradient optimization.**

## I. Introduction

SIMULTANEOUS localization and mapping (SLAM) stands as a crucial technique for real-time localization and 3-D reconstruction in unknown environments. 3-D light detection and ranging (LiDAR) based SLAM [1], [2], [3] systems have been widely employed in automatic robots, due to their ability to provide accurate state estimation and dense 3-D map. However, for some real-world applications such as inspection robots working on construction sites [4] and automatic vehicles driving through tunnels [5], LiDAR-based SLAM tends to degenerate in unconstrained directions (e.g., the direction of the tunnel extension). Consequently, the significant drop in localization accuracy disqualifies the system from practical application. This problem becomes more serious when the LiDAR has a small field of view (FoV) [6].

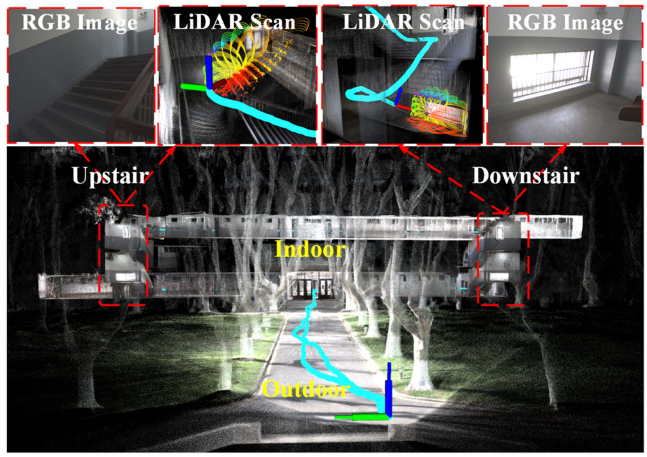


Fig. 1. localization and mapping result of PL-LIVO under the *stairs_02* sequence. In this sequence, there was a significant exposure change when the operator entered indoors from outside. Moreover, challenging degeneration occurred when the operator was moving upstairs and downstairs, attributed to the limited FoV of the LiDAR scan.

To improve the system performance in challenging scenes, many studies [7], [8] fuse the additional Inertial Measurement Unit (IMU) measurements which provide high-frequency motion data. Furthermore, by integrating visual point features of the image, several LiDAR-inertial-visual odometry (LIVO) systems [9], [10], [11] demonstrate their robust pose estimation capabilities in LiDAR-degenerate scenes. However, there are still challenges for current LIVO systems to handle degenerate environments: (1) the visual subsystems of previous methods rely heavily on the brightness invariance assumption, which is prone to be disrupted by significant exposure change and camera lens flare; (2) abundant line features in man-made environments can provide effective constraints, but remain underutilized; (3) most existing LiDAR-assisted visual feature depth extraction methods require the high density of LiDAR scans. These methods adapt poorly to relatively sparse scans from multi-line spinning LiDAR sensors.

To overcome the aforementioned limitations, we propose PL-LIVO, a point-line LIVO modified from FAST-LIVO [11], using patch-based gradient optimization for degenerate scenes. The key idea is integrating points and lines in the direct visual odometry subsystem to enhance pose estimation. Specifically,

Received 25 May 2024; accepted 8 September 2024. Date of publication 23 September 2024; date of current version 27 September 2024. This article was recommended for publication by Associate Editor B. Englot and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61573101 and in part by the Science and Technology Project of State Grid Corporation under Grant 5700-202318305A-1-1-ZN. *(Corresponding author: Kun Qian.)*

Tong Shi, Kun Qian, Yixin Fang, and Yun Zhang are with the School of Automation and the Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing 210096, China (e-mail: kqian@seu.edu.cn).

Hai Yu is with the Information and Communication Research Institute of China Electric Power Research Institute Company, Ltd., Nanjing 210000, China, and also with the School of Automation, Southeast University, Nanjing 210000, China.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3466088, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3466088

a novel patch-based gradient residual is conducted to overcome the frequent exposure change, and a novel LiDAR-assisted depth extraction method for visual features is proposed to improve the generalization across various LiDAR sensors. The detailed contributions are listed below:

1) We propose a Direct Visual Odometry subsystem integrating both Points and Lines (PL-DVO) by minimizing the patch-based gradient residual. This residual holds higher adaptability to illumination changes based on the gradient invariance assumption. A rectangular patch pattern aligned with line direction is designed for line features, enabling searching more informative pixels and establishing more accurate patch associations between different images.
2) We propose a novel LiDAR Map assisted Visual features Depth Extraction (LM-VDE) method by mapping 2-D visual features onto the 3-D planes of the LiDAR map, rather than directly fitting 3-D positions with the LiDAR scan points. Thus LM-VDE is independent of the scan's density and generalizes better across different LiDAR sensors.
3) Extensive experiments are conducted on general datasets (e.g., NTU VIRAL [12] Dataset) and degenerate datasets (collected in degenerate scenes by R3LIVE [10], FAST-LIVO [11], and our platform as Fig. 5 shows). Results show that our system achieves competitive performance in general scenes and outperforms other state-of-the-art (SOTA) methods in LiDAR-degraded scenes.

## II. Related Works

### A. LiDAR-Visual-Inertial Odometry

The integration of multiple sensors in LiDAR-visual-inertial SLAM systems mitigates individual sensor' degradation. LVI-SAM [9] incorporates depth information from LiDAR measurements to visual points and utilizes the pose estimated from the visual-inertial subsystem to initialize LiDAR scan matching. R2LIVE [13] integrates data from LiDAR, IMU, and camera within a filter-based framework. These two systems employ KLT optical flow to track feature points and conduct re-projection errors for the visual constraints. R3LIVE [10] enhances the system's state estimation by minimizing the photometric error between tracked points and the map. FAST-LIVO [11] reuses LiDAR points as visual features to enhance real-time performance.

Since line features are less sensitive to large viewpoint changes [14] and can provide more geometric constraints [15], many studies focus on integrating line features into the visual subsystem to introduce additional information. Huang et al. [16] utilized line features in LiDAR-visual odometry by constructing point-to-line re-projection residual on the image. They proposed a method for line depth extraction by minimizing 3-D point-to-line distance and 2-D re-projection distance. Cao et al. [17] built a spherical coordinate system to associate the geometric features from the visual and LiDAR subsystems, thus enabling the reconstruction of line segment depths. However, these depth extraction methods rely on the high density of LiDAR scans. In contrast, our proposed LM-VDE employs the 3-D planes stored in the LiDAR maps, requiring limited steps, and performing better adaptation to relatively sparse LiDAR sensors (e.g., 16-line spinning LiDAR).

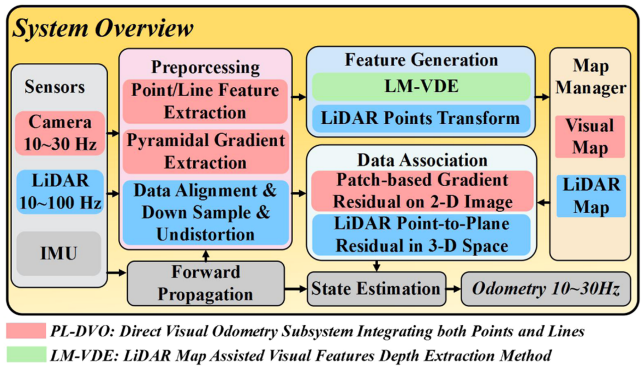


Fig. 2. System overview of the proposed PL-LIVO. This system comprises 4 basic modules: Proprocessing, Data Association, Feature Generation, and Map Manager. The measurements are fused based on an iterated error-state Kalman filter.

### B. Direct Visual Odometry

Direct methods [18], [19], [20] are among the most widely used approaches for estimating pose in visual odometry. Unlike feature-based methods that require extracting salient feature points and generating robust feature descriptors, direct methods select the gradient pixels and construct photometric errors, performing more robustly in low-texture scenes. DVL-SLAM [21] projects LiDAR scan onto the image and selects those with high-gradient to construct photometric residuals between image frames. PL-SVO [22] extends the direct method to line features by distributing points uniformly along each line segment, thus obtaining additional constraints in the structured scenes. Fang et al. [23] utilized depth measurements from the LiDAR scan to recover the positions of visual points and visual lines. Different from these visual-like systems, FAST-LIVO [11] tightly incorporates image-to-map alignment, LiDAR scan registration, and IMU measurements, improving the system's robustness in challenging environments.

These methods rely on the strict assumption of brightness invariance, which will be compromised by frequent exposure changes and camera lens flare (e.g., indoor-outdoor scenes). Some studies mitigate this issue by online photometric calibration [24], [25], but these methods introduce additional computation costs. Differently, we design a patch-based gradient optimization method that is established on a less strict assumption of gradient invariance, thereby enhancing the robustness of the method in environments with varying illumination. Moreover, a rectangular patch is designed for the visual line and aligned with the line direction, hence the corresponding patches between frames can be associated along the line.

## III. System Overview

Fig. 2 depicts the system overview of our proposed PL-LIVO. This system comprises a backbone LiDAR-inertial odometry (blue and gray components) and a direct visual odometry subsystem PL-DVO (red component). Our proposed LM-VDE (green component) method is employed to extract all visual features' depth. We illustrated the details of PL-DVO and LM-VDE in Fig. 4. The data of IMU, LiDAR, and camera are tightly coupled at the observation level.

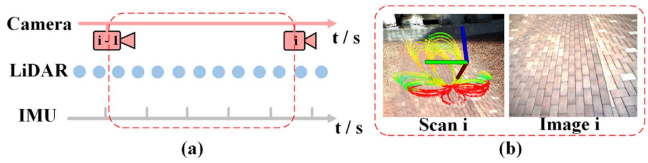


Fig. 3. Image-based Data Alignment. As (a) shows, the LiDAR scan is treated as a points stream (green and blue points from different scans). Points within the interval of two consecutive images (the red dotted box) are gathered as a new scan. As (b) shows, an aligned unit is generated with the new scan and the current image.

Initially, the input data from LiDAR and camera are fed into the Preprocessing module for data alignment and feature extraction, resulting in an aligned unit for the following steps. Concurrently, the IMU data is gathered for state propagation. Then, the aligned unit is utilized to construct observation residuals in the Data Association module. In detail, the point-to-plane residual is calculated for each LiDAR scan point, while the patch-based gradient residual is computed for each visual feature within the camera's FoV. In particular, the lines' patch pattern is shaped as a rectangle aligned with the line direction. All residuals are weighted by different factors and subsequently employed to update the state in an error-state Kalman filter framework. Furthermore, the Feature Generation module recovers 3-D coordinates of all extracted 2-D visual features by our proposed LM-VDE method. Additionally, the current LiDAR scan points are transformed into the world frame. These new features are then registered to the Visual Map and the LiDAR map respectively for further feature association and residual computation. Old features will be removed to limit memory usage.

## IV. Methodology

### A. Preprocessing

The Preprocessing module operates on the input data and generates aligned units for subsequent modules. Unlike most current methods [11], [13] that update state when receiving each data from LiDAR or camera, we conduct an image-based data alignment that supports joint optimization simultaneously. As Fig. 3(a) shows, the LiDAR scan is treated as a stream of points (green points from the last scan and blue points from the current scan). Points within the interval of two consecutive images (in the red dotted box) are collected as a new LiDAR scan, and then down-sampled and undistorted to the time of the current image. The new LiDAR scan and current image are packed as an aligned unit as Fig. 3(b) shows. Besides, point and line features on the image are detected by [26], [27] and the gradient magnitude of the pixels on the image is extracted by the Sobel operator for further computing gradient residuals.

$$
\begin{aligned}
\boldsymbol{\mathcal{M}} &\triangleq SO(3) \times \mathbb{R}^{15} \\
\boldsymbol{x} &\triangleq \begin{bmatrix} {}^W\boldsymbol{q}_b^T & {}^W\boldsymbol{v}_b^T & {}^W\boldsymbol{p}_b^T & \boldsymbol{b}_a^T & \boldsymbol{b}_g^T & {}^W\boldsymbol{g}^T \end{bmatrix}^T \in \boldsymbol{\mathcal{M}} \\
\delta\boldsymbol{x} &\triangleq \begin{bmatrix} {}^W\delta\boldsymbol{\theta}_b^T & {}^W\delta\boldsymbol{v}_b^T & {}^W\delta\boldsymbol{p}_b^T & \delta\boldsymbol{b}_a^T & \delta\boldsymbol{b}_g^T & {}^W\delta\boldsymbol{g}^T \end{bmatrix}^T \in \mathbb{R}^{18} \\
\boldsymbol{u} &\triangleq \begin{bmatrix} \boldsymbol{a}_m^T & \boldsymbol{\omega}_m^T \end{bmatrix}^T \\
\boldsymbol{w} &\triangleq \begin{bmatrix} \boldsymbol{n}_a^T & \boldsymbol{n}_\omega^T & \boldsymbol{n}_{ba}^T & \boldsymbol{n}_{b\omega}^T \end{bmatrix}^T
\end{aligned} \tag{1}
$$

The IMU data corresponding to the aligned unit (within the red dotted box in Fig. 3(a)) is gathered for state propagation. Here we define the system ground truth state $\boldsymbol{x}$, error state $\delta\boldsymbol{x}$, IMU input $\boldsymbol{u}$, and noise $\boldsymbol{w}$ as (1). $W$ and $b$ are the world frame and the IMU (body) frame, respectively. ${}^W\boldsymbol{q}_b$, ${}^W\boldsymbol{v}_b$, and ${}^W\boldsymbol{p}_b$ denote the body rotation, velocity, and position, while $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ represent the IMU biases. ${}^W\boldsymbol{g}$ signifies the gravity in the world frame. The terms in $\delta\boldsymbol{x}$ are error states relative to those in $\boldsymbol{x}$. $\boldsymbol{w}$ consists of measurement noises of IMU input $\boldsymbol{u}$ and random walk noises of IMU biases.

Using the IMU measurements as input, the nominal state $\hat{\boldsymbol{x}}$, the error state $\delta\boldsymbol{x}$, and the corresponding covariance matrix $\hat{\boldsymbol{P}}$ is propagated using a discrete-time kinematic model as (2). $f(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k)$, $\boldsymbol{F}_k$, $\boldsymbol{C}_k$, and $\boldsymbol{Q}_k$ are detailed in [28]. This propagation imposes a prior normal distribution of $\boldsymbol{x}_k$ as (3).

$$
\begin{aligned}
\widehat{\boldsymbol{x}}_{k+1} &= f\left(\widehat{\boldsymbol{x}}_k, \boldsymbol{u}_k\right) \\
\delta\boldsymbol{x}_{k+1} &= \left(I + \boldsymbol{F}_k\Delta t\right)\delta\boldsymbol{x}_k + \left(\boldsymbol{C}_k\Delta t\right)\boldsymbol{w} \\
\widehat{\boldsymbol{P}}_{k+1} &= \left(I + \boldsymbol{F}_k\Delta t\right)\widehat{\boldsymbol{P}}_k\left(I + \boldsymbol{F}_k\Delta t\right)^T \\
&\quad + \left(\boldsymbol{C}_k\Delta t\right)\boldsymbol{Q}_k\left(\boldsymbol{C}_k\Delta t\right)^T
\end{aligned} \tag{2}
$$

$$
\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k = \delta\boldsymbol{x}_k \sim \mathcal{N}\left(\boldsymbol{0}, \widehat{\boldsymbol{P}}_k\right) \tag{3}
$$

### B. Patch-Based Gradient Residuals for Visual Features

For each visual feature within the image FoV, its 3-D position is projected from the visual map onto the current image to establish a 2-D feature association with the corresponding feature on the reference image. Subsequently, the patch association around the 2-D feature is performed by extracting a patch $\mathbf{P}$ on the current image and calculating the aligned patch $\mathbf{Q}$ on the reference image. The pixel-wise differences between the associated patches are computed as the visual residual. Unlike conventional methods that gather the pixels on the grayscale image and calculate photometric residuals, we collect the pixels on the gradient image as Fig. 4(b) shows, which implies that PL-DVO operates under the gradient invariance assumption. This choice is motivated by the fact that exposure changes alter the brightness of the entire image, but have little effect on the relative brightness between the adjacent pixels. Therefore, the gradient of pixels is more stable when illumination changes.

Fig. 4(b) depicts the image patch patterns of visual points and lines. The point patch is a common $2a \times 2a$ square shape while the line patch is shaped as an $a \times 4a$ rectangle whose 4a side is aligned with the line direction. This line patch enables selecting more pixels with informative gradients and helps to associate the corresponding pixels between frames.

Given the ground truth state $\boldsymbol{x}_k$, the visual map $\boldsymbol{\mathcal{M}}_V$ and the current gradient image $\boldsymbol{I}_k$, the patch-based gradient residuals for visual features can be calculated as Algorithm 1 outlines.

For each visual point $\boldsymbol{\mathcal{P}}_i$ (lines 2–8) within the FoV of $\mathbf{I}_k$, we can get its pixel coordinate on $\mathbf{I}_k$ and extract its patch $\mathbf{P}_i$ (line 4). As Fig. 4(a) shows, the associated patch $\mathbf{Q}_i$ on the reference image is calculated through an affine transformation $\mathbf{A}_i$ to align with $\mathbf{P}_i$ (line 5). The pixel-wise difference between the aligned patches is defined as the visual point residual $\mathbf{r}_{\boldsymbol{\mathcal{P}}}$ which should be zero and can be calculated as (4):

$$
\mathbf{r}_{\boldsymbol{\mathcal{P}}}\left(\boldsymbol{x}_k, \mathcal{P}_i\right) = \mathbf{P}_i - \mathbf{A}_i\mathbf{Q}_i = \mathbf{D}^{\mathbf{P}}\left(\boldsymbol{\pi}\left({}^C\boldsymbol{T}_W\mathcal{P}_i\right)\right) - \mathbf{A}_i\mathbf{Q}_i \tag{4}
$$

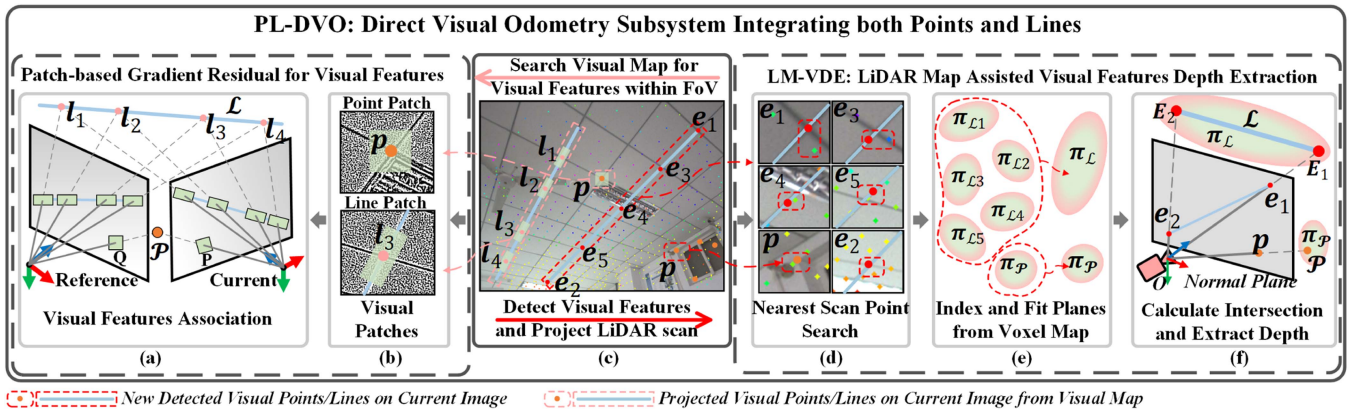


Fig. 4. Overview of the proposed PL-DVO. The proposed patch-based gradient residual for points and lines is constructed as (a). The utilized patch pattern is designed as (b). (c-f) depict the proposed LM-VDE method for extracting the depth of visual features.

**Algorithm 1:** Patch-Based Gradient Residuals.

**Data:** visual local map $\boldsymbol{\mathcal{M}}_V$, gradient image $\mathbf{I}_k$, ground truth state $\boldsymbol{x}_k$
**Result:** visual point residuals $\{\mathbf{r}_\mathcal{P}\}$, visual line residuals $\{\mathbf{r}_\mathcal{L}\}$

1 ——— **visual point residual** ———;
2 **for** *each point feature* $\boldsymbol{\mathcal{P}}_i$ *in* $\boldsymbol{\mathcal{M}}_V$ **do**
3  **if** $\boldsymbol{\mathcal{P}}_i$ *within the FoV of* $\mathbf{I}_k$ **then**
4  compute square gradient patch $\mathbf{P}_i$ ;
5  calculate associated reference patch $\mathbf{Q}_i$ by affine transformation $\mathbf{A}_i$, and compute point patch residual $\mathbf{r}_\mathcal{P}$ (4);
6  **end**
7  add $\mathbf{r}_\mathcal{P}$ to $\{\mathbf{r}_{\mathcal{P}_i}\}$ (7)
8 **end**
9 ——— **visual line residual** ———;
10 **for** *each line feature* $\boldsymbol{\mathcal{L}}_i$ *in* $\boldsymbol{\mathcal{M}}_V$ **do**
11  **if** $\boldsymbol{\mathcal{L}}_i$ *within the FoV of* $\mathbf{I}_k$ **then**
12  **for** *each sample point* $\boldsymbol{l}_m$ *on* $\boldsymbol{\mathcal{L}}$ **do**
13  compute rectangle gradient patch $\mathbf{P}_m$ aligned with line direction (5);
14  calculate associated reference patch $\mathbf{Q}_m$ by line direction and scale, and compute point-on-line patch residual $\mathbf{r}_l$ (5);
15  **end**
16  add $\mathbf{r}_l$ to line patch residual $\mathbf{r}_\mathcal{L}$ (6) ;
17  **end**
18  add $\mathbf{r}_\mathcal{L}$ to $\{\mathbf{r}_{\mathcal{L}_i}\}$ (7);
19 **end**

where $\boldsymbol{\pi}(\bullet)$ is the camera projection model and $\mathbf{D}^\mathbf{p}$ denotes the operation to extract the neighbor pixels within the $2a \times 2a$ square on the gradient image. ${}^C\boldsymbol{T}_W$ represents the transformation from the world to the camera frame, which is computed from the extrinsic ${}^C\boldsymbol{T}_b$ and the rotation and position state.

For each visual line $\boldsymbol{\mathcal{L}}_i$ (lines 10-19) within the camera FoV, $\boldsymbol{n}$ points $\{\boldsymbol{l}_m\}$ has been sampled on $\boldsymbol{\mathcal{L}}_i$ when it was extracted (see Section. IV-D). We extract the neighbor pixels of each $\boldsymbol{l}_m$ within an $a \times 4a$ rectangle on the gradient image (this operation is denoted as $\mathbf{D}^\mathbf{l}$) to form a line patch $\mathbf{P}_m$ (line 13). As Fig. 4(b) shows, the long side of the rectangle is aligned with the line direction to cover more pixels with distinct gradients. As depicted in Fig. 4(a), this also enables calculating the associated reference patch $\mathbf{Q}_m$ by orienting it along the line direction without an affine transformation (line 14). To further improve the patch alignment, a scaling factor $\boldsymbol{s}_m$ that describes the depth ratio of $\boldsymbol{l}_m$ between the current frame and the reference frame should be employed to scale the $\mathbf{Q}_m$. We define the pixel-wise difference between each pair of $\mathbf{P}_m$ and $\mathbf{Q}_m$ as $\mathbf{r}_\mathbf{l}$ which will then be accumulated as the line patch residual $\mathbf{r}_\mathcal{L}$ (line 16). $\mathbf{r}_\mathbf{l}, \mathbf{r}_\mathcal{L}$ should be zero and can be calculated as (5), (6):

$$0 = \mathbf{r}_\mathbf{l}\left(\boldsymbol{x}_k, \boldsymbol{l}_m\right) = \mathbf{D}^\mathbf{l}\left(\boldsymbol{\pi}\left({}^C\boldsymbol{T}_W \boldsymbol{l}_m\right)\right) - \boldsymbol{s}_m \mathbf{Q}_m \tag{5}$$

$$0 = \mathbf{r}_\mathcal{L}\left(\boldsymbol{x}_k, \boldsymbol{L}_i\right) = \sum_{m=1}^{n} \mathbf{r}_\mathbf{l}\left(\boldsymbol{x}_k, \boldsymbol{l}_m\right) \tag{6}$$

Both residuals will be optimized on a three-level gradient image pyramid to update the state as detailed in Section. IV-C.

### C. LiDAR Observation and State Updates

Our LiDAR map employs VoxelMap [29] that stores a collection of plane features in voxels. Given the ground truth state $\boldsymbol{x}_k$, the LiDAR local map $\boldsymbol{\mathcal{M}}_L$, and the down-sampled LiDAR scan $\{\mathbf{p}_i\}$, we search for valid point-to-plane matches and calculate the LiDAR observation residual $\mathbf{r}_\mathbf{L}(\boldsymbol{x}_k, \mathbf{p}_i)$ following [29]. $\mathbf{r}_\mathbf{L}$ is subject to a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\mathbf{L})$.

By combining the prior state distribution (3) with observation distributions of LiDAR points and visual features (4) (6), we derive the maximum a posteriori estimation for $\mathbf{x}_k$ as (7):

$$\min_{\boldsymbol{x}_k \in \mathcal{M}} \left( \left\| \boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k \right\|^2_{\widehat{\boldsymbol{P}}_k} + \sum_{i=1}^{m_L} \left\| \mathbf{r}_\mathbf{L}\left(\boldsymbol{x}_k, \mathbf{p}_i\right) \right\|^2_{\boldsymbol{\Sigma}_\mathbf{L}} + \sum_{i=1}^{m_p} \left\| \mathbf{r}_\mathcal{P}\left(\boldsymbol{x}_k, \mathcal{P}_i\right) \right\|^2_{\boldsymbol{\Sigma}_\mathcal{P}} + \sum_{i=1}^{m_l} \left\| \mathbf{r}_\mathcal{L}\left(\boldsymbol{x}_k, \mathcal{L}_i\right) \right\|^2_{\boldsymbol{\Sigma}_\mathcal{L}} \right) \tag{7}$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}$. $\boldsymbol{\Sigma}_\mathcal{P}$ and $\boldsymbol{\Sigma}_\mathcal{L}$ are two pre-defined factors to weigh $\mathbf{r}_\mathcal{P}$ and $\mathbf{r}_\mathcal{L}$ during the optimization. $m_L$ LiDAR

**Algorithm 2:** LM-VDE: LiDAR Map Assisted Visual Features Depth Extraction.

**Data:** LiDAR local map $\boldsymbol{\mathcal{M}}_L$, down-sampled LiDAR scan $\mathbf{L}_k$, detected points $\{\boldsymbol{p}_i\}$ and lines $\{\mathbf{e_1e_2}_i\}$
**Result:** 3-D points $\{\boldsymbol{\mathcal{P}}_i\}$ and 3-D lines $\{\boldsymbol{\mathcal{L}}_i\}$
1 Let $\mathbf{e_1}$, $\mathbf{e_2}$ be start point, end point of $\mathbf{e_1e_2}$;
2 Project $\mathbf{L}_k$ onto the image;
3 ——— **point depth extraction** ———;
4 **for** *each* $\boldsymbol{p}$ *in* $\{\boldsymbol{p}_i\}$ **do**
5 search on the image for the nearest LiDAR scan point $\mathbf{p}_L$ in $\mathbf{L}_k$;
6 use $\mathbf{p}_L$ index the plane features $\boldsymbol{\pi_{\mathcal{P}}}$ from $\boldsymbol{\mathcal{M}}_L$;
7 in camera frame: intersect the projective line and $\boldsymbol{\pi_{\mathcal{P}}}$ to solve depth $\mathbf{z}$ (9);
8 in camera frame: recover 3-D coordinate $\boldsymbol{\mathcal{P}}^C$ along the projective line using $\mathbf{z}$ (8);
9 transforme $\boldsymbol{\mathcal{P}}^C$ to world frame $\boldsymbol{\mathcal{P}}$;
10 add $\boldsymbol{\mathcal{P}}$ to $\{\boldsymbol{\mathcal{P}}_i\}$;
11 **end**
12 ——— **line depth extraction** ———;
13 **for** *each* $\mathbf{e}_1\mathbf{e}_2$ *in* $\{\mathbf{e_1e_2}_i\}$ **do**
14 sample $\boldsymbol{n}$ points $\{\mathbf{e}_m\}$ on $\mathbf{e}_1\mathbf{e}_2$;
15 **for** *each* $\mathbf{e}_m$ **do**
16 repeat 5 - 6, index $\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}_m}$;
17 add $\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}_m}$ to $\{\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}_m}\}$;
18 **end**
19 perform RANSAC on $\{\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}_m}\}$ and fit $\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}}$;
20 repeat 7 - 9 on $\mathbf{e}_1$, $\mathbf{e}_1$ and recover their 3-D coordinate $\mathbf{E}_1$, $\mathbf{E}_2$ in world frame;
21 parameterize $\boldsymbol{\mathcal{L}}$ with endpoint $\mathbf{E}_1$, $\mathbf{E}_2$;
22 add $\boldsymbol{\mathcal{L}}$ to $\{\boldsymbol{\mathcal{L}}_i\}$;
23 **end**

points, $m_p$ visual points, and $m_l$ visual lines are fused to optimize the state $\mathbf{x}_k$.

This optimization is solved by an iterated error-state Kalman filter, starting from the top pyramid level with the lowest resolution and progressing to subsequent levels. Since the propagated state provides the initial pose for visual patch alignment, all levels undergo only a single iteration, except for the bottom level, which undergoes two iterations using the high-resolution image. During each iteration, residuals from LiDAR and visual observations are computed using the progressively optimized state to perform the current optimization.

### D. LM-VDE and Map Update

A local visual map $\mathbf{M}_V$ is dynamically updated by inserting new visual features that enter the image and removing the features that exit the image. The depth of the new detected visual features is extracted by our proposed LM-VDE method. This method leverages the down-sampled LiDAR scan $\mathbf{L}_k$ to index 3-D planes from the local LiDAR map $\boldsymbol{\mathcal{M}}_L$, then maps the 2-D visual features on the 3-D planes. The visual points and lines that lie on the 3-D planes feature stable backgrounds and maintain relatively stable gradient distribution across different viewpoints, making them suitable for constructing residuals. We discuss LM-VDE in Algorithm 2 and detail it in the following paragraphs.

For each detected point $\boldsymbol{p}$ (lines 4-11), as Fig. 4(f) shows, its 3-D coordinate $\boldsymbol{\mathcal{P}}^C$ in the camera frame can be calculated along the projective line using (8). The depth $\boldsymbol{z}$ is ambiguous and needs to be solved. Initially, we project $\mathbf{L}_k$ onto the image and perform the nearest neighbor search to get the nearest scan point $\mathbf{p}_L$ as Fig. 4(c)–(d). $\mathbf{p}_L$ is then employed to index the leaf voxel from the LiDAR map $\boldsymbol{\mathcal{M}}_L$ as Fig. 4(e) (lines 5–6). Subsequently, if the minimum eigenvalue of all contained points in the leaf voxel is larger than a threshold, indicating a satisfactory plane $\boldsymbol{\pi_{\mathcal{P}}}$ stored in it, we can formulate the relation that $\boldsymbol{\mathcal{P}}$ lies on $\boldsymbol{\pi_{\mathcal{P}}}$ as (9) to solve $\boldsymbol{z}$ (Fig. 4(f) and line 7). At last, $\boldsymbol{\mathcal{P}}^C$ is recovered by (8) and transformed to the world frame $\boldsymbol{\mathcal{P}}$ using $\boldsymbol{x}_k$ (lines 8–10).

$$\mathcal{P}^{\mathrm{C}} = \begin{bmatrix} \boldsymbol{z}\left(\boldsymbol{p_x} - \boldsymbol{c_x}\right)/\boldsymbol{f_x} \\ \boldsymbol{z}\left(\boldsymbol{p_y} - \boldsymbol{c_y}\right)/\boldsymbol{f_y} \\ \boldsymbol{z} \end{bmatrix} \tag{8}$$

$$\left({}^{W}\boldsymbol{T}_{C}\boldsymbol{\mathcal{P}}^{\mathrm{C}} - \mathbf{c}\right) \cdot \mathbf{n}^{T} = 0 \tag{9}$$

The term $\boldsymbol{f_x}$, $\boldsymbol{f_y}$, $\boldsymbol{c_x}$, and $\boldsymbol{c_y}$ denote camera intrinsic. $\mathbf{c}$ and $\mathbf{n}$ represent the center point and the normal of $\boldsymbol{\pi_{\mathcal{P}}}$.

The methodology for point depth extraction is extended to recover line depths (13–23). For each detected line $\boldsymbol{e}_1\boldsymbol{e}_2$ on the image, its 3-D position $\boldsymbol{\mathcal{L}}$ is parametrized with the endpoints $\mathbf{E}_1$ and $\mathbf{E}_2$ in the world frame. As Fig. 4(c-d) shows, we uniformly sample $\boldsymbol{n}$ 2-D points $\{\boldsymbol{e}_m\}$ on $\boldsymbol{e}_1\boldsymbol{e}_2$ and search their nearest LiDAR points. These points are utilized to index $\boldsymbol{n}$ related planes $\{\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}_m}\}$ from $\boldsymbol{\mathcal{M}}_L$ (Fig. 4(e) and lines 14–18). Subsequently, a RANSAC is performed on $\{\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}_m}\}$ to fit a new plane $\boldsymbol{\pi}_{\boldsymbol{\mathcal{L}}}$ on which $\boldsymbol{\mathcal{L}}$ is expected to lie (Fig. 4(f) and line 19). Similar to the last paragraph, we can solve the depth of two endpoints $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ by (9) and recover their 3-D coordinates in the camera from by (8). At last, we transform the endpoints to the world frame using $\boldsymbol{x}_k$ to parameterize $\boldsymbol{\mathcal{L}}$ (lines 20–22). Moreover, $\boldsymbol{n}$ 3-D points $\{\boldsymbol{l}_m\}$ with the maximum Shi-Tomas score are sampled on $\boldsymbol{\mathcal{L}}$, and these points will be utilized to construct patch-based line residuals as Section. IV-B.

Using the updated state $\boldsymbol{x}_k$, we re-transform the down-sampled LiDAR scan $\mathbf{L}_k$ to the world frame and compute their covariances following [29]. Then, these points are inserted into the LiDAR map $\boldsymbol{\mathcal{M}}_L$ while those oldest voxels are removed.

## V. Experiments

In this section, we compared our proposed PL-LIVO with SOTA odometry systems in general scenes and challenging degenerate scenes. The generalization of our proposed LM-VDE was also verified on a relatively sparse 16-line spinning LiDAR sensor. All experiments were conducted on a computer equipped with an Intel i5-1135G7 CUP and a 16GB RAM. Moreover, we collected a dataset in degenerate scenes using the platform as Fig. 5 shows, which comprised a Livox Avia LiDAR sensor (with a built-in IMU) and an industrial camera. All sensors were hard-synchronized following FAST-LIVO2 [30]. We empirically set the parameter $\boldsymbol{a}$ of the visual patch pattern to 4 pixels to adequately cover the gradient distribution near the visual features. Both factors $\mathbf{r}_{\mathcal{P}}$ and $\mathbf{r}_{\mathcal{L}}$ that weight the patch-based visual residuals during state optimization were pre-defined as 100. We employed a 3-level gradient image pyramid, performing

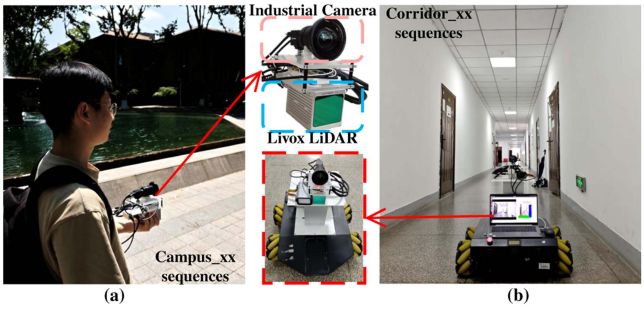


Fig. 5. Our platform with two usage configurations: (a) handheld and (b) robot-mounted. The LiDAR and the camera are hard synchronized with homogenous PWM signals.

TABLE I
ABSOLUTE TRAJECTORY ERRORS (ATE, METERS) ON NTU-VIRAL SEQUENCES

| | FAST-LIO | VoxelMap | FAST-LIVO | R3LIVE | PL-LIVO |
|---|---|---|---|---|---|
| eee_01 | 0.028 | 0.023 | 0.084 | 0.147 | **0.022** |
| eee_02 | 0.021 | 0.028 | 0.079 | 0.159 | **0.020** |
| eee_03 | 0.039 | 0.040 | 0.176 | 0.120 | **0.027** |
| nya_01 | 0.027 | **0.024** | 0.079 | 0.111 | 0.024 |
| nya_02 | 0.032 | 0.029 | 0.098 | 0.110 | **0.024** |
| nya_03 | 0.038 | 0.035 | 0.109 | 0.123 | **0.026** |
| sbs_01 | 0.041 | 0.040 | 0.099 | 0.080 | **0.029** |
| sbs_02 | 0.034 | 0.033 | 0.097 | 0.138 | **0.025** |
| sbs_03 | 0.034 | 0.034 | 0.088 | 0.146 | **0.027** |

pose optimization for each aligned unit over 4 iterations to ensure convergence.

### A. Accuracy Evaluation on the NTU VIRAL Datasets

Quantitative experiments were conducted on the 9 sequences of the public dataset NTU-VIRAL [12], which provided the ground truth of the trajectory captured by the Leica laser tracker. We compared our proposed PL-LIVO with several open-sourced SOTA odometry methods, including two LiDAR-inertial odometry, FAST-LIO [8] and VoxelMap [29], and two LiDAR-inertial-visual odometry, R3LIVE [10] and FAST-LIVO [11]. For a fair comparison, we unified the voxel size of the downsampling filter as 0.2 m for indoor scenes (*nya_xx*) and 0.5 m for outdoor scenes (*eee_xx* and *sbs_xx*). The grid sizes of the LiDAR map were unified as 0.4 m and 1.0 m for indoor and outdoor scenes, respectively. Other parameters in each system were set as the default values.

We recorded the trajectories of the IMU frame and calculated the absolute trajectory error using the official evaluation tools provided by NTU-VIRAL [12]. The results are depicted in Table I. The LiDAR subsystem of PL-LIVO is the same as VoxelMap [29], which achieved higher accuracy than FAST-LIO [8] on 6 of 9 sequences owing to its adaptive and coarse-to-fine voxel construction. Unexpectedly, the performance of two LiDAR-inertial systems surpassed the two LiDAR-inertial-visual systems. The reason is that R3LIVE [10] and FAST-LIVO [11] updated the state with each received measurement (LiDAR scan or image), so the image blur would significantly influence the pose estimation without the constraints from LiDAR. In contrast, beneficial from the image-based data alignment, PL-LIVO jointly updated the state with tightly coupled constraints from both LiDAR and image. Moreover, the proposed PL-DVO enhanced the system's accuracy because visual line features introduced more effective constraints. Consequently, PL-LIVO achieved the highest accuracy on 8 out of 9 sequences.

### B. Robustness Evaluation on the Degenerate Datasets

We evaluated the robustness of PL-LIVO on several degenerate datasets including 5 public sequences and 6 our collected sequences. For all the testing sequences, the start point and the end point were the same. Thus we used the start-to-end drift error (distance between the first and the last pose) to evaluate all the candidate methods. The results are listed in Table II.

FAST-LIO [8] and VoxelMap [29] exhibited noticeable drift errors on all sequences because LiDAR provided insufficient constraints in degenerate directions. While FAST-LIVO [11] and R3LIVE [10] mitigated degradation and reduced drifts in several sequences by incorporating constraints from visual points, they still encountered significant drifts in certain sequences. For instance, R3LIVE [10] performed poorly in *LiDAR_Degenerate* where the Livox Avia LiDAR sensor faced a tiled wall with repetitive texture. This is due to the low image sampling frequency (10 Hz) resulting in the significant disparity between consecutive frames. Consequently, the optical flow tracking fell into local minima, imposing erroneous visual constraints on pose estimation. FAST-LIVO [11] yielded unsatisfactory results in *degenerate_00* where the LiDAR sensor faced the ground and moved aggressively. This issue arises from the LiDAR sensor's proximity to the ground, leading to a restricted field of view and insufficient LiDAR-projected points to complement the visual features. Another factor resulting in FAST-LIVO's limited performance in *degenerate_xx* is the lack of hardware synchronization in these sequences. On our collected challenging dataset, both FAST-LIVO [11] and R3LIVE [10] exhibited suboptimal performance due to the low camera sampling frequency (10 Hz), along with aggressive motion and significant exposure variations. Compared to these methods, our proposed PL-LIVO extracts sufficient visual features (points and lines) and recovers their 3-D positions through LM-VDE. Additionally, it constructs more robust visual constraints by leveraging the gradient image pyramid. As a result, it achieved the highest accuracy in 10 out of 11 sequences. Notably, its start-to-end drift errors across all testing sequences were less than 0.1m, while other methods showcased fluctuating levels of drifts throughout different sequences. This demonstrates the robustness of PL-LIVO in handling degenerate environments.

We conducted an ablation study to verify the beneficial effect of the proposed patch-based gradient residuals and visual line features. As shown in Table II, two variants of PL-LIVO were tested: P-LIVO (only visual points) and L-LIVO (only visual lines). We employed the illumination residuals (brightness invariation) and the proposed gradient residuals (gradient invariation) for each variant. The result shows that those adopting gradient residuals achieved lower drifts on 9, 10, and 11 out of 11 sequences. Using the same visual residual, P-LIVO performed higher accuracy than L-LIVO on 6, 7 out of 11 sequences because the points were more common than lines and provided more visual constraints. PL-LIVO outperformed both variants on all sequences due to the complementary constraints from

TABLE II
START-TO-END DRIFT ERRORS (METERS) ON DEGENERATE SEQUENCES

| | Degenerate$_{(*)}$ Type | Camera Frequency | FAST-LIO | VoxelMap | FAST-LIVO | R3LIVE | Proposed | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Brightness Invariance | | | Gradient Invariance | | |
| | | | | | | | P-LIVO | L-LIVO | PL-LIVO | P-LIVO | L-LIVO | PL-LIVO |
| degenerate_00 | L, V2 | 20Hz | 5.282 | 4.927 | 6.062 | 0.076 | fail | fail | 3.021 | 2.070 | 7.324 | **0.061** |
| degenerate_01 | L, V2 | 20Hz | 8.847 | fail | 0.227 | 0.100 | 0.198 | 0.377 | 0.081 | 0.917 | 0.199 | **0.078** |
| degenerate_02 | L, V1 | 30Hz | fail | fail | fail | 0.102 | fail | fail | fail | 0.013 | fail | **0.010** |
| LiDAR_Degenerate | L | 10Hz | 4.732 | 3.332 | 0.034 | 8.037 | 0.042 | 0.745 | 0.021 | 0.013 | 0.021 | **0.007** |
| Visual_Challenge | L, V1 | 10Hz | 1.023 | **0.029** | 0.043 | 0.057 | 0.043 | 0.047 | 0.031 | 0.035 | 0.042 | 0.029 |
| corridor_01 | L | 10Hz | fail | fail | 7.890 | 1.166 | 0.378 | 0.117 | 0.086 | 0.099 | 0.039 | **0.014** |
| corridor_02 | L | 10Hz | fail | 3.499 | 1.122 | fail | 0.242 | 5.952 | 0.178 | 0.083 | 0.780 | **0.017** |
| stairs_01 | L, V2 | 10Hz | 2.134 | 2.312 | 3.812 | 3.046 | fail | 2.609 | 1.047 | 0.173 | 0.682 | **0.046** |
| stairs_02 | L, V2 | 10Hz | fail | 4.394 | 7.913 | fail | 9.122 | 2.210 | 1.351 | 0.522 | 0.359 | **0.010** |
| campus_01 | L | 10Hz | 2.539 | fail | 1.655 | 3.937 | 0.081 | 0.198 | 0.041 | 0.189 | 0.045 | **0.024** |
| campus_02 | L, V2 | 10Hz | 2.393 | fail | 2.946 | 9.298 | 2.962 | fail | 1.258 | 0.326 | 0.920 | **0.053** |

[*] L denotes LiDAR degeneration, V1, V2 denote camera faces a white wall or undergoes significant exposure changes, respectively.
[1] The best results are marked in bold and the second-best results are underlined
[2] fail means a drift error $\geq$ 10 m.

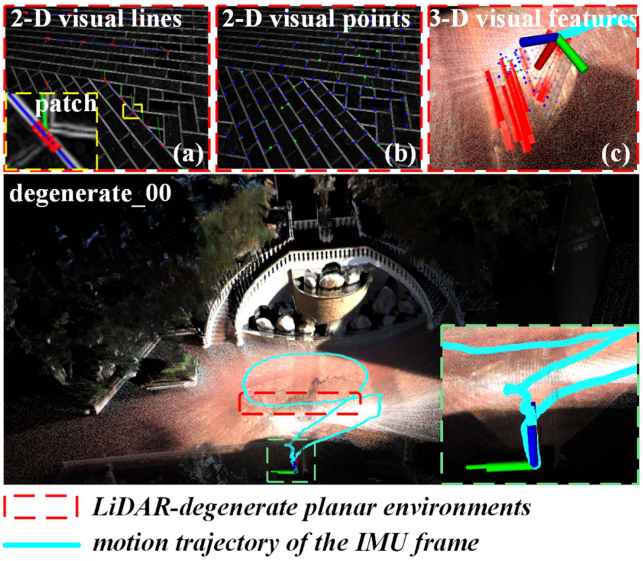


Fig. 6. Localization and mapping result of PL-LIVO under the *degenerate_00* sequence. In this sequence, the LiDAR experienced degradation while oriented towards the ground. Due to the constraints provided by the gradient residuals of visual points and lines, PL-LIVO achieved stable localization and accurately returnd to the origin.

visual points and lines, illustrating the benefit of additional line features for pose estimation.

Fig. 1 demonstrates that PL-LIVO achieved stable localization and high-precision mapping despite significant exposure changes, LiDAR degradation, and IMU fluctuations in the *stairs_01* sequence. Fig. 6 demonstrates the robust localization and mapping capabilities of PL-LIVO in the *degenerate_00* sequence, even with severe LiDAR degradation and a highly constrained field of view. Additionally, Fig. 6(a)–(b) depicts the 2-D visual lines and points on the gradient image, while Fig. 6(c) shows their corresponding 3-D points and lines.

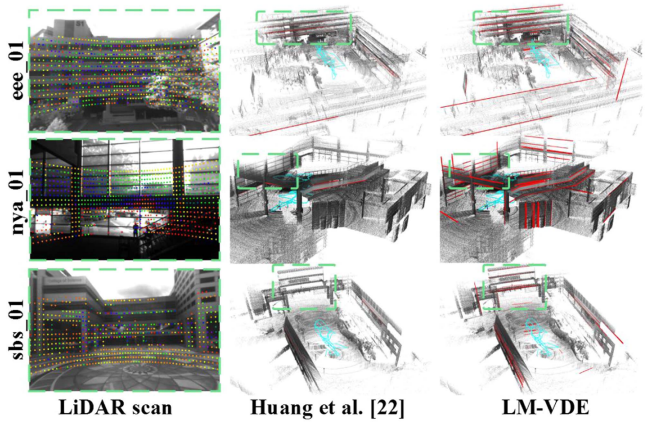


Fig. 7. Line map constructed by Huang et al. [16] and LM-VDE.

### C. Line Depth Extraction Method Evaluation

To demonstrate the generalization of our proposed LM-VDE across different LiDAR sensors, we compared LM-VDE with the depth extraction method proposed by Huang et al. [16]. Since Livox LiDAR scans were of high density, both LM-VDE and [16] could recover the depth of most detected lines. However, the extraction task became more challenging when dealing with relatively sparse scans from the spinning LiDAR. For this evaluation, we visualized and compared the line map constructed using both two methods on the NTU-VIRAL [12] *eee_01*, *nya_01*, and *sbs_01* sequences. The results, depicted in Fig. 7, indicate that LM-VDE recovered more lines than [16]. This discrepancy illustrates LM-VDE's superior adaptability to relatively sparse LiDAR data, highlighting its generalization across diverse LiDAR sensors.

### D. Time Consumption Evaluation

In this section, we evaluate the average processing time of the full PL-LIVO system on our collected dataset, including indoor and outdoor environments. The average processing time

TABLE III
AVERAGE PROCESSING TIME (MILLISECONDS) ON OUR COLLECTED DATASET

| Modules | Process | Time |
|---|---|---|
| Preprocessing | Image Preprocessing | 11.0 |
| | LiDAR Scan Preprocessing | 1.2 |
| | IMU Propagation | 1.2 |
| Data Association(*) | Visual Observation Model | 6.6 |
| | LiDAR Observation Model | 25.6 |
| State Estimation(*) | IESKF Update | 5.6 |
| Feature Generation and Map Manager | LM-VDE and Visual Map Update | 4.5 |
| | LiDAR Map Update | 1.6 |
| PL-LIVO | | 57.3 |

* The module undergoes 4 iterations during optimization, and the table presents the total time consumption of the 4 iterations.

of PL-LIVO per aligned unit is 57.3 ms, whereas FAST-LIVO [11] achieves a processing time of 49.3ms. This additional computational overhead arises from the extraction of 2-D visual features and the reconstruction of their 3-D positions. Nevertheless, our system maintains real-time performance and improves robustness in degenerate environments. We further detail the average processing time of PL-LIVO in Table III.

## VI. CONCLUSION

In this letter, a point-line LiDAR-visual-inertial odometry (PL-LIVO) is proposed that outperforms previous methods in mitigating LIDAR degeneration. The patch-based gradient optimization improves the robustness of the method against exposure changes and lens flare. A LiDAR map assisted visual features depth extraction (LM-VDE) method is also proposed to recover the 3-D coordinates of visual lines, resulting in superior adaption to relatively sparse LiDAR data as well as more line features being recovered. Comparative evaluations demonstrate that PL-LIVO achieves higher accuracy in general environments and superior robustness in challenging degenerate scenes. In practical applications, PL-LIVO can be used for inspection robots or handheld mapping equipment working in challenging environments. However, it is worth mentioning that PL-LIVO requires high-precision calibration results between the camera and liDAR. For future works, we plan to extend our research by introducing online extrinsic calibration utilizing point and line features, thereby creating a more application-friendly system.

## REFERENCES


[1] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, Jul. 2014, vol. 2, pp. 1–9.

[2] H. Shen, Q. Zong, B. Tian, and H. Lu, "Voxel-based localization and mapping for multirobot system in GPS-denied environments," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10333–10342, Oct. 2022.

[3] Y. Fang, K. Qian, Y. Zhang, T. Shi, and H. Yu, "Segmented curved-voxel occupancy descriptor for dynamic-aware LiDAR odometry and mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5701916.

[4] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, "The hilti SLAM challenge dataset," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7518–7525, Jul. 2022.

[5] I. Filip, J. Pyo, M. Lee, and H. Joe, "LiDAR SLAM comparison in a featureless tunnel environment," in *2022 22nd Int. Conf. Control, Automat. Syst.*, Nov. 2022, pp. 1648–1653.

[6] H. Li, B. Tian, H. Shen, and J. Lu, "An intensity-augmented LiDAR-inertial SLAM for solid-state LiDARs in degenerated environments," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 8503610.

[7] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2020, pp. 5135–5142.

[8] 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.

[9] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *2021 IEEE Int. Conf. Robot. Automat.*, Jun. 2021, pp. 5692–5698.

[10] J. Lin and F. Zhang, "R$^3$LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *2022 Int. Conf. Robot. Automat.*, May 2022, pp. 10672–10678.

[11] 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 *2022 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2022, pp. 4003–4009.

[12] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T. H. Nguyen, and L. Xie, "NTU VIRAL: A visual-inertial-ranging-LiDAR dataset, from an aerial vehicle viewpoint," *Int. J. Robot. Res.*, vol. 41, no. 3, pp. 270–280, 2022.

[13] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R $^2$ 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.

[14] A. P. Gee and W. Mayol-Cuevas, "Real-time model-based SLAM using line segments," in *Proc. Int. Symp. Vis. Comput.*, 2006, pp. 354–363.

[15] S. Lin, X. Zhang, Y. Liu, H. Wang, X. Zhang, and Y. Zhuang, "FLM PL-VIO: A robust monocular point-line visual-inertial odometry based on fast line matching," *IEEE Trans. Ind. Electron.*, early access, Apr. 23, 2024, doi: 10.1109/TIE.2024.3379661.

[16] S.-S. Huang, Z.-Y. Ma, T.-J. Mu, H. Fu, and S.-M. Hu, "LiDAR-monocular visual odometry using point and line features," in *2020 IEEE Int. Conf. Robot. Automat.*, Jun. 2020, pp. 1091–1097.

[17] K. Cao et al., "Tightly-coupled LiDAR-visual SLAM based on geometric features for mobile agents," in *2023 IEEE Int. Conf. Robot. Biomimetics*, 2023, pp. 1–8.

[18] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 834–849.

[19] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE Int. Conf. Robot. Automat.*, 2014, pp. 15–22.

[20] J. Jiang, J. Yuan, X. Zhang, and X. Zhang, "DVIO: An optimization-based tightly coupled direct visual-inertial odometry," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11212–11222, Nov. 2021.

[21] Y.-S. Shin, Y. S. Park, and A. Kim, "DVL-SLAM: Sparse depth enhanced direct visual-LiDAR SLAM," *Auton. Robots*, vol. 44, no. 2, pp. 115–130, Jan. 2020.

[22] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "PL-SVO: Semi-direct monocular visual odometry by combining points and line segments," in *2016 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 4211–4216.

[23] B. Fang, Q. Pan, and H. Wang, "Direct monocular visual odometry based on LiDAR vision fusion*," in *2023 WRC Symp. Adv. Robot. Automat.*, Aug. 2023, pp. 256–261.

[24] P. Bergmann, R. Wang, and D. Cremers, "Online photometric calibration of auto exposure video for realtime visual odometry and SLAM," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 627–634, Apr. 2018.

[25] Z. Feng, J. Li, L. Zhang, and C. Chen, "Online spatial and temporal calibration for monocular direct visual-inertial odometry," *Sensors*, vol. 19, no. 10, May 2019, Art. no. 2273.

[26] J. Shi and Tomasi, "Good features to track," in *1994 Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.

[27] I. Suárez, J. M. Buenaposada, and L. Baumela, "ELSED: Enhanced line segment drawing," *Pattern Recognit.*, vol. 127, Jul. 2022 , Art. no. 108619.

[28] J. Sola, "Quaternion kinematics for the error-state Kalman filter," 2017, *arXiv:1711.02508*.

[29] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8518–8525, Jul. 2022.

[30] C. Zheng et al., "FAST-LIVO2: Fast, direct LiDAR-inertial-visual odometry," 2024, *arXiv:2408.14035*.