MetroLoc: Metro Vehicle Mapping and Localization With LiDAR-Camera-Inertial Integration

Yusheng Wang, Weiwei Song, Yapeng Wang, Xinye Dai, and Yidong Lou

Abstract—In this paper, we propose an accurate and robust multi-modal sensor fusion framework, MetroLoc, towards one of the most extreme scenarios, the large-scale metro environments. MetroLoc is built atop an IMU-centric state estimator that tightly couples light detection and ranging (LiDAR), visual, and inertial information with the convenience of loosely coupled methods. The proposed framework is composed of three submodules: IMU odometry, LiDAR-inertial odometry (LIO), and Visual-inertial odometry (VIO). The IMU is treated as the primary sensor, which achieves the observations from LIO and VIO to constrain the accelerometer and gyroscope biases. Compared to previous point-only LIO methods, our approach leverages more geometry information by introducing both line and plane features into motion estimation. The VIO also utilizes the environmental structure information by employing both lines and points. Our proposed method has been tested in the long-during metro environments with a maintenance vehicle. Experimental results show the system more accurate and robust than the state-of-the-art approaches with real-time performance. The proposed method can reach 0.278% maximum drift in translation even in the highly degenerated tunnels. Besides, we develop a series of Virtual Reality (VR) applications towards efficient, economical, and interactive rail vehicle state and trackside infrastructure monitoring tasks.

Index Terms—Metro vehicle, sensor fusion, mapping and positioning, train localization.

I. INTRODUCTION

A. Motivation

TRAIN positioning and railway monitoring is of critical importance for railroad systems since either a train localization failure or a railroad clearance intrusion might lead to fatal accidents. The train positioning strategy nowadays is dominated by trackside infrastructures like track circuits and Balise. The latter approach models the train positioning as a 1D problem and divides the railway track into separate cantons, with a Balise placed at the beginning of each canton. When a train passes over one Balise, the Automatic Train Control (ATC) system knows that a train is within that canton.

Received 1 November 2021; revised 21 May 2022, 22 April 2024, and 10 September 2024; accepted 28 November 2024. Date of publication 17 December 2024; date of current version 4 February 2025. This work was supported by the Joint Foundation for Ministry of Education of China under Grant 6141A0211907. The Associate Editor for this article was C. Ding. (Corresponding author: Weiwei Song.)

Yusheng Wang is with CHC Navigation, Wuhan 430073, China (e-mail: yushengwu@whu.edu.cn).

Weiwei Song, Yapeng Wang, Xinye Dai, and Yidong Lou are with the GNSS Research Center, Wuhan University, Wuhan 430079, China (e-mail: sww@whu.edu.cn; wangyp@whu.edu.cn; xinyed@whu.edu.cn; ydlou@whu.edu.cn).

Digital Object Identifier 10.1109/TITS.2024.3512000

If another train is detected within the safe limits to this canton, the train is stopped automatically by Automatic Train Protection (ATP) system. Therefore, the train positioning accuracy is restricted by the length of a canton. Besides, the trackside infrastructure-based systems require substantial capital investment and inherent maintenance cost, where hundreds of Balises are needed in a single railway line and each Balise means thousands of dollars.

With the rapid development of sensor technologies and the worldwide standardization of railroad signal systems, new on-board localization sensors have found a gap in rail vehicles, where they can supplement the limitations of trackside ones. The precise location of the rail vehicles can be obtained through global navigation satellite system (GNSS), such as the global positioning system (GPS) and Beidou. There are numerous works on precision evaluation of GNSS for rail vehicles and data fusion with IMU, odometer as well as track geometry map. These localization methods merely acquire the vehicle positioning data without extra perceptual information, which can be further utilized to operation environment monitoring and hazard forecasting. For instance, a detected crack may prevent a potential tunnel collapse and an indicated powerline failure may inhibit an accident in the future. Besides, the GNSS-based methods cannot work for subway trains which runs underground.

With the framework of estimating odometry and mapping the surroundings at the same time, simultaneously localization and mapping (SLAM) is a promising solution to this tricky conundrum. Recent advances in light detection and ranging (LiDAR) hardware have cultivated research of LiDAR SLAM [1], [2], [3]. The accurate range measurement, invulnerability to illumination variations and long detection range of LiDARs make them suitable of localization, navigation, and mapping. They have been applied to supplement the IMU odometry.

However, in environments with degenerate geometries, such as the long metro tunnels with repetitive structures, LiDAR-only approaches may fail if frame-to-frame correspondences are estimated using highly consistent scans. Since the IMU mechanization alone cannot provide reliable pose estimates for more than a few minutes, the system failure is often nonreversible. To cope with these situations, three kinds of solutions have been discussed in the scholarly works.

The first is through analyzing feature distribution, and determining the degenerate directions in the state space. Thus, the pose estimation is partially solved in the well-conditioned...
directions. However, this approach is limited in the small scale, where drifts are inevitable in the unobservable directions for large scale applications. The second solutions leverage the geometrical regularities in the environments, such as planes, lines, and vanishing points. The third approaches directly add additional measurements to the system. Such measurements include the visual, radar, and wireless communication sensors.

B. Challenges

We are motivated to tackle the metro vehicle localization and mapping problem with the latter two approaches. There are many similarities between underground mine and metro tunnels, such as low illumination, textureless and self-repetitive. But the metro vehicle mapping and localization is a much more demanding task over tunnel exploration for three reasons:

High speed: the unmanned aerospace vehicles (UAV), unmanned ground vehicles (UGV) and legged robots used for fast deployment all run with low velocities. On the other hand, the metro vehicles have a much larger velocity, which raises the awareness of computation efficiency and velocity-related optimizations, such as distortion removal.

Long journey: many sub terrain datasets only have hundreds of meters length coverage whereas, the distance between two metro stations is already more than one kilometer. The accumulated odometry errors will be incredibly exaggerated without proper compensation methods. Besides, commonly used incremental map association methods usually build the map by registering new scans, leaving the accumulated errors unsolved. In addition, metro scenes are more structured than the caves.

No revisited areas: although the metro vehicles run in a circle, there are no loops for a one-way operated transportation system, thus no loop can be detected at the back-end.

Highly constrained motions: the rail vehicles are constrained to move along planar trajectories, leading to insufficient information gain to the unobservable directions for IMU biases. This potential observability issue will result in large scale drift for many Visual-inertial approaches.

C. Main Contributions

Addressing the problems mentioned above, we present MetroLoc, a specific study on LiDAR-visual-inertial fusion for metro vehicle localization and mapping in this paper. The main contributions of our work can be summarized as follows:

- We propose a real-time and lightweight simultaneous localization and mapping scheme for large-scale map building in perceptually-challenging metro tunnels. Unlike previous LiDAR or visual centric approaches, our IMU-centric formulation fully leverages the inertial information in both the state estimation and following LiDAR or visual process.
- We explore the specific features in railroad. Instead of directly extracting planar points on the railroad, we leverage the rail tracks to register track planes. This process ensures highly accurate ground constraints.
- We utilize the geometry regularities to supplement both the LiDAR and visual correspondence tracking, which significantly improve the accuracy and robustness in structured areas.

To the best of the authors’ knowledge, MetroLoc is the first solution to real-time and large-scale metro vehicle SLAM. Some of the mapping results are shown in Fig. 1, demonstrating that our system is of high precision.

D. Organization

The rest of the article is organized as follows. Section II reviews the relevant work. Section III gives the overview of the system along with some preliminary. Section IV presents the detailed IMU-centric graph optimization process applied in our system, followed by experimental results in Section V. Finally, Section VI concludes the article and demonstrates some future research directions.

II. RELATED WORK

Few restrictions exist for the localization solution of rail applications regarding the weight and power consumption. Therefore, researchers are pretty open in choosing suitable sensor modalities and estimation algorithms. Prior works on multi-modal sensor fusion has gained great popularity using combinations of LiDAR, camera and IMU and can be classified into either loosely or tightly coupled methods. The former scheme processes the measurements from individual sensors separately, and they are preferred more for their extendibility and low computation consumption. In contrast, tightly coupled methods jointly optimize sensor measurements to obtain odometry estimation, with favorable accuracy and robustness.

A. Train Localization and Mapping Solutions

The existing train positioning strategy is mainly dependent on trackside infrastructures like track circuits and Balises. Since the accuracy of these systems is determined by the operation interval, they are neither accurate nor efficient for intelligent rail transportation systems. Considering its large capital investment and low efficiency, many researchers seek to complement the system limitations with onboard sensors. The satellite-based methods utilizes the GNSS for train positioning,
and the accuracy can be further improved with integrated track odometry, wheel odometer, and IMU [4]. However, these methods merely achieve train state information without awareness of environmental information.

The current railroad environment monitoring is still a human-intensive work, and manual inspections are scheduled regularly. Although the visual-only approaches have been largely investigated, they are inaccurate for range measuring and sensitive to illumination conditions. In many of the previous works, laser scanners have been included in the mobile mapping system for railroad monitoring tasks. As a direct geo-referencing approach, this system requires high-precision GNSS/IMU determination and survey-grade laser scanners. Although these solutions can achieve highly-accurate 3D maps, they are costly for large deployment and less-efficiency for real-time perception.

The aforementioned conflicts can be solved by SLAM solution. The performance of Visual-inertial odometry on rail vehicles has been extensively evaluated in [5], indicating that the monocular Visual-inertial odometry is not reliable for railroad applications. Unfortunately, the 3D LiDAR or LiDAR-visual based SLAM for railroad has not been well investigated.

**B. Handling Degeneracy in SLAM**

In many of the recent works, [6], [7], [8], [9], degeneracy is modeled as a factor and estimated online to prevent estimation failure. This is done by evaluation of geometric structure of the problem constraints, where associated eigenvalues and eigenvectors are computed and analyzed. When degeneracy is detected, such method automatically separates the state space, left the problem solved only in well-conditioned directions. Zhen and Scherer define the localizability vector through projecting information matrix into the eigenspace, and model the degeneration with a frictionless force-closure [10]. Tagliabue et al. [7] also use the smallest eigenvalue of point-to-plane cost to indicate the least observable direction. Instead of solving the degeneracy problem using the LiDAR-inertial SLAM directly, it will switch to other parallel-running odometry algorithms [11] when the metric is below a self-defined threshold. Based thereupon, a feature-selection method is proposed in [12]. With the assumption of the geometric constraints contribute differently to the localization accuracy, this approach defines the most informative ones as good features through degeneracy evaluation. Since the data association and state estimation utilizes good features only, both the pose estimation accuracy and computation efficiency can be boosted. The above methods require an empirical value of degeneracy detection, which may vary in different scenarios. Besides, the accumulated errors in unsolved directions are non-negligible for large scenarios.

As line and planar features are dominant in indoor environments, researcher have attempted to use planes as the landmarks [13], [14], [15], [16]. In these works, planes are parameterized based on the quaternion, and the convergence speed is improved thorough the relative plane formulation. Since the point-to-plane cost converges faster and generates accurate results, the recent work [13], [14] adopts the point-to-plane cost instead of the plane-to-plane cost. Zhou et al. [13] make a further improvement to the scan registration through applying first-order Taylor expansion. However, the above works all use random sample consensus (RANSAC) [17] to extract and refine ground planes, which is non-applicable for rail tracks. In practice, the sleepers and trackside infrastructures all make the plane extraction a difficult problem.

To address this problem, we propose to leverage the virtual plane constraints generated by rail tracks for further refinement.

**C. Loosely Coupled LiDAR-Visual-Inertial Odometry**

One of the early works of LiDAR-visual-inertial, V-LOAM, is proposed in [18], which leverages the Visual-inertial odometry as the motion model for LiDAR scan matching. Since this scheme only performs frame-to-frame motion estimation, the global consistency is not guaranteed. To cope with this problem, Wang et al. propose a direct Visual-LiDAR fusion scheme DV-loam [19]. The coarse states are estimated using a two-stage direct visual odometry module, and are further refined by the LiDAR mapping module, finally a Teaser-based [20] loop detector is utilized for correcting accumulated drifts. The robustness of the loosely coupled system can be further increased through incorporating additional constraints, such as thermal-inertial prior for smoky scenarios, incremental odometry or legged odometry for autonomous exploring robots.

**D. Tightly Coupled LiDAR-Visual-Inertial Odometry**

In many of the recent works, tight integration of multi-modal sensing capabilities are explored for degeneracy avoidance and robustness enhancement. The filter-based approaches employs a Kalman filter for joint state estimation, such as the error state Kalman filter (ESKF) utilized in [23] and the multi-state constraint Kalman filter (MSCKF) applied in [26]. The authors of the latter work refine a novel plane feature tracking across multiple LiDAR scans within a sliding-window, making the pose estimation process more efficient and robust. The filter-based methods are usually less extendible to other sensors and may be vulnerable to potential sensor failures. In contrast, the optimization-based approaches have proved advantageous for their expandability, where each sensor input can be encapsulated as a factor in the graph.

Shan et al. proposes LVI-SAM in [24], where the LiDAR-inertial and Visual-inertial subsystems can run jointly in feature-rich scenarios, or independently with detected failures in one of them. However, the “point to line” and “point-to-plane” based LiDAR odometry factor cost functions are not robust to feature-poor environments, and may generate meaningless result in metro or cave scenarios. To handle these situations, the geometry of 3D primitives are employed with the line and plane landmarks extracted in [25]. The robustness and accuracy of this system has been proved with two small-scale DARPA SubT datasets (one 167 m long, the other is 490 m long). In addition, the performance of Visual-inertial subsystem in structured man-made environments can also be improved with the detected line features in [26].
From the discussion of the above, we can see that the LiDAR-visual-inertial integrated pose estimation and mapping has not been well solved in large-scale datasets. To address this problem, this paper aims to achieve low-drift and robust odometry and mapping for large-scale metro scenarios.

III. SYSTEM OVERVIEW

The overview of our system is shown in Fig. 2, which is composed of three subsystems: IMU odometry, LiDAR-inertial odometry (LIO), and Visual-inertial odometry (VIO). Our system treats the IMU as the primary sensor as long as the bias can be well-constrained by other sensors. The constrained IMU odometry provides the prediction to the VIO and LIO. The LIO and VIO submodule extracts features from raw scans and images, which are used for state estimation. Both the two modules leverage the factor graph optimization to refine the poses. Finally, the IMU odometry submodule achieves these observations to constrain the accelerometer bias and gyroscope.

The notations used throughout this paper is shown in TABLE I. In addition, we define \( \Phi^B_W \) as the transformation from world frame to the IMU frame. The \( k \)-th vehicle state vector \( x_k \) can be written as:

\[
x_k = \begin{bmatrix} p^W_{1k}, v^W_{1k}, q^W_{1k}, b_{1k}, b_{1k} \end{bmatrix}
\]  

(1)

where \( p^W_{1k} \in \mathbb{R}^3, v^W_{1k} \in \mathbb{R}^3, \) and \( q^W_{1k} \in \text{SO}(3) \) are the position, linear velocity, and orientation vector. The last two elements are the IMU gyroscope and accelerometer biases.

A. Multi-Sensory Calibration

The intrinsic and extrinsic of sensors are configured offline through open source packages. Since the mounting errors are inevitable each time, these parameters are recomputed before every data gathering process.

B. Time Synchronization

The LiDAR, IMU, and camera are hardware synchronized with an external rubidium clock using pulse per second (pps) signal. The timestamp is unified and stored for convenient per-frame comparison with post-processing results.

IV. METHODOLOGY

We use a factor graph to model this maximum a posterior (MAP) problem, and we adopt three types of factors for graph construction as shown in Fig. 2, namely: (1) IMU odometry factors; (2) LIO factors; (3) VIO factors.

A. IMU Odometry Factors

Following [27], we add the accelerometer and gyroscope bias \( b_a \) and \( b_g \) into the estimated state of IMU odometry. The edge between two consequent state nodes \( i \) and \( j \) can be obtained from the relative motion measurement. The residual of preintegrated IMU measurements \( [\alpha^B_{1i}, \beta^B_{1i}, \gamma^B_{1i}] \) can be formulated by:

\[
e^I_{ij}^{MU} = \begin{bmatrix} \delta \alpha^B_{1i} & \delta \beta^B_{1i} & \delta \gamma^B_{1i} & \delta b_a & \delta b_g \end{bmatrix}^T
\]

\[
= R^B_W \left( \begin{bmatrix} p^W_{1i} - p^W_{1j} + \frac{1}{2} g^W W(t^2 - v^W_{1i} \Delta t) - \hat{\alpha}^B_{1j} \\ v^W_{1i} + g^W \Delta t - v^W_{1j} - \hat{\beta}^B_{1j} \\ \frac{1}{2} \left[ \begin{bmatrix} \delta W^0_{1} \right]^{-1} \begin{bmatrix} \delta W^0_{1} \end{bmatrix} \begin{bmatrix} \delta W^0_{1} \end{bmatrix}^{-1} \end{bmatrix} \right] \\
\right)
\]

(2)

where \( g^W = [0, 0, g]^T \) is the gravity factor in the world frame and \( \Delta t \) is the time sweep between \( i \) and \( j \). Since the
VIO and LIO are both used to constrain IMU preintegration measurements, we can obtain $\mathbf{p}_{B}^{H}$, $\mathbf{V}_{B}^{H}$ and $\mathbf{q}_{B}^{H}$ from them. Besides, the bias errors are jointly optimized in the graph. With the IMU residuals $e_{ij}^{LIO}$ and $e_{ij}^{VIO}$ calculated by LIO and VIO, the IMU odometry optimization problem can be defined by:

$$E = \sum (e_{ij}^{LIO})^T \mathbf{W}_{ij}^{-1} e_{ij}^{LIO} + \sum (e_{ij}^{VIO})^T \mathbf{W}_{ij}^{-1} e_{ij}^{VIO} + E_{m}$$

(3)

where $\mathbf{W}_{ij}$ is the covariance matrix and $E_{m}$ is the marginalized prior. Since both the pose estimation of LIO and VIO are not globally referenced, we only utilize the relative state estimation as local constraints to correct the bias of IMU preintegration.

B. LIO Factors

As illustrated in Fig. 3, we utilize an image-grade hybrid solid state LiDAR Innovusion Jaguar Prime\(^1\) throughout our experiments. The point cloud within the 65° × 40° field of view (FoV) has a similar density coverage with that of a 300-line rotating LiDAR. A mechanical spinning LiDAR perceives the surrounding environment by rotating a vertically aligned laser beam array. Thus, the density and angular resolution is determined by the number of laser photodiodes. On the contrary, the image-grade LiDAR has an advanced optical apparatus design as illustrated in Fig. 4, with one rotating polygon mirror focusing and guiding the laser beams [28], achieving a similar point cloud density with that of a high resolution rotating LiDAR sensor.

1) Points adjacent to the margin of FoV. It is seen the outliers lie mostly on the scan fringe. We hereby discard a 1-degree fringe region on both sides, and retain a 63-degree horizontal view for the Jaguar Prime LiDAR.

2) Points behind objects. It can be noticed that some outliers are ‘hidden’ behind the foreground objects. We hereby leverage the empirical value described in [29] to remove such outliers.

3) Underground points. Many of the hybrid solid state LiDARs have the problem of mirroring high intensity objects, such as a above ground road sign may be mirrored to the underground. Therefore, the points right below the detected ground are removed.

With the assumption of a constant angular velocity and linear velocity between two consecutive LiDAR frames, we can use the preintegration results to correct the distortion in the current frame. We then project the point cloud into a 2D range image. For the $k$-th scan input $S_k = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$, it is computed based on the horizontal and vertical angle of each point. Given a point $\mathbf{p}_i = (x_i, y_i, z_i)$, the horizontal angle $\alpha_i$ and vertical angle $\theta_i$ can be calculated by:

$$\alpha_i = \arctan \left( \frac{z_i}{y_i} \right),$$

$$\theta_i = \arctan \left( \frac{x_i}{y_i} \right).$$

(4)

Note that unlike mechanical spinning LiDARs having horizontal $xy$-plane, the coordinate of image-grade LiDAR has a similar definition with that of a camera, where the $xy$-plane is parallel to the image plane. Then we apply the feature extraction method in [30] through counting local smoothness. To make the distribution of features in space as uniform as possible, we divide the range image horizontally into equal sub blocks. Then we sort the points in each row and distinguish features upon thresholds. In contrast with LOAM or LeGO-LOAM using a same threshold, we set different thresholds based on the distance of points.

The LiDAR odometry seeks to estimate the sensor motion between two consecutive scans, which is solved by performing point-to-line and point-to-plane scan matching. For each edge point $\mathbf{p}_1^e$ in the current scan, we search two points from its nearest neighbor in the local map, $\mathbf{p}_2^e$ and $\mathbf{p}_3^e$. In order to increase the searching efficiency, we build local maps for edge and planar feature. The point-to-line residual $d_{e2e}$ can then be formulated by:

$$d_{e2e} = \frac{|(\mathbf{p}_2^e - \mathbf{p}_1^e) \times (\mathbf{p}_2^e - \mathbf{p}_3^e)|}{|\mathbf{p}_1^e - \mathbf{p}_2^e|}.$$  

(5)

Similarly, for each planar point $\mathbf{p}_1^o$, we search for 3 nearest points from the local planar feature map $\mathbf{p}_1^o, \mathbf{p}_2^o$, and $\mathbf{p}_3^o$. The point-to-plane residual $d_{p2p}$ is established by:

$$d_{p2p} = \frac{|(\mathbf{p}_1^o - \mathbf{p}_1^e)^T ((\mathbf{p}_1^o - \mathbf{p}_2^o) \times (\mathbf{p}_1^o - \mathbf{p}_3^o))|}{|\mathbf{p}_1^o - \mathbf{p}_2^o| \times |\mathbf{p}_1^o - \mathbf{p}_3^o|}.$$  

(6)

Suppose the number of edge and planar correspondences is $N_e$ and $N_p$ in the current frame, the residual can be

\(^{1}\)https://www.innovusion.com/jaguar-prime
calculated using:
\[
e^{LIO_{k \rightarrow iL}} = \sum_{i=1}^{N_k} (d_{i2k})^2 + \sum_{j=1}^{N_p} (d_{j2p})^2.
\]

(7)

Considering that the high point cloud density, employing all the feature points for state estimation is not efficient. In practice, the implementation of LOAM cannot run real time with a high-performance laptop. We can derive a large amount of constraints from (5) and (6), which is of great burden for real-time optimization. It will be promising if we can exclude some less informative features within the acceptable range of accuracy loss, instead of employing all of the information. In other words, this manner should keep a balance between correspondence amounts and the accuracy. In the field of satellite navigation, many researches have utilized the Dilution of Precision (DOP) to quantify the position errors infected by geometry distribution of satellites. We hereby seek to select the most informative features with best geometry distributions. Using the Jacobian matrix J of LiDAR odometry residual, the information matrix can be defined by:

\[
\Lambda = J_k^T \sum_k^{-1} J_k.
\]

(8)

There are several information matrix based feature selection methods, such as trace, minimum eigenvalue, and log determinant. We follow the last approach to define the good feature selection process as:

\[
\arg \max_{\bar{G}_k} \log det (\Lambda),
\]

(9)

where \(\bar{G}_k\) represents the ideal feature set, and \(\log det\) is the log determinant function. We employ the stochastic greedy method \([31]\) to solve this problem and the result is visualized in Fig. 5. Note that the points selection threshold is set according to the current and empirical degeneracy factor \(\lambda_i\) and \(\lambda_{emp} [6]\), where \(\bar{G}_k\) is set as 70% of \(G_k\) when \(\lambda_i \geq \lambda_{emp}\), and 30% of \(G_k\) when \(\lambda_i < \lambda_{emp}\).

We notice that the metro tunnel is a highly structured scenario with many line features, for instance, rail tracks, power lines, and pipelines. Such line features have proved to be a promising approach towards man made structured scenarios. Since the rail tracks are the most distinctive line structures in the point cloud, they are first extracted based on the geometric patterns.

We first determine the track bed area using the LiDAR sensor mounting height. The track bed is composed of rail sleepers, rail tracks, and track-side sensors, where the rail tracks are the highest infrastructures. With the assumption of the LiDAR is centered between two rail tracks, we can set two candidate areas around the left and right rail tracks and search the points above a certain threshold over the track bed. Two straight lines can then be fixed using RANSAC method. Finally, we exploit the idea of region growing for further refinement. As a prevailing segmentation algorithm, region growing examines neighboring points of initial seed area and decides whether to add the point to the seed region or not. We set the initial seed area within the distance of 3 m ahead of the LiDAR, and the distance threshold of the search region to the fitted line is set to 0.07 m, which is the width of the track head.

Note that we only extract the current two tracks where the metro vehicle is on, and a maximum length of 15 m tracks are selected for each frame.

The other line features are extracted following the depth-continuous edge extraction method proposed in [32]. The point cloud is divided into small voxels for each input frame, and planes are fitted repeatedly in the voxels. Then the depth-continuous edges can be detected at the plane intersections. We sample multiple points on each extracted line including the two tracks, and retain the lines that satisfy the infinite straight line model. The extracted lines are shown in Fig. 6 thanks to the high point cloud density coverage.

The infinite straight lines can be parameterized by a rotation matrix \(R\) and two scalars \(a, b \in \mathbb{R}\). Let the error operator \(\Theta\) between two lines \(L_i, L_j\) be defined as:

\[
L_i \Theta L_j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}^T \log \left( R_i^T R_j \right) = \begin{bmatrix} a_i \ -a_j \\ b_i \ -b_j \end{bmatrix},
\]

(10)

Then the residual between the two lines can be formulated by:

\[
e^{LIO_{ij \rightarrow iL}} = \left( T_{L_i}^T L_i \right) \Theta L_j,
\]

(11)

where \(T_{L_i}^T L_i = (R_i^T, p_i^T)\) describes the transformation between the two lines. We use the derivatives of equation (11) in the optimization process with symmetric difference method.

Additionally, a fitted plane \(P\) can be parameterized by the unit normal direction vector \(n_p\) and a distance scalar \(d_p\), such that \(P = [n_p^T, d_p]^T\). The measurement residual from a point to a plane can be expressed as:

\[
e^{LIO_{po \leftarrow pl}} = T_{p_i}^T p_i.
\]

(12)

Suppose we have \(N_{p, p}\) sets of point-to-plane correspondences and \(N_p\) points, the plane-to-plane residual can be
formulated by:

$$e^{LIO}_{pl \rightarrow pl} = \min \sum_{i=1}^{N_{pl}} \sum_{j=1}^{N_{pl}} e_{ij}^{LIO}_{pl \rightarrow pl}. \quad (13)$$

We use the second-order derivatives of point-to-plane cost for efficient optimization as illustrated in [33]. In addition, we use the IMU forward propagation to predict the location of the previous lines and planes in the current scan. Two lines are considered a match if their directions and center distances divergences are lower than 10° and 0.5 m. Similarly, two planes are regarded as a match when difference between their normal and the distance scalar are smaller than 5° and 0.25 m.

Given equation (7), (11) and (13), the minimization problem of LIO can be expressed as follows:

$$\min \left\{ \sum_{i \in N_L} w_i \epsilon_i^{LIO}_{li \rightarrow li} + \sum_{i \in N_P} w_i \epsilon_i^{LIO}_{pl \rightarrow pl} + \sum_{i \in N_{I}} \epsilon_i^{IMU} + \sum_{i \in N_S} w_i \epsilon_i^{LIO}_{sc \rightarrow sc} + e_i^{LIO} + \epsilon_{prior}^{imuodom} \right\} \quad (14)$$

where $N_L$, $N_P$, $N_I$, and $N_S$ are the number of line-to-line, plane-to-plane correspondences, IMU preintegration factors, and scan-to-scan feature points correspondences. $e_i^{LIO}$ is the marginalization factors and $\epsilon_{prior}^{imuodom}$ is the predicted pose constraints from IMU odometry. The optimization problem can be efficiently solved by Levenberg-Marquardt algorithm [34].

C. VIO Factors

Our VIO follows the pipeline of Vins-mono [35] as shown in Fig. 2. The point features are detected by [36], tracked by KLT sparse optical flow, and refined by RANSAC. For the line features, we employ the LSD [38] for line segment extraction as shown in Fig. 7(a). In addition, we register LiDAR frames to the camera frames, and project the 3D distortion-free point cloud to the 2D image as illustrated in Fig. 7(b). Then we can use the depth information for scale correction and joint graph optimization following [26]. The minimization problem of VIO can be expressed as follows:

$$\min \left\{ \sum_{i \in N_{li}} \epsilon_i^{VIO}_{li \rightarrow li} + \sum_{i \in N_{po}} \epsilon_i^{VIO}_{po \rightarrow po} + \sum_{i \in N_{I}} \epsilon_i^{MU} + \right\} \quad (15)$$

where $\epsilon_i^{VIO}_{li \rightarrow li}$, $\epsilon_i^{VIO}_{po \rightarrow po}$, and $\epsilon_i^{MU}$ are the relative line and point projection error, and marginalization factors. $N_{li}$ and $N_{po}$ represent the amount of line and point features tracked by other frames.

D. Map Association

We maintain local planar and edge points as well as line feature maps as described in LOAM. The incremental registration of new scan to local map will lead to accumulate and non-reversible errors, where current errors will propagate to following registrations. Therefore, we implement the bundle adjustment (BA) at our backend to refine LiDAR mapping. The map-refinement performs a local BA on a sliding window of lidar poses. This is done by minimizing the second order approximation of the total cost consisting of all relevant voxels. The refined poses are then used to update the center points and normal vectors of all involved voxels.

Considering the high point cloud density within each scan, incorporating a new frame to existing ones will be inefficient and time-consuming. We hereby leverage a dynamic kd-tree structure, ikd-tree [39], in our system, enabling incremental update of the map by updating the new points only.

V. Experiments

A. Hardware Setup and Ground Truth Description

We select a metro maintenance vehicle as shown in Fig. 8, which includes a hybrid solid-state LiDAR Innovusion Jaguar Prime, a tactical grade IMU Sensoror stim 300, and a Hikvision supervisory camera. All the three sensors are hardware synchronized with an external atomic clock. The dataset is processed by a high-performance onboard computer, with i9-10980HK CPU, 64 GB RAM. According to the safety regulations on the railroad, we are only able to conduct experiments in the midnight. Our algorithms are implemented in C++ and executed in Ubuntu Linux using the ROS. Since the output of the Hikvision camera is in the video format, we first merge the video with the recorded dataset using the synchronized timestamp. The nonlinear optimization and factor graph optimization problem is solved using GTSAM, respectively. Our proposed system can reach real-time performance for all the captured datasets.

The odometry and map ground truth is captured by two survey grade 3D laser scanners, a navigation grade IMU containing fiber optic gyros, and processed by professional post-processing software. The initial global position is set by Real Time Kinematic (RTK) at the starting point outdoors.

We conduct extensive experiments utilizing the maintenance vehicle platform, and two novel LiDAR-visual-inertial fusion algorithms are selected for comparison: LVI-SAM [24] and...
R2LIVE [21]. In addition, LOAM [30], Lio-sam [40], and VINS-Mono [35] are also selected for ablation study here. The LIO and VIO part of our system is denoted as MetroLIO and MetroVIO, respectively.

B. Experiment-1: Outdoor-Only

We manually drive the maintenance rail vehicle leaving the maintenance station and entering the underground tunnel in this experiment. The vehicle is moving with a high speed towards the tunnel (maximum 70 km/h) in this scene, and the travelled distance is around 2.52 km. Due to the bad illumination conditions, the VIO is unable to provide useful pose estimations in this experiment. Therefore, LVI-SAM and R2LIVE is mainly dependent on its LIO subsystem in this case, namely, Lio-sam and Fast-lio2 [8]. We deactivate the loop detection module in Lio-sam and LVI-SAM since the structure patterns in the scene are repetitive, which may lead to wrongly detected loops. Note that the GNSS information in not included for state estimation.

We first present the quantitative results w.r.t. ground truth. As mentioned in Section III-B, both the ground truth results and real-time odometry share the same timestamp, and we perform the per-frame accuracy evaluation through time alignment. To ensure no systematic errors, we set the start point of each point sharing the same coordinates with post processing results. The trajectories of various approaches are plotted in Fig. 9.

As mentioned in our previous work [42], different from the spinning LiDARs, the LIO of small FoV LiDARs are prone to suffer from rotational errors. It is seen the loosely-coupled LOAM has large horizontal errors mainly due to lack of surround feature constraint. Besides, the incremental manner of map construction accelerates this error accumulation. Lio-sam and LVI-SAM has a similar performance as expected. This is because the LIO and VIO subsystem of LVI-SAM works independently. Once sensor failure (no data, insufficient input, etc.) is detected in one module, the system will automatically switch to the other module instead. Similarly, R2LIVE relies on the Fast-lio2 for state estimation, and it has a much higher accuracy than the above two methods. This benefits twofold: the first is the direct registration of the raw point clouds without feature extraction; the second is the usage of larger local map in scan registration process whereas the former two methods only employ a certain number of keyframes. However, as a filter-based algorithm, R2LIVE is sensitive to wrong inertial measurements. The inertial reading will see large vibrations at the joint of rail tracks and the rail track turnouts, and Fast-lio2 may generate some vertical errors at such districts as visualized in Fig. 10. In our approach, this error can be corrected by the plane constraints, and our trajectory is approximal to the ground truth. Besides, we plot the 6 degree-of-freedom (DoF) errors of our system in Fig. 11. We can infer that the 6 DoF errors all remain at low level at this stage. It is seen that the vertical error has a large jump in the beginning. This is because the maintenance vehicle started at a turning where the two rail tracks are not of the same height, and the virtual track plane is not precise to constrain the vertical displacements. When the track heights are the same for the following lines, it is seen the error curve is almost flat.

Although the FoV is small, the Jaguar Prime generates as many points as that of a 64-channel spinning LiDAR, causing great burden to the feature tracking process. In practices, the LiDAR odometry part of LOAM have exceed 100 ms, denoting the whole system unable of real-time operation. Without proper feature selection or efficient data association methods, Lio-sam and LVI-SAM also cannot reach real-time performance. Although R2LIVE leverage all the points for scan registration, the efficient map representation of ikd-tree makes the R2LIVE capable of real-time operation. Our system also benefits from the ikd-tree and can run up to three times faster than data gathering speed.

We then show some qualitative mapping results. The global mapping result is shown in Fig. 12, in which the height variation is not enormous, demonstrating our method is of
Fig. 12. The overall mapping result of experiment-1, with the two insets showing details of related areas. The color of global mapping is coded by height variations, whereas the color of two insets indicate the initial reflectivity.

Fig. 13. Visualization of different trajectories of experiment-2.

C. Experiment-2: Underground and Outdoor

In this experiment, the maintenance vehicle finishes its daily task and returns to the station. The journey is around 1.93 km but with a smaller velocity, the time consumption is 212 s. Half of the journey is inside the underground tunnel, while the other half is outdoor. Since the outdoor environment is dark similar with experiment-1, the visual approaches cannot work in such district. We hereby follow the setup of the former experiment for evaluation. The trajectories of various methods are plotted in Fig. 13.

It is seen that LOAM and R2LIVE fails inside the tunnel. The reason for the former failure is due to lack of degeneracy analysis module. As a loosely-coupled system, LOAM merely depend on the scan registration module for pose estimation. Once the environment is repetitive as visualized in Fig. 14(a), the scan-to-scan correspondences is hard to track, and the real-time odometry will ‘stop’ in the tunnel as shown in Fig. 14(b).

As for the latter failure, this is mainly because the LIO part of R2LIVE, Fast-lio2, is not robust to the long-during backward motion of the rail vehicle. As a result, large errors will appear in all 6 DoF. On the contrary, both the loosely-coupled system and optimization-based tightly-coupled scheme is immune to this influence. With the assistance of geometric regularities on the railroad, our system can achieve consistent and accurate pose estimation results.

To further reveal the contribution of such constraints, we define MetroLoc w/o GR and MetroLoc w/ GR as our system without and with geometric constraints. The absolute trajectory error (ATE) of two methods are plotted in Fig. 15. We can directly infer that the usage of typical geometric pattern improves 6 DoF accuracy significantly. As illustrated above, the small FoV makes the LIO sensitive to vibrations on the rail tracks. It is seen that the attitude errors in roll direction has a continuous growth when ground constraint is not available, and the maximum error is ten times larger than MetroLoc w/ GR. Besides, the vertical errors also accumulate quickly when ground constraint is not available.
The overall mapping result is plotted in Fig. 16, in which the flat and consistent tunnel map denoting our system is robust to such degeneracy problems. Note that the rail tracks and trackside infrastructures cannot be manually distinguished due to the imprecise reflectivity outputs.

D. Experiment-3: Underground-Only

In this experiment, the maintenance rail vehicle is inside the tunnel along the path, and the metro staff is carrying out manual inspection of trackside infrastructures. We divide this scene into three different datasets for evaluation. In contrast with the former two experiments, the VIO can provide meaningful results in this scenario due to sufficient illuminations.

Scenario-1: Feature-Rich Crossover and Station. Depicted in Fig. 17(a) and Fig. 17(b), the metro crossover and station are feature-rich areas with multiple rail tracks, guardrails, pipelines, and platforms. Besides, the visual sequences are not degraded because of the head light and station lighting.

Scenario-2: Structured Tunnel. In this experiment, we aim to evaluate the robustness of our algorithm under a 750 m metro tunnel scenario, where the only observable features are repetitive and structured man-made infrastructures as shown in Fig. 18(a) and Fig. 18(b). Since the tunnel walls are almost flat and no extra dynamic objects can be employed for feature matching, the metro tunnel is one of the most challenging scenarios for SLAM. The state estimation errors of different
methods are listed in TABLE II. Due to the highly regulated and constrained motion on the rail tracks, the IMU biases cannot be initialized and estimated correctly with inadequate axis excitation. This leads to serious scale drift and result in significant pose estimation errors in VINS-Mono. Besides, the wrong feature matches from repetitive pattern of the surround also exacerbate the deviations.

The feature-based LiDAR SLAM also suffers from feature tracking problems. With only frame-to-frame matching, LOAM fails several times in the tunnel, and the LiDAR odometry even moves backwards without accurate pose estimation. Since Lio-sam relies heavily on the LiDAR odometry to further constrain the pre-integrated IMU states, it also generates unfavorable results. On the contrary, both the MetroLIO and MetroVIO achieves less drift state estimation employing typical geometrical regularities and local BA.

**Scenario-3: Long-During Mapping and Odometry.** In this scenario, we aim to show that MetroLoc is accurate and robust enough to reconstruct a long-during map in metro environments. The overall length is around 5.2 km with 4 stations along the path. In addition, the real-time reconstruction video can be founded on the website.²

We plot the real-time reconstructed 3D maps as well as the ground truth in Fig. 19, where both the horizontal headings and height variations are visually well matched. Besides, the detailed mapping comparison with high precision instruments are visualized in Fig. 1, demonstrating that our mapping is of high precision locally. Note that there are many outlier points around tunnel structures in the reconstructed map, we believe they are mainly caused by the LiDAR range measurement noises. In addition, we adopt the EVO³ package to compare each trajectory against ground truth as shown in Fig. 20. We can infer that the monocular visual-inertial methods have the worst performance in this situation, as explained above. Our MetroLoc outperforms the selected methods with the lowest ATE of 5.62 m.

### E. Runtime Analysis

Thanks to the multi-threaded computation and adaptive voxelization method, all the subsystems can run in parallel efficiently. The average time consumption of each submodule is listed in TABLE III, illustrating that MetroLoc can achieve real-time performance for the onboard computer.

Besides, we also list the average and maximum time cost of different methods in TABLE IV. With increased data gathering period, the size of maintained factor graph of LVI-SAM and Lio-sam will grow continuously, and the optimization time will increase. On the contrary, our proposed method can keep a relatively low computation cost with sliding window design.

2³https://www.youtube.com/watch?v=cSGnnmULvMY&ab_channel=YUSHENGWANG

2³https://github.com/MichaelGrupp/evo
TABLE III

| Data       | MetroLO | MetroVIO | IMUODOM |
|------------|---------|----------|---------|
| Experiment-1 | 38      | 24       | 0.5     |
| Experiment-2 | 34      | 23       | 0.4     |
| Scenario-1  | 37      | 21       | 0.5     |
| Scenario-2  | 32      | 19       | 0.5     |
| Scenario-3  | 31      | 23       | 0.5     |

TABLE IV

| Data       | Experiment-1 | Experiment-2 | Scenario-1 | Scenario-2 | Scenario-3 |
|------------|--------------|--------------|------------|------------|------------|
| LVI-SAM    | 84/285       | 91/274       | 62/214     | 57/184     | 61/174     |
| R2LIVE     | 61/204       | 54/192       | 49/196     | 48/187     | 51/209     |
| LOAM       | 86/97        | 87/102       | 84/94      | 104/221    | 91/154     |
| Lio-sam    | 72/235       | 81/227       | 43/96      | 44/102     | 63/197     |
| Vins-mono  | 48/92        | 39/73        | 38/88      | 32/77      | 31/80      |
| MetroLoc   | 62/103       | 57/112       | 58/99      | 51/90      | 54/92      |

VI. CONCLUSION

In this paper, we propose a robust and versatile simultaneous localization and mapping approach for metro vehicles. Our method tightly integrates measurements from LiDAR, camera, and IMU with an IMU-centric state estimator. To cope with the highly repeated man-made structures, we propose to extract and track geometric features. The proposed method has been validated in extreme metro environments. The results show that our framework is accurate and robust towards long-duration motion estimation and mapping problems. We hope that our experimental work and extensive evaluation could inspire follow-up works to explore more structural or situational awareness information in highly-repetitive scenarios, such as tags or signs on the wall as shown in Fig. 22. Besides, we wish the community pay more attention to railroad applications, especially for facility and environment monitoring. A small advance towards autonomous system will save tremendous amount of manpower for construction and maintenance.

REFERENCES

[1] S. Guo, Z. Rong, S. Wang, and Y. Wu, “A LiDAR SLAM with PCA-based feature extraction and two-stage matching,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–11, 2022.
[2] G. He, X. Yuan, Y. Zhuang, and H. Hu, “An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–9, 2021.
[3] T. Du, S. Shi, Y. Zeng, J. Yang, and L. Guo, “An integrated INS/LiDAR odometry/polarized camera pose estimation via factor graph optimization for sparse environment,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–11, 2022.
[4] J. Otegui, A. Bahillo, I. Lopetegi, and L. E. Díez, “Evaluation of experimental GNSS and 10-DOF MEMS IMU measurements for train positionning,” IEEE Trans. Instrum. Meas., vol. 68, no. 1, pp. 269–279, Jan. 2019.
[5] F. Tschopp et al., “Experimental comparison of visual-aided odometry methods for rail vehicles,” IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 1815–1822, Apr. 2019.
[6] J. Zhang, M. Kaess, and S. Singh, “On degeneracy of optimization-based state estimation problems,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2016, pp. 809–816.
[7] A. Tagliabue et al., “LION: Lidar-Inertial observability-aware navigator for vision-denied environments,” in Proc. Int. Symp. Exp. Inert. Robots, Malta: Springer, 2020, pp. 380–390.
[8] A. Hinduja, B.-J. Ho, and M. Kaess, “Degeneracy-aware factors with applications to underwater SLAM,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2019, pp. 1293–1299.
[9] K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A.-A. Agha-Mohammadi, “DARE-SLAM: Degeneracy-aware and resilient loop closing in perceptually-degraded environments,” 2021, arXiv:2102.05117.
[10] W. Zhen and S. Scherer, “Estimating the localizability in tunnel-like environments using LiDAR and UWB,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 4903–4908.
[11] A. Santamaria-Navarro, R. Thakker, D. F. Fan, B. Morrell, and A. Agha-Mohammadi, “Towards resilient autonomous navigation of drones,” in Proc. Int. Symp. Robot. Res. Geneva, Switzerland: Springer, 2019, pp. 922–937.
[12] J. Jiao et al., “Greedy-based feature selection for efficient LiDAR SLAM,” 2021, arXiv:2103.13090.
[13] L. Zhou, D. Koppel, and M. Kaess, “LiDAR SLAM with plane adjustment for indoor environment,” IEEE Robot. Autom. Lett., vol. 6, no. 4, pp. 7073–7080, Oct. 2021.
[14] L. Zhou, S. Wang, and M. Kaess, “IL-SLAM: LiDAR smoothing and mapping with planes,” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), 2021, pp. 5751–5757.
[15] J. Wen, X. Zhang, H. Gao, J. Yuan, and Y. Fang, “CAE-RLSM: Consistent and efficient redundant line segment merging for online feature map building,” IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 4222–4237, Jul. 2020.
[16] J. H. Jung, S. Heo, and C. G. Park, “Observability analysis of IMU intrinsic parameters in stereo visual-inertial odometry,” IEEE Trans. Instrum. Meas., vol. 69, no. 10, pp. 7530–7541, Oct. 2020.

[17] M. A. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.

[18] 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, Dec. 2018.

[19] W. Wang, J. Liu, C. Wang, B. Luo, and C. Zhang, “DV-LOAM: Direct visual LiDAR odometry and mapping,” Remote Sens., vol. 13, no. 16, p. 3340, 2021.

[20] H. Yang, J. Shi, and L. Carbone, “TEASER: Fast and certifiable point cloud registration,” IEEE Trans. Robot., vol. 37, no. 2, pp. 314–333, Apr. 2020.

[21] J. Lin, C. Zheng, W. Xu, and F. Zhang, “R2LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping,” 2021, arXiv:2107.06829.

[22] X. Zuo et al., “LiFusion 2.0: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking,” in Proc. IEEE/RSJ Int. Conf. Robot. Syst. (IROS), Nov. 2019, pp. 5848–5854.

[23] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2020, pp. 3126–3131.

[24] J. Zhang and S. Singh, “LOAM: LiDAR odometry and mapping in real-time,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1004–1011, Apr. 2021.

[25] 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 Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2020, pp. 1091–1097.

[26] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, “Super odometry: IMU-centric LiDAR-inertial-visual-inertial estimator for challenging environments,” 2021, arXiv:2104.10851.

[27] D. Wisth, M. Camurri, S. Das, and M. Fallon, “Unified multi-modal landmark tracking for tightly coupled LiDAR-visual-inertial odometry,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1004–1011, Apr. 2021.

[28] X. Zuo et al., “LiFusion: LiDAR-inertial-camera odometry with sliding-window plane-feature tracking,” in Proc. IEEE/RSJ Int. Conf. Robot. Syst. (IROS), Oct. 2020, pp. 5112–5119.

[29] J. Lin et al., “LOAM: LiDAR-inertial-visual tightly-coupled state estimator and mapping,” 2021, arXiv:2102.12400.

[30] T. Qin, P. Li, and S. Shen, “VINS-mono: A robust and versatile monocular visual-inertial state estimator,” in Proc. IEEE/RSJ Int. Conf. Robot. Syst. (IROS), Oct. 2020, pp. 5135–5142.

[31] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, “LiDAR-inertial-camera odometry,” in Proc. IEEE/RSJ Int. Conf. Robot. Syst. (IROS), Oct. 2020, pp. 5112–5119.

[32] J. Jiao et al., “Greedy-based feature selection for efficient lidar slam,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 1004–1011, Apr. 2021.

[33] J. Shi, “Good features to track,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 1994, pp. 593–600.

[34] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proc. IJACAI, Vancouver, BC, Canada, 1981, pp. 1–11.

[35] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, “LSD: A fast line segment detector with a false detection control,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 4, pp. 722–728, Apr. 2010.

[36] Y. Cai, W. Xu, and F. Zhang, “lkdtree: An incremental K-D tree for robotic applications,” 2021, arXiv:2102.10808.

[37] R. Shah, 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. Robot. Syst. (IROS), Oct. 2020, pp. 5135–5142.

[38] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast direct LiDAR-inertial odometry,” 2021, arXiv:2107.06829.

[39] Y. Wang, Y. Lou, W. Song, and Z. Tu, “A tightly-coupled framework for large-scale map construction with multiple non-repetitive scanning LiDARs,” IEEE Sensors J., vol. 22, no. 4, pp. 3626–3636, Feb. 2022.

[40] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in Field and Service Robotics. Tokyo, Japan: Springer, 2018, pp. 621–635.