Multi-Modal Features and Accurate Place Recognition with Robust Optimization for Lidar-Visual-Inertial SLAM

Xiongwei Zhao, *Student Member, IEEE*, Congcong Wen, *Member, IEEE*, Sai Manoj Prakhya, Hongpei Yin, Rundong Zhou, Yijiao Sun, Jie Xu, Haojie Bai and Yang Wang

***Abstract*—Lidar-Visual-Inertial SLAM (LVINS) provides a compelling solution for accurate and robust state estimation and mapping, integrating complementary information from multi-sensor data. However, in the front-end processing of existing LVINS systems, methods based on visual line feature matching typically suffer from low accuracy and are time-consuming. Additionally, the back-end optimization of current multi-sensor fusion SLAM systems is adversely affected by feature association outliers, which constrains further enhancements in localization precision. In the loop closure process, existing lidar loop closure descriptors, relying primarily on 2D information from point clouds, often fall short in complex environments. To effectively tackle these challenges, we introduce the Multi-Modal Feature-based Lidar-Visual-Inertial SLAM framework, abbreviated as MMF-LVINS. Our framework consists of three major innovations. Firstly, we propose a novel coarse-to-fine visual line matching method that utilizes geometric descriptor similarity and optical flow verification, substantially improving both efficiency and accuracy of line feature matching. Secondly, we present a robust iterative optimization approach featuring a newly proposed adaptive loss function. This function is tailored based on the quality of feature association and incorporates graduated non-convexity, thereby reducing the impact of outliers on system accuracy. Thirdly, to augment the precision of lidar-based loop closure detection, we introduce an innovative 3D lidar descriptor that captures spatial, height, and intensity information from the point cloud. We also propose a two-stage place recognition module that synergistically combines both visual and this new lidar descriptor, significantly diminishing cumulative drift. Extensive experimental evaluations on six real-world datasets, including EuRoc, KITTI, NCLT, M2DGR, UrbanNav and UrbanLoco, demonstrate that our MMF-LVINS system achieves superior state estimation accuracy compared to existing state-of-the-art methods. These experiments also validate the effectiveness of our advanced techniques in visual line matching, robust iterative optimization, and enhanced lidar loop closure detection.**

***Index Terms*—Lidar-Visual-Inertial SLAM, State Estimation, Robust Iterative Optimization, 3D Lidar Loop Closure Descriptor, Two-Stage Loop Detection**

I. Introduction

In last two decades, simultaneous localization and mapping (SLAM) has been one of the most active areas of research in the field of robotics. SLAM is the key enabling technology behind self-driving cars, autonomous mobile robots, be it, humanoid like legged robots, wheeled robots or underwater robots, indoor robots and even augmented reality [1]–[3]. Essentially, using sensor data, SLAM enables a robot to localize itself in an unknown environment, taking its initial position as reference, and during this process, it also creates map, a coherent representation of the environment. At this point of time, SLAM systems for outdoor mobile robotics can mostly be categorized into three major classes by the type of sensors: visual SLAM, lidar SLAM and multi-sensor fusion SLAM [4], [5]. Multi-sensor fusion SLAM offers greater robustness and accuracy in complex scenarios compared to single-sensor SLAM systems [6]–[8].

Incorporating a broader range of visual features, such as line features, has been shown to significantly enhance the accuracy of visual localization [9]. However, existing line feature matching methods based on feature descriptors are notably time-consuming [10]. Additionally, relying exclusively on optical flow for line feature matching introduces high complexity and often lacks the desired accuracy [11]. Therefore, optimizing the balance between accuracy and efficiency in line feature association is critical.

In the backend of the SLAM system, feature associations are formulated as a least squares problem to optimize and achieve an optimal system state [12]. A major challenge here is that the Bundle Adjustment (BA) in contemporary SLAM systems is highly susceptible to outliers from feature matching. These outliers are frequently encountered in complex scenarios involving dynamic objects or feature mismatches [13]–[15]. The conventional optimization techniques employing standard loss functions, such as Huber or Cauchy, are less effective in rejecting outliers in these complex scenarios, ultimately diminishing system accuracy [16]–[18].

Moreover, loop closure detection plays a pivotal role in correcting cumulative drift [19]. Current lidar loop closure

This work was supported in part by Science and Technology Project of Shenzhen under Grant JCYJ20200109113424990; in part by Marine Economy Development Project of Guangdong Province under Grant GDNRC [2020]014. (Corresponding author: Yang Wang)

Xiongwei Zhao, Yijiao Sun, Haojie Bai, and Yang Wang are with the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518071, China (e-mail: xwzhao@stu.hit.edu.cn, sunyijiao@stu.hit.edu.cn, hjbai@stu.hit.edu.cn, yangw@hit.edu.cn).

Congcong Wen is with NYU Tandon School of Engineering, New York University, USA (e-mail: wencc@nyu.edu).

Sai Manoj Prakhya is with Huawei Munich Research Center, Germany (e-mail: sai.manoj.prakhya@huawei.com).

Hongpei Yin is with Guangdong Institute of Artificial Intelligence and Advanced Computing, China (e-mail: wyinhongpei@163.com).

Rundong Zhou is with the School of Electronic and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: 19b305001@hit.edu.cn).

Jie Xu is with the School of Mechanical and Electrical Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: jeff_xu@hit.edu.cn).

descriptors, like FreSco [20] and ISC [30], create 2D representations by projecting 3D points into 2D grids and use the maximum height or intensity information in each grid for lidar frame similarity detection. This reliance often leads to less than optimal detection accuracy, underscoring the need for more sophisticated loop closure detection methods.

To address the aforementioned issues, we propose MMF-LVINS, a tightly-coupled multi-modal feature-based lidar-visual-inertial SLAM framework. In the frontend feature processing module, we extract multimodal features: visual point and line features, along with lidar edges and surf features. For visual line features, we design a novel geometric descriptor based on their geometric properties and utilize the Heaviside function [22] for detecting similarities. Additionally, we propose a new visual line matching method, employing geometric descriptor similarity and optical flow verification for enhanced accuracy. In the backend optimization module, inspired by graduated non-convexity [23] and Black-Rangarajan duality [24], we propose a robust iterative optimization approach, introducing a novel adaptive loss function based on the quality and the residuals of feature matching. This approach effectively suppresses the impact of feature matching outliers on the optimization process, thus enhancing the precision and convergence speed of backend optimization. In the loop closure module, a novel 3D lidar loop closure descriptor is designed by encoding spatial, height, and intensity information of point clouds and devising an associated weighting function. Furthermore, we employ a two-stage loop closure module that combines both visual and the proposed lidar descriptors to corrects the accumulated drift and building a globally consistent map.

In summary, we propose a Multi-Modal Feature-based Lidar-Visual-Inertial SLAM framework, enhancing the existing LVINS framework in three aspects: front-end, back-end, and loop closure detection. The main contributions of this work can be summarized as follows:

- A novel coarse-to-fine visual line segment matching method is introduced, leveraging geometric descriptor similarity and optical flow verification. This method significantly improves the line matching ratio while also reducing computational time, thereby optimizing both efficiency and accuracy.
- We present a robust iterative optimization technique incorporating a novel adaptive loss function based on the quality of feature association and graduated non-convexity to mitigate the impact of outliers on system accuracy.
- We design a novel global 3D lidar descriptor that encodes spatial, height, and intensity information, enhancing lidar loop closure detection. Building upon this, we incorporate visual loop closure detection, forming a two-stage robust place recognition module.
- An open-source implementation that's extensively validated in terms of visual line matching, loop closure detection, iterative optimization approach and complete system's localization accuracy on multiple real-world datasets at ***https://github.com/Grandzxw/MMF-LVINS***.

The remainder of this paper is organized as follows. In Section II, we review related work on SLAM and Lidar Loop Detection. The methodology of the proposed SLAM framework is presented in Section III. Section VII showcases both the quantitative and qualitative results of the proposed framework. Finally, the conclusions and future work are presented in Section VIII.

II. RELATED WORK

In this section, we review existing works related to our system, including Multi-Feature Visual SLAM, Lidar Loop Detection, and Lidar-Visual Inertial SLAM. We select the most representative works in each domain for review.

A. Multi-Feature Visual SLAM

To achieve better accuracy, many researchers have added additional feature constraints into the SLAM system. PLP-SLAM [25] leverages both points and lines for visual localization, and simultaneously performs a piece-wise planar reconstruction for structured environments. UV-SLAM [26] utilizes the vanishing points directional constraints obtained from the line features to enhance performance in structured environments. The aforementioned methods extract line features using LSD [10] and match them using LBD [27]. This method of matching line features is not only time-consuming but also has low matching accuracy, which affects the real-time performance of the system. To improve real-time performance, PLC-VIO [11] employs optical flow tracking to match point features located near the pixels associated with line features, achieving lines matching between two images, which determines whether two line segments match by judging the distance from the tracked point feature to the line feature, the computational complexity is $O\left(N^2\right)$. The optimal line feature matching is determined by jointly utilizing LBD and optical flow in [28]. Although the aforementioned methods employ optical flow, their computational complexity is still high.

B. Lidar Loop Detection

In recent years, lidar loop closure detection has garnered increasing attention from researchers. Current lidar loop closure descriptors mainly include handcrafted descriptors and learning-based descriptors [29]. Scancontext [21] is the most popular lidar handcrafted descriptor. It first divides the raw point cloud data into discrete bins on the Bird's Eye View (BEV) using polar coordinates. Subsequently, the maximum height of each bin is encoded into the descriptor. Intensity Scan Context [30] incorporates intensity information of the point cloud into the descriptor encoding based on [21]. lidar-Iris [31] generates the binary signature images image representation to detect potential loops. NDD [19] utilizes the spatial distribution of point clouds in Bird's Eye View to detect similar loop frames. The aforementioned handcrafted descriptors reduce 3D point cloud information to 2D, and specifically, Scancontext [21] only takes the maximum height in a spatial bin as the representative value, leading to severe loss in information and inability to effectively detect similarities in complex scenarios.

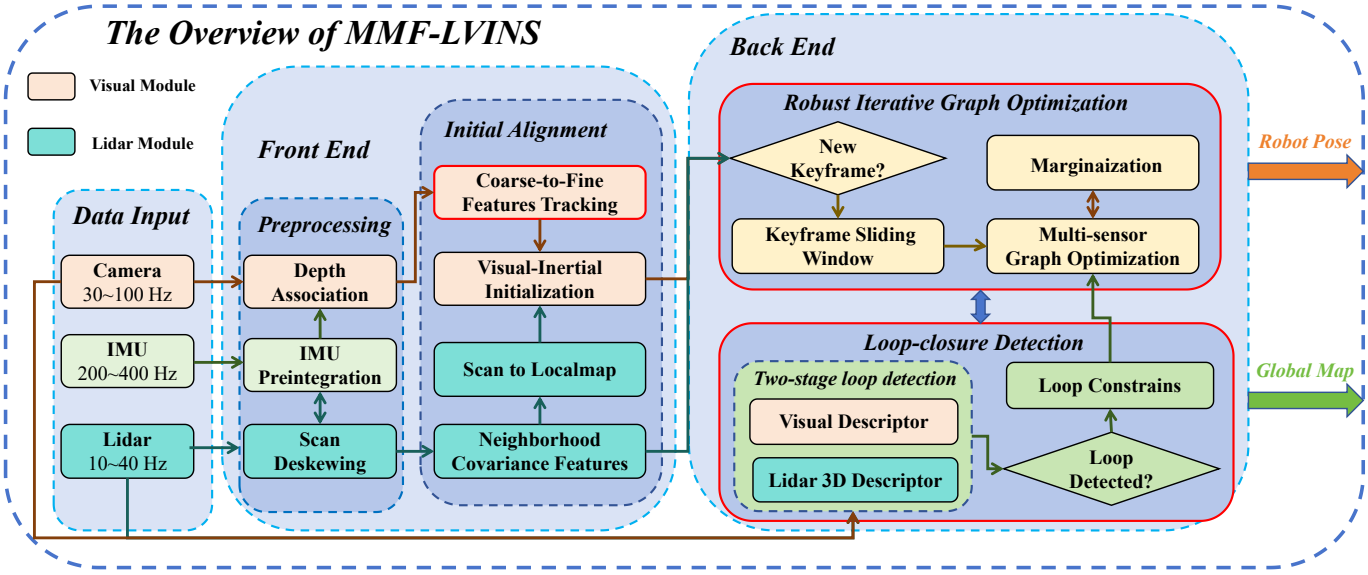


Figure 1: Overview of the proposed MMF-LVINS system. The system mainly includes two parts: the frontend and the backend. Four main processing modules are introduced to construct the backbone of the algorithm. (a) Preprocessing, (b) Initial Alignment, (c) Robust Iterative Graph Optimization, (d) Loop-clousre detection. These modules are described in Sections III. The content corresponding to the red border is our main innovative work.

Learning-based descriptors are also gaining increasing attention. PointNetVLAD [32] leverages PointNet [33] and NetVLAD [34] to encode global descriptors through end-to-end training. OverlapNet [35] adopts a siamese network estimates the similarity between pairs of scans. However, the biggest shortcoming of the learning-based descriptor is that they are poor at generalization [36]. In addition, these learning-based descriptors require a lot of training data.

C. Lidar-Visual Inertial SLAM

Effectively fusing measurements from lidar, camera, and IMU is the most favourable option for localization and mapping. Zhang *et al* propose a lidar-inertial-visual system that uses a loosely-coupled VINS as the motion constrain to initialize the lidar odometry subsystem [37]. Similarly, Shao *et al*. propose a stereo-inertial-lidar SLAM that incorporates the tightly-coupled stereo visual-inertial odometry with lidar mapping and lidar-enhanced visual loop detection [38].

While previous works used a loosely-coupled fusion of lidar and camera measurements, there are several tightly-coupled sensor data fusion frameworks proposed to enhance accuracy and robustness. Shan *et al*. propose LVI-SAM [6] that fuses the lidar, visual and inertial sensors in a tightly-coupled smooth and mapping framework. A similar tightly-coupled system is R2LIVE [7], which fuses the lidar and camera measurements in an on-manifold iterated kalman filter framework. R3LIVE [8] uses direct methods in both LINS and VINS to exploitany subtle features in the environments, whose accuracy is easily affected by changes in scene lighting conditions. The above lidar-inertial-visual systems don't take full advantage of the feature information in the environment, and in the aforementioned graph-based optimization systems, the impact of feature association outliers on backend optimization is not considered.

III. METHODOLOGY

A. The System Overview

To simultaneously estimate the robot's pose and reconstruct the 3D map of the environment, we design a tightly-coupled lidar-visual-inertial sensor fusion framework, as shown in Fig.1. The system consists of two primary components: the frontend and the backend. The frontend comprises two processing modules: data preprocessing and initial alignment, while the backend includes robust graph optimization module and loop closure module. In the data preprocessing module, we use IMU preintegration measurements to deskew lidar scans, and camera images to associate feature depths with the lidar scan. In the initial alignment module, we extract multi-modal features from both lidar and vision. For the lidar part, we utilized neighborhood covariance [39] to extract its edge and surf features. For the visual part, we extract point and line features, and propose a novel visual line feature tracking method. By integrating the geometric descriptors of line segments with optical flow tracking of sampled points on the line, we achieved efficient and robust line segment matching. Furthermore, similar to [6], the scan to localmap module of lidar-inertial odometry (LINS) module provides the 6Dof pose to expedite the initialization of visual-inertial odometry (VINS) module and the LINS module provides accurate depth information for the features extracted by the VINS module.

In the backend, we address system state estimation through the robust iterative optimization, which optimizes the contributions of IMU factor constraints, visual landmarks constraints, lidar landmarks constraints, and loop closure constraints in the factor graph. Additionally, we utilize both the image data and the 3D spatial information from the point cloud for

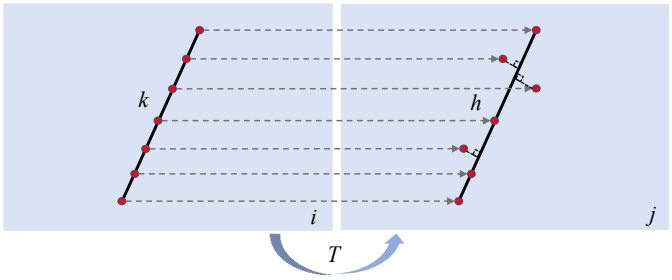


Figure 2: Illustration of lines fine matching phase bewteen image i and image j. T is the pose transformation from image i to image j. The red points denote adaptively extracted points, while the dashed lines represent the optical flow prediction relationships

loop closure detection, which further enhances the system's accuracy and helps establish a globally consistent map.

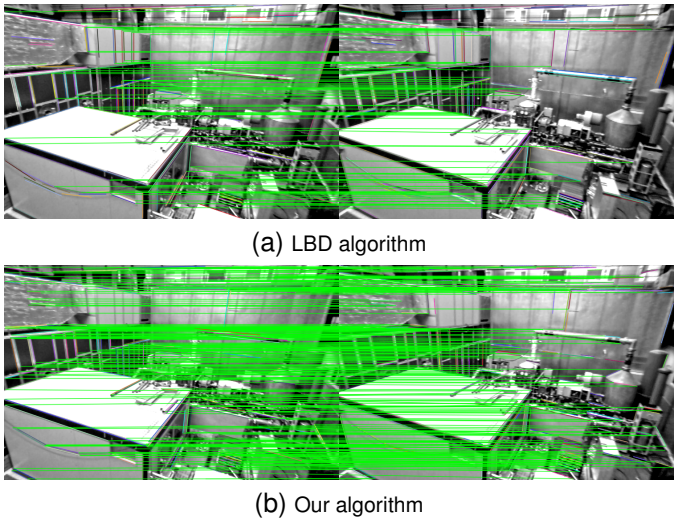
B. Multi-modal Feature Processing

The system frontend processes the raw sensor measurements to obtain the feature association. For visual point features, we initially predict the coordinates of extracted points in the next frame using the IMU-provided pose, and then utilize the sparse optical flow algorithm [40] to track these extracted features. For visual line features, we introduce an advanced coarse-to-fine line matching algorithm that eliminates the need for extracting LBD descriptors [27]. Our algorithm uses a two-stage line matching approach. Initially, in the coarse matching phase, we construct descriptors for line segments based on their geometric characteristics in the images. Utilizing the Heaviside function [22], we then perform coarse matching between line segments to find candidate line matches. In the fine matching phase, we dynamically extract points along line segments based on their lengths. Subsequently, we employ the optical flow verification to identify the optimal matching line segment.

1) The Coarse Matching Phase: The IMU measurements provide an initial pose transformation between two consecutive frames, image i and image j. After the initial pose transformation, the matched line segments between the two images are expected to exhibit similar geometric properties, such as the length, direction and endpoints coordinates. Therefore, we encode these geometric properties into descriptors for the line segments and select candidate matches by comparing these descriptors' similarity, akin to loop detection. These descriptors, unlike LBD descriptors, capture inherent line segment characteristics without complex computations. For a line segment k in image i, its line length, direction, and coordinates of its two endpoints are encoded into into a descriptor $M_k\left(l_k, \theta_k, p_{sk}, p_{ek}\right)$. Similarly, for a line segment h in image j, we derive its descriptor $M_h\left(l_h, \theta_h, p_{sh}, p_{eh}\right)$. The similarity between two line segments is defined as:

$$S\left(M_k, M_h\right) = \sum_u H\left(M_k(u_k), M_h(u_h)\right) \tag{1}$$

where $H\left(x\right)$ denotes the Heaviside function, which evaluates to 1 when x is below a specific threshold and 0 otherwise.



(a) LBD algorithm

(b) Our algorithm

Figure 3: Results of line matching with our proposed algorithm and LBD. (a) LBD algorithm, (b) Our algorithm. The solid green lines represent the connections between successfully matched line segments in the two images.

$H\left(M_k(u_k), M_h(u_h)\right)$ respectively denotes the similarity of two line segments in terms of their length, direction, and coordinates of the two endpoints. Specifically, while matching the line segments k from image i and h from image j, the Heaviside function for their similarity is computed as follows:

$$H\left(M_k(u_k), M_h(u_h)\right) = \begin{cases} 1 & \text{if } |u_k \times T - u_h| \leq \delta \\ 0 & \text{else} \end{cases} \tag{2}$$

where T represents the initial pose transformation provided by the IMU measurements. δ is the detection error of line segments and the uncertainty in IMU pose. During the matching process, as the magnitude of IMU-provided pose increases, along with the uncertainty of the matching, the value of δ needs to be adjusted accordingly. For line length and the coordinates of its two endpoints, we set $\delta = 30 + T/10$. For line direction, it's $\delta = 1 + T/10$. The greater the value of $S\left(M_k, M_h\right)$, the more similar the two lines are. Thus, in the coarse matching phase, we select the candidate lines with the highest similarity. Specifically, in our experiments, we exclusively select those candidate line segments with a similarity score of exactly 4 (meaning they match in all 4 considered geometric characteristics), calculated from Eq.(1), for the subsequent matching step.

2) The Fine Matching Phase: We eliminate most lines with low similarity scores, greatly reducing computational complexity in the coarse matching phase. Next, in the fine matching phase, for a given line segment k in image i as shown in Fig.2, we initially employ an adaptive step-down approach for sampling points based on the direction θ_k and length l_k of the line. The initial step length s_0 is set to 10px, then the sampling step length s for each line segment is set to $s = s_0 + \lceil {}^{l_k}/_{s_0} \rceil$. This adaptive method yields a series of sampling points on line k. For a sampling point p_i^k, we can

Algorithm 1 Pseudocode of Lines Matching Algorithm

Input: k: a given line on image i; T: the pose transformation; h_j: sets of detected lines on image j;
Output: pairs of matched lines between two images;

1: Construct the geometric descriptors for each line segment and compare their similarity using the Heaviside function.
2: Select the candidate lines h_j^m with the highest similarity based on Eq.(1).
3: Downsample the line k with an adaptive step length to obtain a series of sampling points k_i^n;
4: Obtain the set of predicted points k_j^n through optical flow tracking;
5: **while** $s < m$ **do**
6: **for** $t = 1$; $t <= n$; $t++$ **do**
7: Calculate the distance Δd from the predicted point k_j^t to the line h_j^s with Eq.(5);
8: **if** $\Delta d < d_{\text{thre}}$ **then**
9: Match point k_j^t with line h_j^s
10: **end if**
11: **end for**
12: $s++$
13: **end while**
14: Calculate matching rate between k and each candidate line in the set h_j^m, and then return the pair of lines that have the highest matching rate as the optimal match.

predict its corresponding point $p_j^{k'}$ in image j, utilizing IMU-provided pose T:

$$\lambda_j K^{-1} p_j^{k'} = \lambda_i T K^{-1} p_i^k \tag{3}$$

$$p_j^{k'} = K T K^{-1} p_i^k \tag{4}$$

where K is the the camera intrinsic parameter matrix, λ_i and λ_j are the depths of the point p_i^k and $p_j^{k'}$. Inspired by [41], we consider λ_i is equal to λ_j approximately, which simplifies Eq.(3) to Eq.(4). After obtaining $p_j^{k'}$, the optical flow is used to track $p_j^{k'}$ on image j. After tracking, we obtain the optical flow points p_j^k and measure their distances to each candidate line segment h in image j, identifying the closest line segment.

$$d_{p_j^k} = \left| \left(p_j^k - p_s^h \right) \times \theta^h \right| \tag{5}$$

where p_s^h, θ^h are respectively the endpoint and direction vector of the candidate line h. If the distance $d_{p_j^k}$ exceeds the threshold (set to 2px), the predicted point p_j^k is considered not to be on the line h. We repeat this process process and consider the two lines as matched if more than over half of line k's predicted points are computed on the line h. The line matching process is described in detail in Algorithm 1: steps 1 and 2 correspond to the coarse matching stage, while steps 3 to 14 are for fine matching.

The qualitiative results of line matching with our proposed method are shown in Fig.3, where it can be seen that we successfully tracked 136 line segments compared to the 46 line segments tracked by the LBD algorithm in the same scene, highlighting that our method outperforms the LBD. A more comprehensive evaluation of our algorithm's efficiency and accuracy is provided in SectionIV-B.

C. Robust Iterative Graph Optimization

To mitigate the impact of outliers on the backend optimization, we propose a robust iterative optimization method that incorporates adaptive loss function based on the quality of feature matching and errors based on graduated non-convexity [23]. Our proposed method is explained in the following sections.

1) Proposed Outlier-Robust Bundle Adjustment Formulation: During the system's backend optimization, we primarily address the outliers in the matching of visual and lidar features. To effectively suppress these outliers, a new robustified objective function can be formulated based on graduated non-convexity [23] and Black-Rangarajan duality [24]:

$$\rho\left(w_i, r\right) = \left[{w_i}^2 \left\| \boldsymbol{r}\left(\boldsymbol{z}_i, \boldsymbol{x}\right) \right\|^2 + \Phi\left(w_i\right) \right] \tag{6}$$

where $\left\| \boldsymbol{r}\left(\boldsymbol{z}_i, \boldsymbol{x}\right) \right\|^2$ is the residual of the feature during the optimization process, $w_i \in [0,1]$ is the weight corresponding to each feature's residual. $\Phi\left(w_i\right)$ defines a penalty function on the weight on w_i.

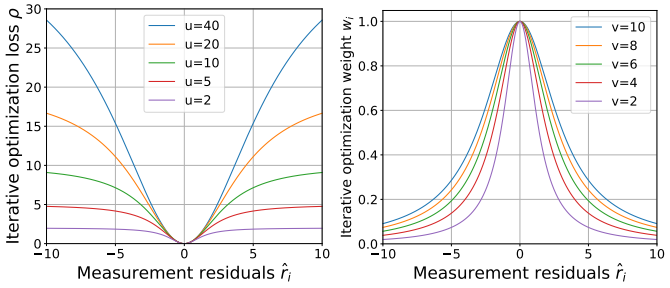
To maintain real-time performance of the optimization process, we only use keyframes to for sliding windows. For a sliding window of N_k keyframes, according to Eq.(6), the system optimal states are obtained through minimizing:

$$\begin{aligned} \min_{x, w_i \in [0,1]} \Bigg\{ & \left\| \boldsymbol{r}_{\mathfrak{p}} \right\|^2 + \left\| \boldsymbol{r}_{\boldsymbol{D}} \right\|^2 + \sum_{i=1}^{N_k} \left\| \boldsymbol{r}_{J_i} \right\|^2 + \sum_{i=1}^{N_{E_k}} w_e^2 \left\| \boldsymbol{r}_{\mathcal{E}_i} \right\|^2 \\ & + \sum_{i=1}^{N_{S_k}} w_s^2 \left\| \boldsymbol{r}_{\mathcal{S}_i} \right\|^2 + \sum_{i=1}^{N_{P_k}} w_p^2 \left\| \boldsymbol{r}_{\mathcal{P}_i} \right\|^2 \\ & + \sum_{i=1}^{N_{L_k}} w_l^2 \left\| \boldsymbol{r}_{\mathcal{L}_i} \right\|^2 + \sum_{i \in \{e,s,p,l\}} \Phi(w_i) \Bigg\} \end{aligned} \tag{7}$$

where x represents the system state, w_e, w_s, w_p and w_l are the adaptive weights of lidar and visual features residuals respectively. $\boldsymbol{r}_{\mathfrak{p}}$ is the system prior factor marginalized by Schur-complement [42], $\boldsymbol{r}_{\boldsymbol{D}}$ represents the system loop detection factor. $\boldsymbol{r}_{J_i}$ is the residual of IMU measurement. $\boldsymbol{r}_{\mathcal{E}_i}$ and $\boldsymbol{r}_{\mathcal{S}_i}$ define the residual of lidar edge features and surf features constraints. $\boldsymbol{r}_{\mathcal{P}_i}$, $\boldsymbol{r}_{\mathcal{L}_i}$ denote the reprojection residuals of the visual points and lines. Φ is the penalty terms of adaptive weights. In this paper, we define the penalty function Φ as:

$$\Phi\left(\omega_i\right) = \mu v \left(w_i - 1\right)^2 \tag{8}$$

Here, μ a control parameter whose value changes after each iterative optimization, similar to graduated non-convexity [23]. v is the parameter corresponding to the quality of each feature match. Specifically, for vision, features that are tracked more often are considered more stable. Consequently, we believe that the tracking times of visual features are an important indicator for evaluating the quality of feature matching. In this regard, for visual features, we define $v = n/w$, where n represents the tracking times of the feature, and w is the size of the sliding window, which is set to 10. We consider the



(a) (b)

Figure 4: Changes of our objective function loss and weight with two adaptive parameters we designed. (a) Iterative optimization loss ρ with respect to the control parameter μ for $v = 1$, (b) Iterative optimization weight w_i with respect to the feature matching quality parameter v for $\mu = 1$.

appearance consistencies of lidar feature associations. For any given lidar feature p, its corresponding v value is defined as:

$$v = \exp\left(-\frac{1}{m}\sum_{j=1}^{m}|I_p - I_j|\right) \tag{9}$$

Where I_p is the intensity value of the feature point p, and I_j is the intensity value of its associated nearest features. m represents the number of associated nearest features, set to 5.

2) Analysis of State Update in Proposed BA Formulation: After obtaining the correspondence of measurements from the frontend, we solve Eq.(7) by iterative optimization, where it first optimizes over the system state x_t, and then optimize over the weight w_i^t of the measurements residuals for each inner iteration, and these weights are used to update the system state x_{t+1} for the next iteration. More specifically, after obtaining the system state x_t and the corresponding residuals, we can update the weight w_i^t with fixed x_t:

$$w_i^t = \min_{w_i\in[0,1]}\sum_{i=1}^{N}\left[w_i^2\left\|\boldsymbol{r}\left(\boldsymbol{z}_i,\boldsymbol{x}\right)\right\|^2 + \Phi\left(w_i\right)\right] \tag{10}$$

where $\left\|\boldsymbol{r}\left(\boldsymbol{z}_i,\boldsymbol{x}\right)\right\|^2$ is a constant for fixed x_t. For ease of subsequent expression, we use $\hat{r}_i^2$ to represent $\left\|\boldsymbol{r}\left(\boldsymbol{z}_i,\boldsymbol{x}\right)\right\|^2$. Moreover, combining with Eq.(8), the weight w_i^t in Eq.(10) update at iteration t can be solved in closed form as:

$$w_i^t = \frac{\mu v}{\hat{r}_i^2 + \mu v} \tag{11}$$

According to Eq.(11), we can observe that the larger the matching error $\hat{r}_i$ of the feature, the smaller is its corresponding weight. By substituting the weights w_i^t obtained from Eq.(11) into Eq.(6), we can derive the latest optimization objective function loss $\rho\left(w_i^t, \hat{r}_i\right)$ for each feature as follows:

$$\rho\left(w_i^t, \hat{r}_i\right) = \frac{\mu v \hat{r}_i^2}{\hat{r}_i^2 + \mu v} \tag{12}$$

Based on Eq.(11) and Eq.(12), we can derive the curves representing the change in loss and weights of our optimization objective function with the two parameters we designed, as showed in Fig.4. The impact of change in control parameter μ on the iterative optimization loss for each feature is showed in Fig.4(a), and it can be seen that the value of μ gradually decreases after each optimization step. In our experiments, μ is reduced to $\mu/5$ after each step. From Fig.4(a), we observe that during the optimization process, as the control parameter μ gradually decreases, the change curve of our objective function's loss becomes increasingly smoother. Simultaneously, the non-convexity of the objective function's loss progressively increases, and concurrently, its ability to suppress outliers intensifies. We initiate the optimization with a relatively high value of μ, starting from a convex surrogate model, and gradually reduce the μ value throughout the optimization process. This approach prevents the objective function from exhibiting high non-convexity, thereby avoiding getting trapped in local optima. Specifically, for the initial value of μ, we set μ to be 200 times the maximum residual of the feature associations in our experiments.

Fig.4b illustrates the changes in weights of our optimization objective function with respect to the feature matching quality parameter v. We can observe that as the quality of feature matching decreases, signifying a higher instability of the feature, its corresponding weight diminishes. This leads to a reduced influence on the optimization direction of the objective function, thereby accelerating the convergence speed of the objective function. Additionally, it can be observed that the larger the residuals of feature matching, the lower the corresponding weight assigned to them. Therefore, our designed optimization method effectively suppresses features with poor matching quality and large matching residuals, such as outliers that commonly occur in dynamic and low-light scenes. This suppression reduces their impact on the optimization process. More detailed comparisons of our optimization approach with existing optimization approaches is provided in SectionIV-D.

D. Two-stage Loop Closure Detection

We propose a two-stage loop closure detection strategy, which simultaneously uses visual and lidar information for detecting similar places. For the visual part, we utilize DBoW3 [43] for loop detection. When a new image keyframe is detected, we extract BRIEF descriptors [44] and match them with previously extracted descriptors. If the similarity between current and historical image frame descriptors exceeds a certain threshold, a visual loop constraint is formed. The latest visual loop information is then incorporated into the graph optimization. We identify the closest global lidar frames corresponding to the visual loop matches and then perform iterative closest point (ICP) processing for relative pose transformations. Finally, once these transformations are determined, we update the pose of the historical frames. For the lidar part, we design a novel 3D lidar descriptor, which utilizes the 2D information of the point cloud for coarse loop detection and the 3D information of the point cloud for fine loop detection. Similar to the visual part, upon detecting a new lidar loop match, ICP operations are also performed to acquire relative pose transformations, and then the global historical frame poses are updated accordingly.

1) Coarse Loop Detection of Point Clouds: As shown in Fig.5, we define the maximum detection height (vertical angle)

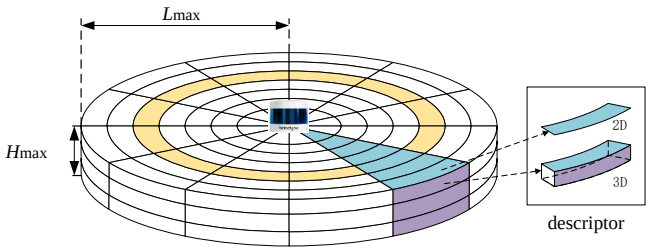


Figure 5: Structure of the proposed coarse-to-fine global 3D lidar descriptor with spatial, height and intensity information.

of the current lidar frame as $H_{\max}$, the maximum detection horizontal distance as $L_{\max}$ with the detection horizontal angle as 2π. We divide the point cloud into $N_h \times N_l \times N_a$ bins along the above three directions, the widths of the three directions are $\frac{H_{\max}}{N_h}$, $\frac{L_{\max}}{N_l}$ and $\frac{2\pi}{N_a}$. In this paper, we set $N_h = 8$, $N_l = 20$ and $N_a = 40$. In contrast to [21], which solely relies on the height information of the point cloud, our approach incorporates the 2D information to quickly search out candidate frames in the coarse detection stage and utilizes 3D spatial information of the point cloud for fine detection stage. When a new lidar frame is received, the point cloud is projected onto the 2D plane, and the projected points are divided into $N_l \times N_a$ bins in the meanwhile. For each 2D bin, its 2D descriptor ϕ_i is encoded as:

$$\phi_i = n \cdot Z_{max} \tag{13}$$

Where n represents the ratio of the projected points in each bin to the total number of points in the point cloud, Z_{max} is the z value of the highest point cloud in each bin. The ring encoding function ψ is defined to convert average coded value for each row to the ring value r_i. Ring values r_i are combined into a N_l-dimension vector called ring vector R. The details of ring vector R can be expressed as:

$$R = (\psi(r_1), \ldots, \psi(r_{N_l})) \tag{14}$$

The ring encoding function ψ we use is the occupancy ratio of a ring using L_0 norm:

$$\psi(r_i) = \frac{\|\phi_1 + \phi_2 + \cdots + \phi_{N_l}\|}{N_a} \tag{15}$$

Where ϕ_i represents the 2D descriptor in Eq.(2). Although being less informative than 3D lidar descriptor, 2D ring vector enables fast search for finding possible candidates for loop. When a new lidar measurement is received, we extract the ring vector of the point cloud and construct a k-d tree for the nearest neighbor search. Through 2D coarse detection, several suboptimal loop candidate frames can be found from database that are most similar to the considered candidate keyframe. We further use finer 3D descriptor to get the optimal candidate frame from these list of selected matches using ring vector.

2) Fine Loop Detection of Point Clouds: In this paper, we leverage the 3D information obtained from lidar frames for loop detection based on [45], [46]. The 3D descriptor in the fine loop detection stage includes height information, maximum intensity information and point density information

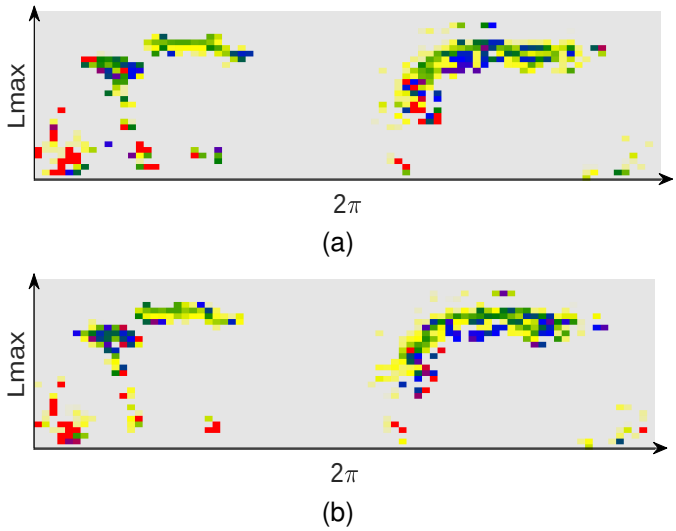


Figure 6: Visualization of the 3D point cloud descriptors for two loop-closure lidar frames on the KITTI dataset.

of each bin. We define a weight for each type of information and design the weighting function. At the same horizontal distance and horizontal angle, a higher bin with a larger vertical angle is therefore assigned a greater height weight. For an arbitrary bin B_{ijk}, the height weight of B_{ijk} can be defined as:

$$H_{ijk} = \frac{2^{k-1}}{\sum_{s \in N_h} 2^{s-1}} \tag{16}$$

where H_{ijk} is the height weight of B_{ijk}, $k \in N_h$, $j \in N_a$ and $i \in N_l$. After encoding height information, we move to intensity information as its an important characteristic that distinguishes different objects based on reflectance. We take the maximum intensity of points in B_{ijk} as the intensity value m_{ijk}. If there is no point in B_{ijk} ($B_{ijk} = \emptyset$), the intensity value m_{ijk} is set to 0. The intensity weight of B_{ijk} can be defined as:

$$I_{ijk} = \begin{cases} 1 & \text{if } \bar{m}_{ij} = 0, m_{ijk} > 2 * \bar{m}_{ij} \\ \frac{m_{ijk}}{2 * \bar{m}_{ij}} & \text{else} \end{cases} \tag{17}$$

where I_{ijk} is the intensity weight of B_{ijk}, $\bar{m}_{ij}$ is the median intensity of bins in the same horizontal distance and horizontal angle. Lastly, we have the point density weight where the bin with higher point density gets a larger weight. The density weight D_{ijk} of B_{ijk} can be defined as:

$$D_{ijk} = \begin{cases} 1 & \text{if } \bar{d}_{jk} = 0, d_{ijk} > 2 * \bar{d}_{jk} \\ \frac{d_{ijk}}{2 * \bar{d}_{jk}} & \text{else} \end{cases} \tag{18}$$

where d_{ijk} is the point density of B_{ijk}, and $\bar{d}_{ij}$ is the median point density of bins in the same vertical angle and horizontal angle. After estimating all weights, we can obtain the 3D descriptor E_{ijk} of B_{ijk} as:

$$E_{ijk} = H_{ijk} * I_{ijk} * D_{ijk} \tag{19}$$

The visualization of the 3D point cloud descriptors is shown in Fig.6. With coarse loop closure detection, we first get a

small subset of possible matches from the set of all possible matches. Let E^h represent the fine 3D descriptor of a possible match and E^c is that of the target frame, then the similarity between these two fine 3D descriptors can be estimated as:

$$d\left(E^h, E^c\right) = \frac{1}{N_a}\sum_{j=1}^{N_a}\left(1 - \frac{c_j^h \cdot c_j^c}{\left\|c_j^h\right\|\left\|c_j^c\right\|}\right) \tag{20}$$

where $d\left(E^h, E^c\right)$ is the similarity between two frames, c_j^h is the jth column vectors of E^h, and c_j^c is the jth column vectors of E^c. The range of similarity is $[0, 1]$, and a smaller value of similarity indicates that the two frames are more similar. We find the closest matching frame from the candidate frames by comparing the similarity value. The performance of our coarse-to-fine descriptor is experimentally verified in SectionIV-C.

IV. Experiments

In this section, the performance of various modules in our proposed algorithm are evaluated and compared with state-of-the-art methods on multiple real-world datasets. All of the experiments were performed on a computer with an Intel Core i7-6700HQ CPU with 2.60 GHz, 16 GB RAM, and ROS melodic [47].

A. Datasets

1) EuRoc Dataset: EuRoc dataset [48] contains stereo images from a global shutter camera and provides a ground-truth pose for each image frame given by the VICON motion capture system. This dataset is utilized to evaluate the performance of our proposed line matching algorithm.

2) KITTI dataset: KITTI dataset [49], equipped with a stereo camera and a Velodyne HDL-64E lidar sensor, provides precise ground-truth poses for each image and lidar frame using an IMU/GPS system. We use this dataset to evaluate the performance of our novel line matching algorithm and 3D loop closure descriptors.

3) NCLT datasets: NCLT dataset [50] contains the velodyne HDL-32E, the omnidirectional camera and Microstrain GX3 IMU data. Notably, the frequency of its ground-truth poses is higher than that of the lidar frames. Therefore, we use this dataset not only to evaluate the localization accuracy of our system but also to assess the performance of our proposed lidar 3D loop closure descriptors.

4) M2DGR datasets: M2DGR datasets [51] are collected by Velodyne VLP-32C, an RGB Camera and a Handsfree A9. provides precise ground-truth poses by GNSS-RTK suite. We utilize this dataset to evaluate the localization accuracy of our system.

5) UrbanNav and UrbanLoco datasets: UrbanLoco [52] and UrbanNav [53] datasets are recorded using a HDL-32E, a RGB camera and an Xsens MTi-10 IMU. We use these datasets to evaluate the localization accuracy of our system.

To facilitate clearer expression in the following sections, we have simplified the sequence names of the NCLT, M2DGR, UrbanNav, and UrbanLoco datasets, as shown in Table 1.

Table I: DATASETS OF ALL SEQUENCES FOR EVALUATION

Sequences	Name	Duration (min:sec)	Distance (km)
unhk-1	UrbanNav-HK-Medium-Urban-1	13: 05	3.64
unhk-2	UrbanNav-HK-Deep-Urban-1	25: 36	4.51
unhk-3	UrbanNav-HK-Data20190428	8: 07	2.01
unhk-4	UrbanNav-HK-Data20200314	5: 00	1.21
ulhk-1	UrbanLoco-HK-Data20190426-2	3: 32	0.8
ulhk-2	UrbanLoco-HK-Data20190426-1	3: 01	0.7
m2dgr-1	M2DGR-Gate-02	5: 27	0.20
m2dgr-2	M2DGR-Street-04	14: 20	1.30
m2dgr-3	M2DGR-Street-08	8: 31	0.50
m2dgr-4	M2DGR-Gate-03	4: 43	0.80
NCLT01/nclt-1	NCLT-2013-01-10	17: 02	0.26
NCLT02/nclt-2	NCLT-2012-04-29	43: 18	3.18

B. Evaluation of visual line features matching

We evaluated the performance of our proposed Coarse-to-Fine (CTF) line segment matching algorithm and compared it to traditional appearance-based tracking with LBD [27] and the line matching method proposed by PLC-VIO [11], using the EuRoC dataset [48] and the KITTI dataset [49]. We tested both matching accuracy and computational cost on all fifteen sequences. To measure matching accuracy, we classified lines as correctly matched if the projection error of corresponding line endpoints was less than two pixels, utilizing the groundtruth pose between two image frames [54]. We used the ratio of correctly matched lines to jointly detected lines as our matching performance metric (Match Accuracy), and measured the time required for lines matching between consecutive frames as the computational cost metric (Match Time). In this experiment, we employ EDLines [55] algorithm for line segment detection.

The results in Table II show the average match time and match accuracy on all sequences of EuRoc dataset and KITTI dataset. Our proposed method is highly efficient and outperformed others by only requiring an average computational time of 3.79ms across all evaluated sequences. In comparison, the PLC-VIO's method (PLC) required an average of 8.12ms, while the LBD descriptor-based method took 33.69ms. Examining accuracy, our proposed method shines with an average matching accuracy of 82.33% across the sequences, while the PLC method registers an accuracy of 74.37%, and the LBD descriptor method trails at 65.33%. The main reason for this improvement is that our method eliminates the need to extract line descriptors and benefits from the strict geometric constraints. Our geometric descriptor for lines and two stage matching significantly minimizes the computational overhead. We further select the first 50 frames of sequences V1-02 and V2-03 for a detailed comparison, as shown in Fig.7. It is apparent from Fig.7 that our algorithms consistently surpass the other two algorithms in both matching rates and matching time.

To verify the performance improvements of our proposed Coarse-to-Fine (CTF) line segment matching algorithm on the localization accuracy (the absolute traditional position error, APE) of SLAM systems, we integrated the algorithm into our MMF-LVINS and UV-SLAM [26] systems, and conducted accuracy comparison experiments on four sequences from

Table II: LINES MATCHING PERFORMANCE COMPARISON

Map ID	Frame count	Ours		PLC-VIO (PLC) [11]		LBD [27]	
		Match time(ms)	Match accuracy(%)	Match time(ms)	Match accuracy(%)	Match time(ms)	Match accuracy(%)
MH-01	3682	3.61	89.63	7.84	83.03	31.02	78.91
MH-02	3040	3.73	88.79	7.87	81.79	31.34	72.77
MH-03	2700	3.34	91.65	8.43	82.11	32.53	76.46
MH-04	2033	3.56	90.98	8.64	83.93	34.18	81.69
MH-05	2273	3.43	88.04	8.85	84.59	34.61	72.17
V1-01	2912	3.64	90.59	8.41	84.72	33.32	75.82
V1-02	1710	4.21	87.60	8.92	79.41	32.38	77.01
V1-03	2149	3.34	83.67	8.89	73.48	32.24	68.69
V2-01	2280	3.80	92.02	8.95	85.16	34.07	80.97
V2-02	2348	3.68	87.45	8.84	77.92	33.05	68.12
V2-03	2336	3.45	85.05	9.31	78.61	34.80	70.65
KITTI-00	4541	4.18	64.60	7.19	54.96	36.37	37.51
KITTI-05	2761	4.69	65.56	6.46	57.01	37.55	40.54
KITTI-07	1101	4.50	68.02	6.44	59.50	34.87	42.18
KITTI-08	4071	3.99	61.30	6.91	49.41	33.23	36.53
Average value		**3.79**	**82.33**	8.12	74.37	33.69	65.33

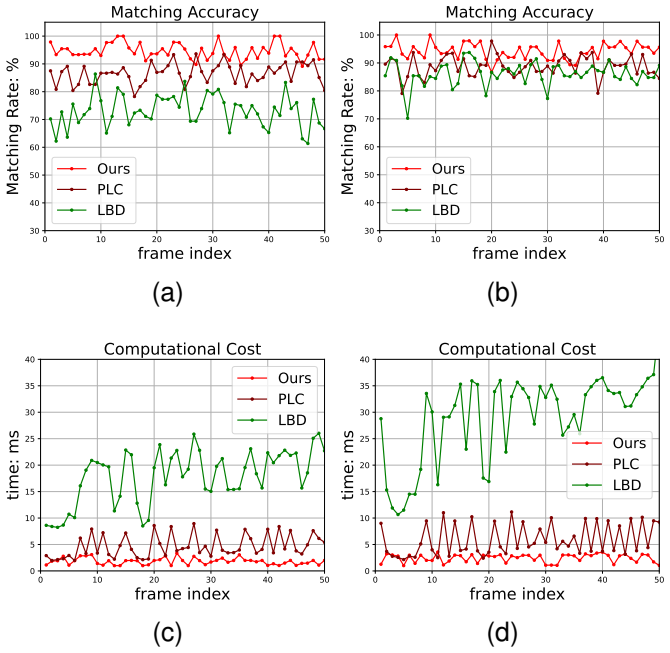


Figure 7: Matching performance and Computational Cost between two consecutive frames with our algorithm, PLC and LBD on sequence V1-02 and V2-03. (a)(c) are the results on sequence V1-02. (b)(d) are the results on sequence V2-03.

Table III: LOCALIZATION ACCURACY ON THE URBANNAV AND M2DGR DATASET [m]

Sequences	Ours		UV-SLAM [26]	
	w/ CTF	w/o CTF	w/ CTF	w/o CTF
unhk-3	**2.88**	3.12	**27.95**	29.25
unhk-4	**1.40**	1.55	**15.53**	18.31
m2dgr-2	**1.22**	1.30	**22.38**	25.23
m2dgr-4	**0.44**	0.51	**9.49**	10.67
Mean	**1.49**	1.62	**18.83**	20.86

CTF represents our proposed Coarse-to-Fine line matching algorithm. w/: with, w/o: without.

UrbanNav and M2DGR datasets, with the results presented in Table III. According to Table III, both systems demonstrated enhanced localization accuracy after incorporating our proposed Coarse-to-Fine line matching (CTF) algorithm. Specifically, our MMF-LVINS system improved its average accuracy by 8% and UV-SLAM by 10% across four sequences with the CTF algorithm, demonstrating its significant impact on enhancing SLAM systems' localization accuracy.

C. Evaluation of lidar loop detection

To evaluate the performance of our lidar loop detection, we compared it with four other state-of-the-art lidar loop detection methods: Scancontext [21], IRIS [31] ISC [30] and FreSco [20], using a 3D point cloud dataset from the KITTI dataset [49] and NCLT dataset [50]. The performance of our proposed descriptors is evaluated by the precision-recall (PR) curve, which is generated by calculating precision P and recall R values. The precision and recall are defined as:

$$P = TP/(TP + FP) \tag{21}$$

$$R = TP/(TP + FN) \tag{22}$$

Where TP is the number of correct matches pairs that have a Euclidean distance under 4 m and a descriptor similarity below the threshold. FN represents pairs within a 4 m distance but with descriptor similarity above the threshold, and FP counts those pairs exceeding a 4 m distance, but with a descriptor similarity under the threshold. We further computed the maximum value of F1 score and Extended Precision [56] (EP), conducting a detailed comparison with all the aforementioned descriptors. The F1 score and EP value are defined as follows:

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{23}$$

$$EP = 0.5 \times (R_{P100} + P_{R_0}) \tag{24}$$

Where R_{P100} is the max recall at 100% precision, and P_{R_0} is the precision at minimum recall.

The PR curve in Fig.8 demonstrates the performance of all descriptors. In the KITTI05, which contain loop closure

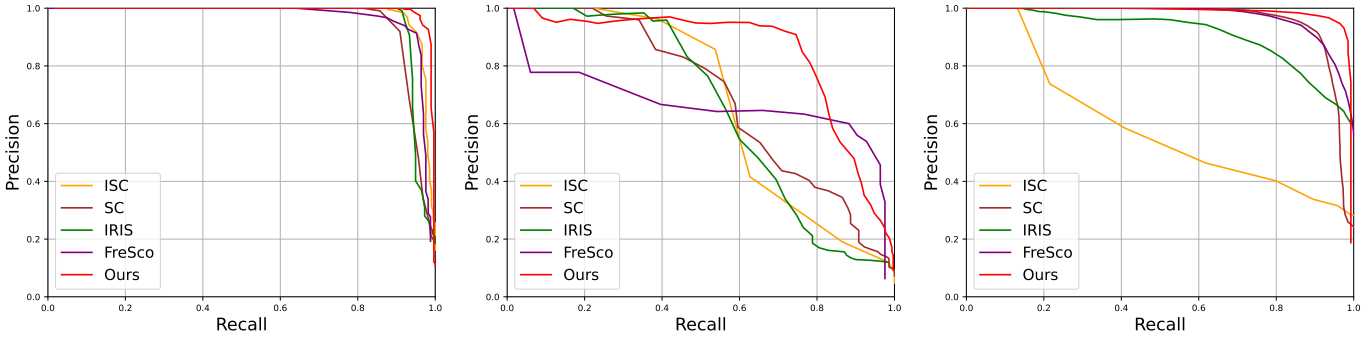


Figure 8: The precision-recall curves of different descriptors. (a) KITTI 05; (b) KITTI 08; (c) NCLT02.

Table IV: F1 SCORE / EXTENDED PRECISION RESULTS ON KITTI DATASET AND NCLT DATASET

Method	KITTI00	KITTI02	KITTI05	KITTI06	KITTI07	KITTI08	NCLT01	NCLT02
SC [21]	0.932/0.938	0.728/0.576	0.889/0.906	0.955/0.946	0.514/0.585	0.639/0.608	0.838/0.730	0.901/0.799
ISC [30]	0.903/0.897	0.722/0.516	0.934/0.927	0.953/0.924	0.540/0.591	0.660/0.611	0.640/0.618	0.665/0.634
IRIS [31]	0.874/0.864	0.829/**0.874**	0.905/0.876	0.940/0.942	0.578/0.522	0.617/0.585	0.862/0.713	0.821/0.672
FreSco [20]	0.943/0.916	0.868/0.849	0.922/0.897	0.973/0.957	0.695/0.708	0.712/0.693	0.898/0.815	0.912/0.851
Ours	**0.963/0.955**	**0.893**/0.856	**0.967/0.953**	**0.984/0.979**	**0.722/0.769**	**0.849/0.842**	**0.916/0.853**	**0.955/0.903**

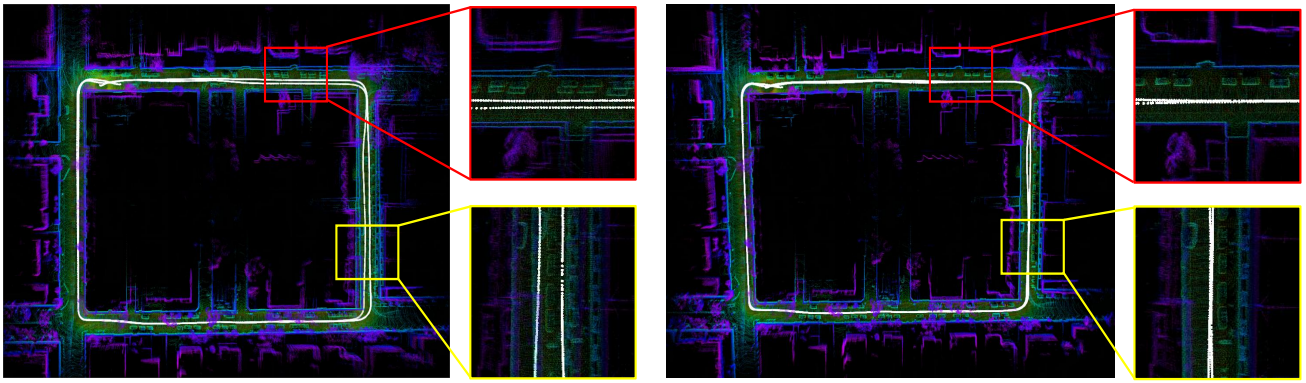


Figure 9: Comparison of system localization and mapping results with and without loop closure on unhk-4, where the same color indicates the same location. (a) without loop closure, the localization accuracy (APE) is 1.59; (b) with loop closure, the localization accuracy (APE) is 1.40.

events with limited rotation, all methods perform reasonably well, as shown in Fig.8a, with our proposed method offering the best performance. In the NCLT02 and KITTI08 sequence, which contains loop closure events with significant rotation and reverse loop closure events, the other four methods didn't achieve good performance, but our method still maintained its top performance, displaying a strong rotation-invariance characteristic.

Table IV presents the maximum F1 scores and Extended Precision (EP) values for all descriptors. As can be discerned from the Table IV, our method consistently achieves the best detection outcomes in most scenarios. Scancontext (SC), ISC, IRIS and FreSco are the 2D descriptors based on the Bird's Eye View (BEV) image. They mainly utilize the 2D information of point clouds and deliver decent performance in scenes with limited rotation. However, their effectiveness falters on the KITTI08 sequence, which only contains reverse loop closure events, and the KITTI07 sequence where misjudgments arise due to traffic congestion. Our average F1 scores and EP values across the eight sequences improved by 16% and 18% compared to SC, and by 15% and 20% compared to ISC. The results confirm that our proposal to employ spatial, height and intensity information in coarse to fine manner is very helpful for identifying challenging place revisits.

We integrated our proposed loop closure module into the whole system and compared its impact on the positioning accuracy and map building. Specifically, we evaluated the performance of the system with and without the loop closure module. Fig.9 demonstrates that the integration of our loop

Table V: THE APE OF DIFFERENT OPTIMIZATION APPROACHS [m]

Sequences	Huber		Cauchy		GMC		Ours	
	w/ loop	w/o loop	w/ loop	w/o loop	w/ loop	w/o loop	w/ loop	w/o loop
unhk-1	6.17	20.92	3.11	12.11	2.91	7.14	**2.62**	**6.37**
unhk-3	5.04	10.88	3.21	7.44	3.16	7.17	**2.88**	**5.58**
unhk-4	1.70	10.57	1.61	2.01	1.63	6.34	**1.40**	**1.59**
m2dgr-1	5.74	11.63	2.14	3.13	1.99	3.71	**1.73**	**2.81**
m2dgr-2	7.16	19.36	2.08	3.09	1.88	2.72	**1.22**	**1.66**
m2dgr-4	3.70	6.17	0.75	1.13	0.54	0.93	**0.44**	**0.74**

w/: with, w/o: without.

closure module results in higher positioning accuracy, leading to the construction of a more consistent map.

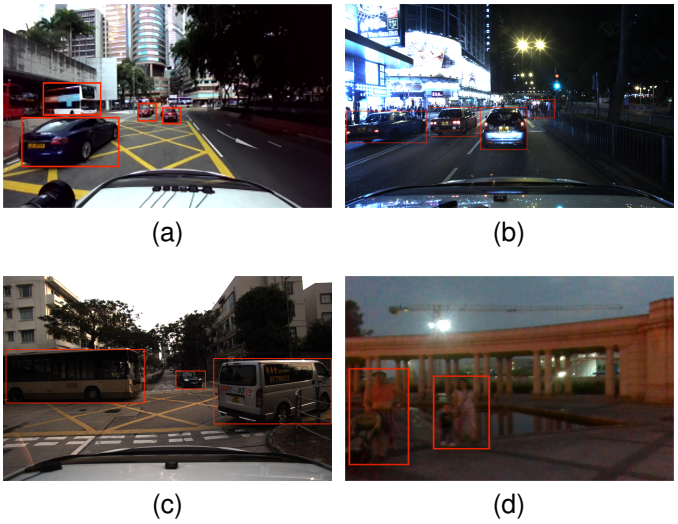


Figure 10: Scene visualization of unhk and m2dgr datasets. (a) is the scene from unhk-1, (b) is the scene from unhk-3, (c) is the scene from unhk-4, and (d) is the scene from m2dgr-1. Dynamic objects in the images are highlighted with red boxes. Data for unhk-3 and m2dgr-1 are collected during night scenes.

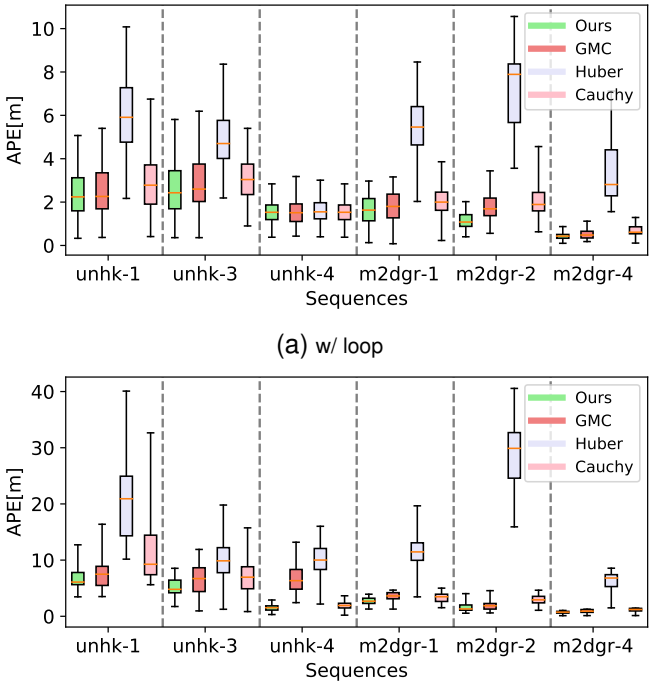


Figure 11: APE results of state-of-the-art optimization approach and ours on six challenging datasets. Our algorithm shows promising performance compared with the other state-of-the-art methods.

D. Evaluation of our Proposed Outlier Robust Optimization

To validate the effectiveness of our optimization method in mitigating the impact of outliers on system accuracy, we conducted detailed experimental comparisons in complex scenarios against traditional optimization methods using Huber, Cauchy loss functions and Geman-McClure (GMC) [57]. We selected six challenging sequences from Table I: unhk-1, unhk-3, unhk-4, m2dgr-1, m2dgr-2 and m2dgr-4. These sequences are dominated by dynamic scenes or low-light conditions, which lead to some feature association outliers. The visualization of the scenes from these sequences are shown in Fig.10.

Table V displays the absolute traditional position error (APE) values of localization accuracy for our system with different optimization approaches on the six datasets, under conditions with and without the loop closure module. From this table, we can discern that our optimization approach achieves the highest localization accuracy in all challenging scenarios, both with and without loop closure. In contrast, the Huber method struggles to reject outliers effectively, resulting in the poorest accuracy. Fig.11 shows the box plot of APE accuracy for all optimization approaches, and it can be seen that our optimization approach performs best in terms of both APE accuracy and data variability (standard deviation). Furthermore, we further compared our optimization approaches with Huber and Cauchy on the sequence m2dgr-4, as shown in Fig. 12. It is observable that the trajectories corresponding to Cauchy and Huber both exhibit varying degrees of deviation when compared to the ground truth. In contrast, our approach demonstrates the highest trajectory accuracy, underscoring the robustness of our optimization approaches in complex scenarios.

We further compared the average number of iterations needed for each optimization approach to converge on all datasets. As shown in Fig.13, during the optimization process, our proposed objective function required about 3 iterations on average to converge, while GMC needed more than 4,

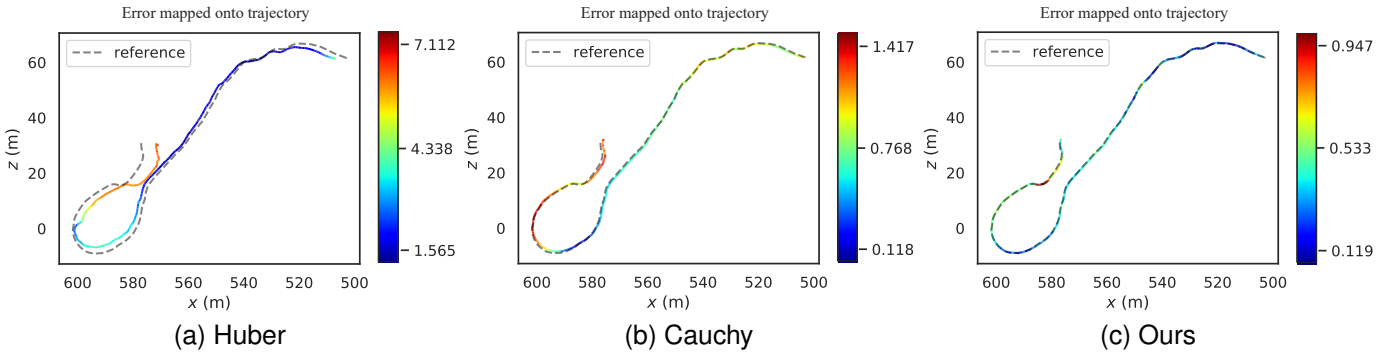


Figure 12: The trajectory accuracy of optimization approach on m2dgr-4 (w/o loop). (a) Huber; (b) Cauchy; (c) Ours.

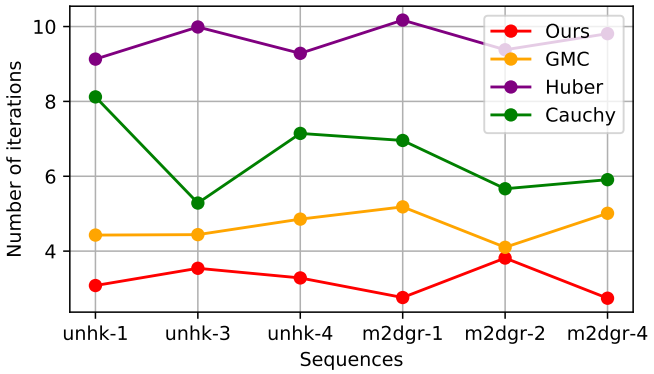


Figure 13: Comparison of the number of iterations for all optimization methods. Our method exhibited the fewest iterations.

Cauchy averages around 7, and Huber kernel with an average of 9 iterations, required the highest number of iterations. This proves that our optimization approach not only converges faster but also ensures the optimization performs correct feature association, which is the reason for high localization accuracy.

E. Evaluation of system localization accuracy

We conducted a series of experiments to validate the localization accuracy of our proposed system on various datasets, and we deploy our proposed system to compare with competing state-of-the-art systems, including (1) visual-inertial odometry: VINS-MONO [42], (2) lidar-only odometry: LOAM [58], (3) lidar-inertial odometry: LIO-SAM [59] and (4) lidar-visual-inertial odometry: LVI-SAM [6] and R3LIVE [8], where LVI-SAM contains only visual and lidar point features. R3LIVE is not an optimization-based method, instead, it is based on the visual and lidar direct method. Therefore, we focus on conducting experimental comparisons with LVI-SAM. The evaluation was performed on four datasets: the UrbanLoco [52], UrbanNav [53], M2DGR [51] and NCLT datasets [50]. All dataset sequences are collected in large scale scenarios with practical usage and deployment. Details of all the tested sequences used in this section, including name, duration, and distance, are listed in Table I.

Table VI shows the absolute traditional position error (APE) and the translational relative pose error per 100 meters (RPE) for all the tested methods. The results in Table VI demonstrate that our proposed MMF-LVINS outperforms state-of-the-art methods for most sequences by achieving smaller APE and RPE values. On m2dgr-1, m2dgr-3 and m2dgr-4 datasets, our proposed method achieves second best results, and the reason being the presence of many irregular objects like trees and vegetation. VINS-MONO, without using lidar information, performs the worst across all sequences and fails in ulhk-1 and ulhk-2 where the camera is facing upwards and they can't extract enough visual features, resulting in system crash. LOAM, without utilizing both lidar and IMU information, exhibits poor performance in some challenging scenarios ulhk-1, ulhk-2 and nclt-1. LIO-SAM does not utilize the multi feature information of vision, although it can achieve good accuracy, its accuracy is inferior to both our system and LVI-SAM in most sequences. R3LIVE performs better than our system on some datasets, but fails in ulhk-1 and ulhk-2 where the camera faces upwards as they cannot extract sufficient visual features. The direct method of vision is highly susceptible to environmental interference.

Compared to LVI-SAM, our system consistently outperforms it, in terms of localization accuracy across most sequences. Fig.14 presents a comparison of the whole trajectories of all systems on unhk-1 and unhk-2, which have the longest distance. Our system achieves the highest APE accuracy with only 2.62m and only 5.91m, respectively, and also has the best RPE accuracy with only 0.70% and 1.78%, respectively. In contrast, VINS-MONO performs the worst on these two datasets, with 100.26m APE accuracy and 29.84% RPE accuracy on unhk-1, and its accuracy on unhk-2 is also very poor. R3LIVE exhibits low APE accuracy on unhk-1 due to significant variations in scene illumination, which greatly impacts the performance of the visual component. On these two datasets, LVI-SAM and LIO-SAM achieve suboptimal localization accuracy, while LOAM performs poorly without the aid of an IMU. Our average APE and RPE accuracies on these two datasets improved by 33% and 26% respectively, compared to LVI-SAM. Finally, Fig.14 shows that our system's superimposed trajectory almost perfectly overlaps with the ground-truth, and each trajectory remains clearly

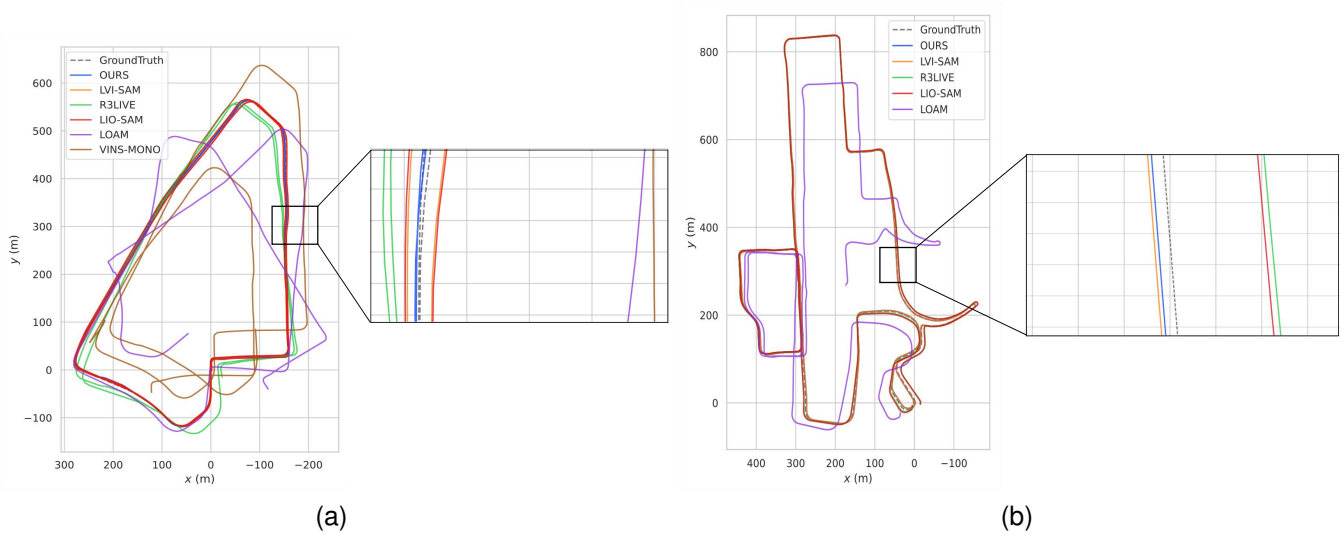


Figure 14: Comparison of trajectories of MMF-LVINS, LVI-SAM, R3LIVE, LIO-SAM, LOAM and VINS-MONO. (a) unhk-1; (b) unhk-2.

Table VI: LOCALIZATION ACCURACY PERFORMANCE COMPARISON

Map ID	OURS		LVI-SAM [6]		R3LIVE [8]		LIO-SAM [59]		LOAM [58]		VINS-MONO [42]	
	APE(m)	RPE(%)	APE(m)	RPE(%)	APE(m)	RPE(%)	APE(m)	RPE(%)	APE(m)	RPE(%)	APE(m)	RPE(%)
unhk-1	**2.62**	**0.70**	6.19	1.46	44.43	3.52	6.71	1.39	105.43	14.74	100.26	29.84
unhk-2	**5.91**	**1.78**	6.59	1.91	10.05	1.85	9.03	1.92	101.48	8.51	153.26	12.39
unhk-3	**2.88**	2.10	5.41	**1.89**	4.55	4.73	9.38	1.95	9.89	10.01	31.45	20.56
unhk-4	**1.40**	**0.42**	1.71	1.26	2.10	0.43	1.63	1.33	10.18	1.38	19.29	6.81
ulhk-1	**2.13**	10.05	2.69	**9.12**	-	-	2.74	15.83	3.70	19.16	-	-
ulhk-2	**1.99**	**5.01**	2.35	9.57	-	-	2.40	9.49	3.30	15.31	-	-
m2dgr-1	1.73	0.86	0.36	1.81	**0.32**	**0.14**	0.41	2.02	2.34	3.27	5.15	4.32
m2dgr-2	**1.22**	0.64	4.03	1.92	2.43	**0.57**	4.63	2.37	5.71	2.71	26.45	10.26
m2dgr-3	0.65	**0.31**	0.51	1.39	**0.37**	0.45	0.62	1.89	3.52	2.54	84.58	26.81
m2dgr-4	0.44	0.43	0.52	0.61	**0.31**	**0.39**	0.64	0.68	2.87	3.12	15.28	10.93
nclt-1	**2.75**	1.35	4.88	2.84	3.41	**1.17**	5.08	3.06	22.41	19.63	45.116	48.785

distinguishable without any noticeable errors.

Fig.15 presents the global map and detailed local map generated by our system on unhk-1. From Fig.15, our system is capable of depicting roads and small objects such as cars and trees with a high degree of clarity. Moreover, we only use keyframes to build the map, which significantly reduces map storage space and improves the efficiency of map building.

F. Evaluation of Real-Time Performance

MMF-LVINS can show good real-time performance for all the above tests. There are four running nodes in our system, including: Preprocessing, Initial Alignment, Robust Iterative Graph Optimization and Loop-clousre detection. As shown in Fig.16, the average runtime of each processing node is recorded for representative unhk-1 on UrbanNav dataset [53]. All the computations are conducted by a computer with an Intel Core i7-6700HQ CPU with 2.60 GHz, 16 GB RAM. The average processing time for the Preprocessing part of our system is 18ms, for the Initial Alignment process it's 25ms, for the Robust Iterative Graph Optimization process it's 72ms, and for the Loop-closure detection process it's 34ms. It's important to note that, as our system's Robust Iterative Graph Optimization and Loop-closure detection components process only keyframes, this also enables us to achieve commendable real-time performance.

V. Conclusion and Future Work

In this article, we propose a tightly coupled lidar-visual-inertial SLAM framework, which achieves high-precision localization and mapping. We made multiple contributions and enhancements to existing systems by introducing a coarse-to-fine visual line segment matching method, a novel global 3D lidar descriptor, a two stage loop closure detection algorithm and a outlier-robust optimization method to handle feature associations across multimodal features arising from visual and lidar data. In the end, through extensive experiments, we validated the effectiveness of our proposed methods on multiple real world datasets. In future work, we plan to enhance the current MMF-LVINS framework by leveraging on the semantic information from sensor data and make it robust to extreme weather conditions of the environment.

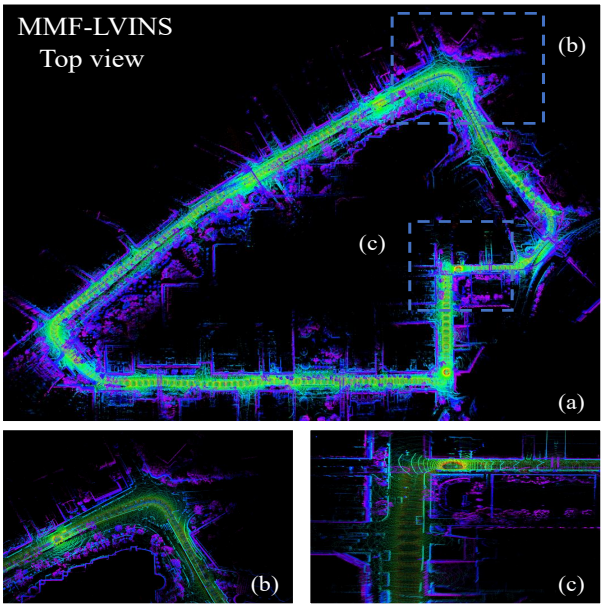


Figure 15: The global map and detailed local map generated by our system on unhk-1. (a) represents the top view of the global map, while (b), (c) depict the detailed local views.

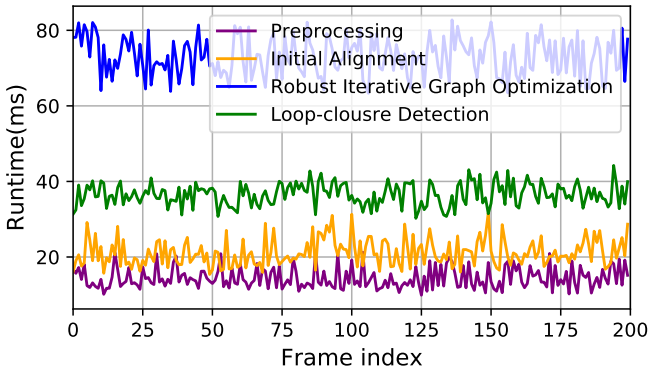


Figure 16: Processing time of each module of our system.

REFERENCES

[1] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, 2016.

[2] J. Huang, S. Wen, W. Liang, and W. Guan, "Vwr-slam: Tightly coupled slam system based on visible light positioning landmark, wheel odometer, and rgb-d camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.

[3] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 20–25.

[4] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau, and Y. Dupuis, "Survey on cooperative perception in an automotive context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 204–14 223, 2022.

[5] H. Saleem, R. Malekian, and H. Munir, "Neural network-based recent research developments in slam for autonomous ground vehicles: A review," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 13 829–13 858, 2023.

[6] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5692–5698.

[7] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.

[8] J. Lin and F. Zhang, "R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.

[9] X. Zhao, C. Miao, and H. Zhang, "Multi-feature nonlinear optimization motion estimation based on rgb-d and inertial fusion," *Sensors*, vol. 20, no. 17, p. 4666, 2020.

[10] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[11] Z. Liu, D. Shi, R. Li, W. Qin, Y. Zhang, and X. Ren, "Plc-vio: Visual–inertial odometry based on point-line constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1880–1897, 2022.

[12] J. Yin, D. Luo, F. Yan, and Y. Zhuang, "A novel lidar-assisted monocular visual slam framework for mobile robots in outdoor environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[13] J. Liu, X. Li, Y. Liu, and H. Chen, "Rgb-d inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9573–9580, 2022.

[14] J. Chang, N. Dong, and D. Li, "A real-time dynamic object segmentation framework for slam system in dynamic scenes," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[15] Y. Ren, B. Xu, C. L. Choi, and S. Leutenegger, "Visual-inertial multi-instance dynamic slam with object-level relocalisation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 055–11 062.

[16] P. Babin, P. Giguère, and F. Pomerleau, "Analysis of robust functions for registration algorithms," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1451–1457.

[17] N. Chebrolu, T. Läbe, O. Vysotska, J. Behley, and C. Stachniss, "Adaptive robust kernels for non-linear least squares problems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2240–2247, 2021.

[18] K. MacTavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers," in *2015 12th Conference on Computer and Robot Vision*, 2015, pp. 62–69.

[19] R. Zhou, L. He, H. Zhang, X. Lin, and Y. Guan, "Ndd: A 3d point cloud descriptor based on normal distribution for loop closure detection," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 1328–1335.

[20] Y. Fan, X. Du, L. Luo, and J. Shen, "Fresco: Frequency-domain scan context for lidar-based place recognition with translation and rotation invariance," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2022, pp. 576–583.

[21] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[22] N. Hatano, M. Ikeda, I. Ishikawa, and Y. Sawano, "Heaviside function as an activation function," *Journal of Applied Analysis*, vol. 29, no. 1, pp. 1–2, 2023.

[23] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[24] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International journal of computer vision*, vol. 19, no. 1, pp. 57–91, 1996.

[25] F. Shu, J. Wang, A. Pagani, and D. Stricker, "Structure plp-slam: Efficient sparse mapping and localization using point, line and plane for monocular, rgb-d and stereo cameras," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2105–2112.

[26] H. Lim, J. Jeon, and H. Myung, "Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1518–1525, 2022.

[27] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[28] H. Lim, Y. Kim, K. Jung, S. Hu, and H. Myung, "Avoiding degeneracy for monocular visual slam with point and line features," in *2021 IEEE*

International Conference on Robotics and Automation (ICRA), 2021, pp. 11 675–11 681.

[29] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang, "A survey on global lidar localization," *arXiv preprint arXiv:2302.07433*, 2023.

[30] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2095–2101.

[31] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "Lidar iris for loop-closure detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5769–5775.

[32] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4470–4479.

[33] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[34] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[35] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, "Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization," *Autonomous Robots*, pp. 1–21, 2022.

[36] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017, pp. 3226–3229.

[37] J. Zhang and S. Singh, "Laser–visual–inertial odometry and mapping with high robustness and low drift," *Journal of field robotics*, vol. 35, no. 8, pp. 1242–1264, 2018.

[38] W. Shao, S. Vijayarangan, C. Li, and G. Kantor, "Stereo visual inertial lidar simultaneous localization and mapping," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 370–377.

[39] S. Guo, Z. Rong, S. Wang, and Y. Wu, "A lidar slam with pca-based feature extraction and two-stage matching," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[40] B. D. Lucas, T. Kanade *et al.*, *An iterative image registration technique with an application to stereo vision*. Vancouver, 1981, vol. 81.

[41] H. Wei, F. Tang, C. Zhang, and Y. Wu, "Highly efficient line segment tracking with an imu-klt prediction and a convex geometric distance minimization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3999–4005.

[42] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

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

[44] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.

[45] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *European conference on computer vision*. Springer, 2004, pp. 224–237.

[46] D. Xu, J. Liu, Y. Liang, X. Lv, and J. Hyyppä, "A lidar-based single-shot global localization solution using a cross-section shape context descriptor," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 189, pp. 272–288, 2022.

[47] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[48] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[49] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[50] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2015.

[51] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2266–2273, 2022.

[52] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "Urbanloco: A full sensor suite dataset for mapping and localization in urban scenes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2310–2316.

[53] L. Hsu, W. Wen, W. Chen, Z. Liu, N. Kubo, T. Suzuki, and J. Meguro, "Urbannav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2021*, Sep. 2021, pp. 226–256.

[54] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Geometric-based line segment tracking for hdr stereo sequences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 69–74.

[55] C. Akinlar and C. Topal, "Edlines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1633–1642, 2011.

[56] B. Ferrarini, M. Waheed, S. Waheed, S. Ehsan, M. J. Milford, and K. D. McDonald-Maier, "Exploring performance bounds of visual place recognition using extended precision," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1688–1695, 2020.

[57] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4331–4339.

[58] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4390–4396.

[59] 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 International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

Xiongwei Zhao received the M.S. degree from University of Science and Technology Beijing, in 2021. Currently, he is working toward the Ph.D. degrees at School of electronic information, Harbin Institute of Technology (Shenzhen), Shenzhen. His research interests include state estimation with multi-sensor fusion, place recognition and multi-sensor calibration.

Congcong Wen (Member, IEEE) received the B.S. degree in Geographic Information System from China University of Petroleum, China, and the Ph.D. degree in Aerospace Information Research Institute, Chinese Academy of Sciences, China. He is currently a Postdoctoral Associate with the Department of Electrical and Computer Engineering at New York University Tandon School of Engineering and New York University Abu Dhabi. His research interests include 3D computer vision, robotics, and remote sensing.

Sai Manoj Prakhya received Ph.D from the School of Computer Science and Engineering, Nanyang Technological University, Singapore in 2017. He pursued his undergraduate studies in Electronics and Communication Engineering at Amrita University, India. He is currently working as a robotics researcher at Huawei Munich Research Center, Germany. He previously worked at autonomous driving firms, Desay SV Automotive and Aptiv as SLAM systems engineer and a medical device company, Stryker, as a Machine Vision Lead. His research interests include 3D mapping and localization, 3D point cloud processing, feature description and place recognition..

Hongpei Yin received the B.S. degree in Mechanical Engineering and Automation from North China University of Technology, Beijing, China, in 2018, and the Ph.D. degree in Mechatronics from Beijing Jiaotong University, in 2023. He is currently an engineer with Guangdong Institute of Artificial Intelligence and Advanced Computing, Guangzhou, China. His current research interests include robotics and intelligent systems, SLAM and State Estimation.

Rundong Zhou was born in Siping, Jilin, China in 1992. She received an M.S. degree in Precision instrument and measurement from Jilin University, Changchun, China in 2016. She is currently pursuing the Ph.D. degree with Harbin institute and technology. Her research interests include semantic image segmentation, simultaneous localization and mapping (SLAM), and object detection for USVs.

Yijiao Sun received the B.E., M.E. degree from the Harbin Institute of Technology,in 2019, 2021 respectively. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen. His research interests include sensor fusion, multi-sensor calibration and simultaneous localization and mapping.

Jie Xu received the B.E., M.S. degree from the Harbin Institute of Technology in 2018, 2020 respectively. He is currently pursuing the Ph.D. degree with the School of Mechanical and Electrical Engineering, Harbin Institute of Technology. His research interests include multi-sensor calibration, RGB-D SLAM and multi-modal SLAM.

Haojie Bai received the B.S. degree in electrical engineering with Zhengzhou University, Zhengzhou, China, in 2020. He is currently pursuing the Ph.D. degree in information and communication engineering with the School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, China. His main research interests include integration of communication and control, automated driving and muti-agent coordination.

Yang Wang received the Ph.D. degree in communication and information system from Harbin Institute of Technology, Harbin, China, in 2005. From 2005 to 2007, he was a Postdoctoral Fellow with Shenzhen Graduate School, Harbin Institute of Technology. He has been with the School of Electronic and Information Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, China, as an Associate Professor since 2007.

He is a senior member of Chinese Institute of Electronics. His research interests cover a wide range of topics in wireless communications and signal processing, including vehicular communications, MIMO systems, MIMO channel measurement and modelling, cooperative communications, and underwater wireless charging and communication systems.