

A Multi-model Fusion of LiDAR-inertial Odometry via Localization and Mapping

An Duy Nguyen¹, Chuong Phuoc Le¹, Pratik Walunj¹, Anton Netchaev²,
Thanh Nho Do³, and Hung Manh La¹

Abstract—This work presents a comprehensive LiDAR-inertial odometry framework featuring robust smoothing and mapping capabilities, effectively correcting LiDAR feature point skewness using an inertial measurement unit (IMU). While the Extended Kalman Filter (EKF) is a common choice for nonlinear motion estimation, its complexity grows when handling maneuvering targets. To overcome this challenge, a new framework that incorporates the Iterated Interactive Multiple Models of Kalman Filter (IMMKF) is given, providing a solution for reliable navigation in dynamic motion and noisy conditions. To ensure map consistency, an ikd-tree that facilitates continuous updates and adaptive rebalance is employed, preserving the map's integrity. To guarantee the robustness of our approach, it undergoes extensive testing across diverse scales of indoor and outdoor environments. This testing scenario simulates absolute GPS denial. In terms of estimated motion, the new algorithm demonstrates superior accuracy compared to existing approaches. The implementation is openly accessible on GitHub⁴ for further exploration.

I. INTRODUCTION

The integration of mobile robots into search, inspection, and rescue operations in unstructured environments has seen remarkable growth over recent decades. Conventional manual approaches in structural inspection tasks suffer from several limitations, including risks to workers, time inefficiency, high labor hours, and susceptibility to human errors. In this context, the precise estimation of a mobile robot's location becomes crucial [1], with a dense 3D map providing essential environmental insights, such as identifying free spaces and obstacles. This information is pivotal for effective path planning toward desired goals. Nevertheless, ground robots encounter challenges in environmental perception, especially in hazardous conditions characterized by a lack of GPS signals, absolute denial, and low visibility.

Light Detection and Ranging (LiDAR) stands out as one of the possible solutions to overcome these challenges, providing precise long-range estimation. However, the temporal

misalignment of laser points during scanning, influenced by dynamic models, introduces motion distortions, compromising localization and map registration. In traditional methods, the fusion of LiDAR and camera [2]–[4] data addresses the estimation problem and enhances visual localization. Inertial measurement units (IMUs) are often employed to enhance odometry estimation due to their higher frequency. Nevertheless, IMU measurements are prone to drift over time, necessitating the estimation of additional states, such as bias and extrinsic parameters.

Over the past decade, many lidar-based state estimation and mapping methods have been proposed, categorized into two strategies: loosely-coupled [5]–[9] and tightly-coupled [10]–[18] LiDAR-Inertial Odometry (LIO). In a loosely coupled LiDAR-inertial odometry method, LiDAR and inertial measurements are processed independently at first and then integrated at a later stage. In the LOAM [5], they first introduced the use of IMU data to de-skew LiDAR scans and provide a prior for scan-matching. Planar and edge feature points are then extracted from target clouds using an Iterated Closest Point (ICP) [19] for registration. The Lego-LOAM [9] adopts a framework akin to the LOAM, distinguishing ground points from the raw source and subsequently matching their correspondences through Euclidean clustering. A commonly adopted method for loosely-coupled fusion involves integrating measurements from lidar and IMU using variants of Extended Kalman Filters (EKF) [6], [7], [20] such as error-state EKF or a Multi-State Constraint Kalman Filter (MSCKF) for state estimation. In contrast, tightly-coupled LiDAR-inertial odometry methods typically integrate the LiDAR point cloud with IMU data, formulating 3D SLAM as a non-linear state estimation problem. The LIOM [10] stands as the first open-source offering implementation and dataset for tightly coupled LiDAR-inertial odometry. They introduce a slicing local window to generate a local map, and a rotation constraint is introduced to enhance the alignment of the local map with the global map. The LIO-SAM [11] presents a real-time method that is user-friendly, leveraging edge and planar features for scan mapping, akin to the LIOM. The incorporation of loop closures and GPS factors is optional, aiming to alleviate the accumulated drift in LiDAR inertial odometry over extended durations. The LINS [18] introduces a filter-based approach in tightly-coupled LiDAR pose optimization using the iterated Kalman Filter and a formula within the odometry. The FAST-LIO [12] extends the framework of the LINS to address linearization errors and decrease computational costs by introducing a

This work was funded by the U.S. National Science Foundation (NSF) under grants NSF-CAREER: 1846513 and NSF-PFI-TT: 1919127, and the U.S. Army's Engineer Research and Development Center under the contract: W912HZ24P0120. The views, opinions, findings, and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NSF and the U.S. Army.

¹ are with the Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA.

² is with the USACE Engineer Research and Development Center (ERDC), Information Technology Lab, Vicksburg, MS 39180, USA.

³ is with the Graduate School of Biomedical Engineering, UNSW Sydney, Kensington Campus, NSW, 2052, Australia.

⁴ <https://github.com/aralab-unr/IMM-LIO>

Corresponding author: Hung La hla@unr.edu

new formulation of the Kalman gain in state dimensions, rather than measurement dimensions. This eliminates the need for downsampling feature points in measurements, preventing information loss in the point clouds. The FAST-LIO2 [13] enhances the FAST-LIO framework with a new data structure, the *ikd-Tree*, which facilitates incremental map updates for dynamically inserted points. This improvement leads to better performance compared to state-of-the-art data structures for *kNN*. We introduce a tightly-coupled Iterated Interactive Multiple Models of the Kalman Filter (IMMKF), and the LiDAR-inertial odometry (LIO), called IMMKF-LIO, to enhance the accuracy of robot displacement estimation. The main contributions of our work can be summarized as follows:

- To the best of our knowledge, our paper introduces the first framework enabling the fusion of multiple models for LiDAR-inertial odometry.
- We introduce a streamlined prediction pipeline that reduces computational costs while preserving the accuracy of state estimation.
- The algorithm undergoes verification through various indoor and outdoor tests across various scales and environments, demonstrating superior performance compared to state-of-the-art LiDAR-inertial algorithms
- The code and dataset are open-sourced.

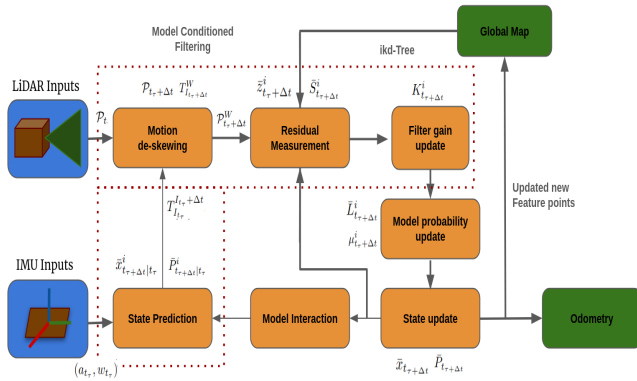


Fig. 1: The system workflow of IMMKF.

II. METHODOLOGY

A. System Overview

The workflow of the IMMKF-LIO is shown in Fig. 1. The LiDAR raw points are initially accumulated and then passed through the feature extraction module to obtain planar and edge features. The feature points and IMU measurements are simultaneously input into the IMMKF for map point registration. Global map points within the extensive local map are structured using an incremental *k-d* tree, denoted as *ikd-Tree*. This updated map is then utilized to map new feature points in the subsequent timestamp.

Assuming the IMU frame conveniently aligns with the body frame, and the extrinsic transformation between the LiDAR frame (L) and the IMU frame (I), $T_L^I = (R_L^I, p_L^I)$ where $R_L^I \in SO(3)$ and $p_L^I \in \mathbb{R}^3$ are the rotation matrix

and translation vector of T_L^I , is known. During initialization, the first IMU frame is aligned as the world frame (W). The system state x can be written as follow:

$$x = [p_I^{WT}, v_I^{WT}, \varphi_I^{WT}, a_I^{WT}, \omega_I^{WT}]^T, \quad (1)$$

where $p_I^W = [p_x, p_y, p_z]^T$, $v_I^W = [v_x, v_y, v_z]^T$, $a_I^W = [a_x, a_y, a_z]^T$, $\omega_I^W = [\omega_x, \omega_y, \omega_z]^T$, and $\varphi_I^W = [\phi, \theta, \psi]^T$ are the position, velocity, acceleration, angular velocity and orientation with the body frame to the world frame.

B. Iterated Interactive Multiple Models of Kalman Filter

The concept of the IMMKF can be outlined in the following steps: Model Interaction, Model Conditioned Filtering, Model Probability Update, and State Estimation.

1) *Model Interaction*: The process involves predicting the mode probability of an active Kalman Filter within the IMMKF framework from the last timestamp t_{k-1} to the current timestamp t_k in response to a LiDAR scan. The interaction and mixing probability can be predicted as follows:

$$\mu_{t_k|t_{k-1}}^i = \sum_j \pi_{ij} \mu_{t_{k-1}}^j, \quad (2)$$

$$\mu_{t_{k-1}}^{j|i} = \frac{\pi_{ij} \mu_{t_{k-1}}^j}{\mu_{t_k|t_{k-1}}^i}, \quad (3)$$

where $(\forall i, j \in M)$, and M is the number of models in the IMMKF. For example, we use three models: constant-velocity (CV), constant-acceleration (CA), and constant-turning rate (CT) in this paper, then $M = 3$. Initially, the model probability shares a uniform distribution as $\mu = [\frac{1}{M}, \dots, \frac{1}{M}]^T \in \mathbb{R}^{M \times 1}$. $\pi_{ij} \in \mathbb{R}^{M \times M}$ presents to the transition probability matrix to shift mode j to mode i .

Assuming the optimal state estimate after fusion at the last LiDAR scan t_{k-1} is represented by $\bar{x}_{t_{k-1}}$, with the corresponding covariance matrix denoted as $\bar{P}_{t_{k-1}}$. The mixing estimates $\hat{x}_{t_{k-1}}^i$ and mixing covariances $\hat{P}_{t_{k-1}}^i$ are combined by each filter at time t_{k-1} using the mixing interaction probability:

$$\hat{x}_{t_{k-1}}^i = \sum_j \bar{x}_{t_{k-1}}^j \mu_{t_{k-1}}^{j|i}, \quad (4)$$

$$\hat{P}_{t_{k-1}}^i = \sum_j [\bar{P}_{t_{k-1}}^j + (\hat{x}_{t_{k-1}}^i - \bar{x}_{t_{k-1}}^j)(\hat{x}_{t_{k-1}}^i - \bar{x}_{t_{k-1}}^j)^T] \mu_{t_{k-1}}^{j|i}. \quad (5)$$

2) *Model Conditioned Filtering*: The estimate prediction is carried out upon the arrival of an IMU measurement. IMUs typically provide a higher rate than LiDAR; hence, the sampling time corresponding to an IMU arrival is denoted as $t_\tau \in [t_{k-1}, t_k)$ $\Delta t = t_\tau - t_{\tau-1}$. Each model i of the IMMKF is linked to a Kalman filter, which is described by the following prediction:

$$\bar{x}_{t_\tau+\Delta t|t_\tau}^i = F_{t_\tau}^i \hat{x}_{t_\tau}^i + G_{t_\tau}^i \delta_{t_\tau}^i, \quad (6)$$

$$\bar{P}_{t_\tau+\Delta t|t_\tau}^i = F_{t_\tau}^i \hat{P}_{t_\tau}^i F_{t_\tau}^{iT} + G_{t_\tau}^i Q_{t_\tau}^i G_{t_\tau}^{iT}. \quad (7)$$

Here, F^i represents the transition matrix of model i in the IMMKF. In this paper, F^i is equal to F_{cv} , F_{ca} , and F_{ct} , respectively. $G_{t_\tau}^i$ is the control input matrix, and $Q_{t_\tau}^i$ is the matrix representing the system noise. The raw LiDAR

points $\mathcal{P}_{t_\tau}$, introduced with motion distortion under different sampling timestamps, are compensated by projecting all points $\mathcal{P}_{t_\tau}$ to $\mathcal{L}_{t_\tau}$ by $T_{I_{t_\tau}}^{I_{t_\tau+\Delta t}}$, obtaining the relative motion of the IMU frame through the predicted state:

$$\mathcal{P}_{t_\tau+\Delta t} = T_L^{I^T} T_{I_{t_\tau}}^{I_{t_\tau+\Delta t}} T_L^I \mathcal{P}_{t_\tau}. \quad (8)$$

Subsequently, the projected point $\mathcal{P}_{t_\tau+\Delta t}$ is utilized to construct a residual measurement. In existing filter-based LiDAR-inertial odometry [12] [14] [21], a common strategy is the adoption of nearest correspondence matching, with the point-to-plane distance calculated as the residual. This approach entails projecting the feature point with motion compensation onto the world frame:

$$\mathcal{P}_{t_\tau+\Delta t}^W = T_{I_{t_\tau+\Delta t}}^W T_L^I \mathcal{P}_{t_\tau+\Delta t}. \quad (9)$$

It's important to note that the transformation matrix T_I^W is included in the state vector x . Let $u_{t_\tau+\Delta t}$ denote the normal vector of the corresponding plane, obtained by searching the five nearest points in the map $\mathcal{P}_{t_\tau+\Delta t}^W$ represented by the *ikd-Tree*, where the feature point $q_{t_k}^W$ is centroid.

$$\tilde{z}_{t_\tau+\Delta t}^i = u_{t_\tau+\Delta t}^T (\mathcal{P}_{t_\tau+\Delta t}^W - q_{t_k}^W) \approx -H^i \tilde{x}_{t_\tau+\Delta t|t_\tau}^i - v_{t_k}, \quad (10)$$

$$\bar{S}_{t_\tau+\Delta t}^i = H^i \bar{P}_{t_\tau+\Delta t|t_\tau}^i H^{iT} + R_{t_{k-1}}, \quad (11)$$

where $\tilde{x}_{t_\tau+\Delta t|t_\tau}^i = \bar{x}_{t_\tau+\Delta t|t_\tau}^i - \bar{x}_{t_k}$, v_{t_k} is the measurement noise, and H^i is the measurement matrix of each model i . R is the covariance matrix of measurements with $R \in \mathbb{R}^{m \times m}$ (m numbers of the feature points). We offer detailed formulations of the involved transition F^i and measurement H^i matrices in the supplementary material [22].

We utilize the formulation outlined in [12] to compute the Kalman Filter gain. This approach, proven effective [23], provides a solution in the state dimension rather than the measurement dimension, resulting in reduced computation costs:

$$K_{t_\tau+\Delta t}^i = (H^{iT} R^{-1} H^i + P_{t_\tau+\Delta t|t_\tau}^{i-1})^{-1} H^{iT} R_{t_k}^{-1}. \quad (12)$$

The updated estimation and covariance are calculated as follows:

$$\bar{x}_{t_\tau+\Delta t|t_\tau+\Delta t}^i = \bar{x}_{t_\tau+\Delta t|t_\tau}^i + K_{t_\tau+\Delta t}^i \tilde{z}_{t_\tau+\Delta t}^i, \quad (13)$$

$$\bar{P}_{t_\tau+\Delta t|t_\tau+\Delta t}^i = (I - K_{t_\tau+\Delta t}^i H^i) \bar{P}_{t_\tau+\Delta t|t_\tau}^i. \quad (14)$$

3) Model Probability Update: Estimating the likelihood of each model involves evaluating the distribution of the residual error $\tilde{z}_{t_\tau+\Delta t}^i$, the difference between current measurements and estimated states, relative to $\bar{S}_{t_\tau+\Delta t}^i$ the residual covariance:

$$\bar{L}_{t_\tau+\Delta t}^i = \frac{1}{\sqrt{2\pi|\det(\bar{S}_{t_\tau+\Delta t}^i)|}} \exp\left(-\frac{1}{2} \tilde{z}_{t_\tau+\Delta t}^{iT} \bar{S}_{t_\tau+\Delta t}^{i-1} \tilde{z}_{t_\tau+\Delta t}^i\right). \quad (15)$$

The likelihood is assessed at zero, then $\bar{L}_{t_\tau+\Delta t}^i \stackrel{\text{assume}}{=} \mathcal{N}(\tilde{z}_{t_\tau+\Delta t}^i; 0, \bar{S}_{t_\tau+\Delta t}^i)$. The mode probability update assesses the likelihood of each model i relative to the cumulative probabilities of all potential models:

$$\mu_{t_\tau+\Delta t}^i = \frac{\mu_{t_\tau+\Delta t|t_\tau}^i \bar{L}_{t_\tau+\Delta t}^i}{\sum_j \mu_{t_\tau+\Delta t|t_\tau}^j \bar{L}_{t_\tau+\Delta t}^j}. \quad (16)$$

4) State Estimation: In conclusion, the overall estimation and covariance are updated based on the new mode probability and the current estimated states:

$$\bar{x}_{t_\tau+\Delta t} = \sum_i \bar{x}_{t_\tau+\Delta t|t_\tau+\Delta t}^i \mu_{t_\tau+\Delta t}^i, \quad (17)$$

$$\bar{P}_{t_\tau+\Delta t} = \sum_i [\bar{P}_{t_\tau+\Delta t|t_\tau+\Delta t}^i + \tilde{x}_{t_\tau+\Delta t} \tilde{x}_{t_\tau+\Delta t}^T] \mu_{t_\tau+\Delta t}^i, \quad (18)$$

where $\tilde{x}_{t_\tau+\Delta t} = \bar{x}_{t_\tau+\Delta t} - \bar{x}_{t_\tau+\Delta t|t_\tau}^i$. The estimated state $\bar{x}_{t_\tau+\Delta t}$ and covariance $\bar{P}_{t_\tau+\Delta t}$ will serve as inputs for the next IMU measurement arrival. The optimal estimation at $t_\tau+\Delta t$, represented as $\bar{x}_{t_\tau+\Delta t}$, is attained if and only if the projected point $\mathcal{P}_{t_\tau+\Delta t}^W$, corresponding to $\bar{x}_{t_\tau+\Delta t}$, is accurately mapped to the true correspondence in the world frame satisfies the condition $(\|\tilde{z}_{t_{k-1}}^{t_\tau+\Delta t} + \sum_{t_{k-1}}^{t_\tau+\Delta t} H \tilde{x}_{t_{k-1}}^{t_\tau+\Delta t}\|_{R_{t_{k-1}}}^2) < \psi$. The optimal estimation in the current LiDAR $\bar{x}_{t_k}$ scan is obtained by minimizing the difference between the residual measurement and the residual mapped from the state space to the observation space:

$$\operatorname{argmin}_{\bar{x}_{t_{k-1}}^{t_k}} (\|\tilde{z}_{t_{k-1}}^{t_k} + \sum_{t_{k-1}}^{t_k} H \tilde{x}_{t_{k-1}}^{t_k}\|_{R_{t_{k-1}}}^2). \quad (19)$$

C. Global map update

With the optimal estimation $\bar{x}_{t_k}$, the transformation $\bar{T}_{I_{t_k}}^W = [\bar{R}_{I_{t_k}}^W, p_{I_{t_k}}^W]$ ($\bar{R}_{I_{t_k}}^W$ the matrix represents a rotation whose roll, pitch, and yaw $\varphi_{I_{t_k}}^W$ angles) of each feature point to the global frame is performed:

$$\mathcal{P}_{t_k}^W = \bar{T}_{I_{t_k}}^W T_L^I \mathcal{P}_{t_k}. \quad (20)$$

For initialization, the first LiDAR scan is considered as the global frame.

III. EXPERIMENTS

We elaborate on a set of experiments conducted in both indoor and outdoor environments, designed to provide both qualitative and quantitative insights into our proposed framework. All experiments are performed using a computer equipped with Intel Core i7-8700 CPU 3.20Hz and 32GB RAM using the robot operating system (ROS-Noetic) [24] in Ubuntu 20.04 Linux.

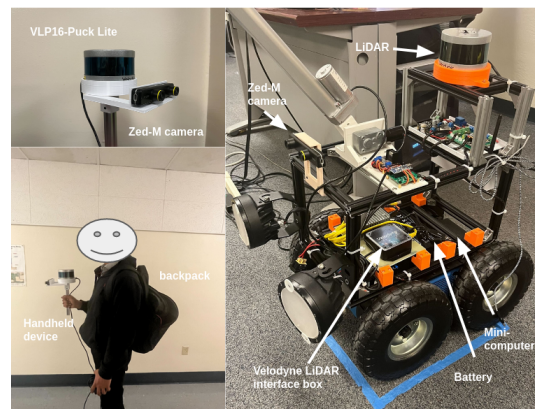


Fig. 2: Datasets from the University of Nevada, Reno (UNR) campus were collected using a custom handheld device and an unmanned ground robot.

A. Quantitative Analysis

In this section, we assess the proposed framework in terms of accuracy and computational complexity compared to state-of-the-art LiDAR-inertial odometry algorithms. To ensure a fair comparison, we utilize open-source datasets from [25], and [26]. These extensive datasets [25] were collected at New College, Oxford, using an OS1-64 LiDAR with a scan rate of 10Hz, coupled with a built-in 6-axis IMU sampled at 100Hz. These datasets are referred to as "*nse_*" and "*nle_*", corresponding to "*01_short_experiment*" and "*02_long_experiment*", respectively, offering different trajectories and time durations. Additionally, [26] provides a localization dataset gathered in Urban Canyons, characterized by high-rising buildings and numerous dynamic objects. This dataset was collected using an HDL-32E Velodyne LiDAR with an 80m range at 10Hz, coupled with a 9-axis Xsens Mti IMU operating at 400Hz, referred to "*ubhk_*". As recommended by the dataset authors, all datasets provide accurate ground truth trajectories and well-calibrated data from multiple sensors.

	Name	Duration (min:sec)	Trajectory Length (meters)
<i>ubhk_01</i>	UrbanNav-HK_TST-20210517_sensors	13:05	3640
<i>ubhk_02</i>	UrbanNav-HK_Whampoa-20210521_sensors	25:36	4510
<i>nle_01</i>	rooster_2020-03-10-12-15-49_14	2:47	193
<i>nle_02</i>	rooster_2020-03-10-12-13-02_13	2:47	195
<i>nle_03</i>	rooster_2020-03-10-12-10-15_12	2:47	195
<i>nle_04</i>	rooster_2020-03-10-12-18-36_15	2:47	190
<i>nse_01</i>	rooster_2020-03-10-10-47-39_4	2:47	160
<i>nse_02</i>	rooster_2020-03-10-10-44-52_3	2:47	175
<i>nse_03</i>	rooster_2020-03-10-10-42-05_2	2:47	168
<i>nse_04</i>	rooster_2020-03-10-10-39-18_1	2:47	173

TABLE I: Details of all the data sequences.

1) *Evaluation of Odometry Accuracy*: We assess odometry accuracy through absolute RMSE in translation and rotation errors, comparing our results with advanced algorithms: the FAST-LIO2 [13], DLO [27], and LIO-SAM [11]. For unbiased results, the GPS and loop closure factors are disabled in all frameworks. As the LIO-SAM necessitates a 9-axis IMU for attitude data, it is not applicable to the New College dataset. The LIO-SAM requires precise manual calibration and initial bias estimation for optimal performance. Despite our efforts to provide the closest possible calibration parameters for comparison, achieving the finest performance with LIO-SAM remains challenging.

A variant of the proposed system, IMMKF-va, is tested by removing the constant-turn (CT) model, compensating for sensor movement when facing a large change in sensor heading. As shown in Table II, IMMKF-va still produces

reliable results (beating the LIO-SAM and the DLO) for the '*ubhk_01*', and '*ubhk_02*' sequence when the car mostly drives in a straight line with constant velocity or accelerates with a slight change in acceleration. However, its performance degrades compared to the original framework for all sequences of the New College dataset, where the sensor undergoes frequent turns. The LIO-SAM and DLO exhibit significant drift in extended duration and distance data '*ubhk_01*', and '*ubhk_02*'. When LIO-SAM encounters failure in graph optimization without GPS factors and loop closures, it is unable to reset or compensate for the growing IMU noise. On the other hand, the accuracy of DLO relies on keyframe selection and adaptive thresholds for mapping with GICP. In cases of motion drift, obtaining an optimal instantaneous guess for the correct correspondence and keyframe becomes challenging. Table II illustrates that IMMKF consistently produces optimal results across various sequences chosen for this analysis. A key contributing factor to this success is the IMMKF's ability to mitigate IMU noise-induced drift through the use of multiple filters. Each filter incorporates individual estimations for state and covariance in different motions, mitigating dependence on raw IMU data for motion compensation. Fig. 3 illustrates the mapping results of all algorithms compared in Table II. The first image depicts the ground truth trajectory aligned with a Google map. The DLO exhibits drift after 345 seconds of running, as indicated by the red part in the image, where only the ground is detected. The LIO-SAM displays significant drift in both position and orientation, as seen in the white box where the lane is split into two. The mapping results of the FAST-LIO2 and our IMMKF appear similar. For a more detailed comparison, we will delve into qualitative results to assess performance on a smaller scale.

2) *Evaluation of Computational Complexity*: To assess computational efficiency, we compare the IMMKF to the FAST-LIO2 [13], utilizing their built *ikd_Tree* for our framework. The primary difference between these two approaches lies in the filter-based method. Our analysis focuses on examining the impact of computational complexity on *ikd_Tree* construction, specifically on incremental updates, *ikd_Tree* size, and pre-processing time shown in Table III. Incremental updates involve efficiently integrating new points into the existing tree structure without rebuilding the entire tree. Pre-process time represents the consumption of the process of accumulating feature points at a certain time into a LiDAR scan, which remains consistently around *1.10 ms*. On average, the mapping procedure in the IMMKF achieves a 10% reduction in computational time compared to [13]. The primary reason for this improvement is that the IMMKF does not require Jacobian matrix calculations, unlike the FAST-LIO2, which needs to repeatedly compute them for estimation using EKF. This complexity increases as the number of feature points grows. We attain faster convergence with fewer nodes in the *ikd_Tree* when utilizing the *nse_* and *nle_* datasets.

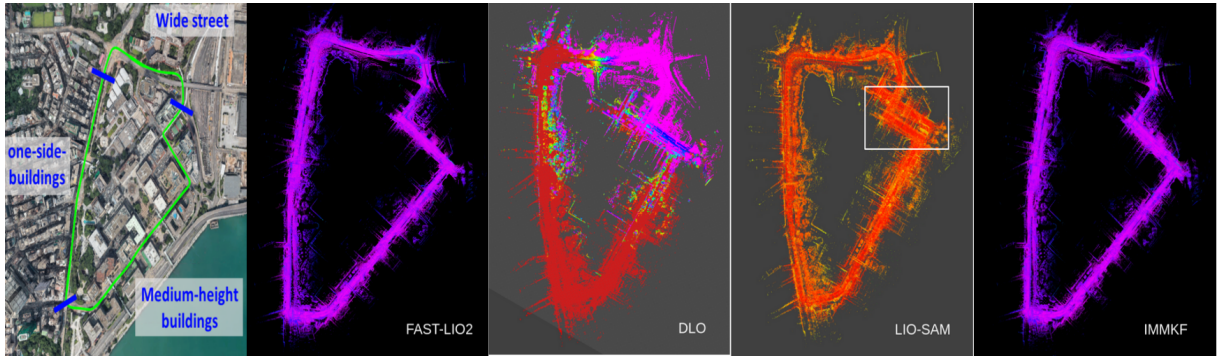


Fig. 3: The mapping results of the FAST-LIO2, the DLO, the LIO-SAM, and the IMMKF in bird's eye view in a typical urban canyon of HongKong 'ubhk_01'. The first map is the trajectory ground truth aligning with a Google map. The mapping results of our framework and the FAST-LIO2 are quite similar in a large-scale environment. Additionally, in the DLO map, the red color represents the ground, while other colors indicate key features. The white box in the LIO-SAM shows the drift of the lane, which is split into two.

	<i>ubhk_01</i>	<i>ubhk_02</i>	<i>nle_01</i>	<i>nle_02</i>	<i>nle_03</i>	<i>nle_04</i>	<i>nse_01</i>	<i>nse_02</i>	<i>nse_03</i>
FAST-LIO2	12.58/0.053	14.56/0.050	0.48/0.046	0.16/0.029	0.31/0.022	0.65/0.047	0.68/0.021	1.43/0.046	0.97/ 0.049
DLO	16.18/0.166	17.24/0.178	1.02/0.039	0.54/0.043	1.49/0.048	1.55/0.089	0.47/0.026	1.74/0.038	1.03/0.057
LIO-SAM	19.49/0.209	16.79/0.184	-	-	-	-	-	-	-
IMMKF-va	13.24/0.055	14.32/0.067	0.41/0.049	0.29/0.045	0.29/0.039	1.28/0.061	0.74/0.065	1.37/0.054	0.83/0.054
IMMKF	10.37/0.044	12.04/0.047	0.24/0.035	0.27/0.037	0.26/0.018	1.02/ 0.033	0.33/0.017	1.07/0.035	0.57/0.055

TABLE II: The absolute translation RMSE [m] and rotation errors RMSE [rad] with precise ground-truth. Since the datasets 'nse_' and 'nle_' captured by a 6-axis IMU do not include attitude quaternion data, LIO-SAM, which relies on a 9-axis IMU, is incompatible with all sequences denoted as "-".

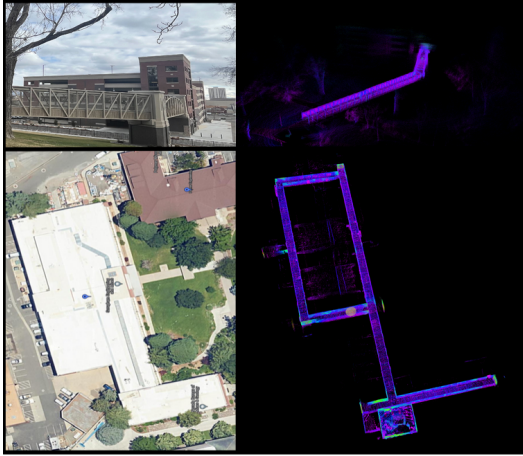


Fig. 4: The mapping results were collected using handheld devices and an unmanned ground robot on the UNR campus.

B. Qualitative Analysis

In this section, we evaluate the real-time performance of the IMMKF using a handheld platform shown in Fig. 2, equipped with a VLP16-PuckLITE LiDAR and a Zed-mini stereo camera. The real-time map generated by the IMMKF is showcased in Fig. 4, illustrating a bridge that poses challenges due to the lack of distinctive features in the bridge joints. The second scenario in Fig. 3 captures the indoor hall of the SEM building at UNR, covering a total traveling distance of approximately 200m, with a drift of approximately 0.12m observed in an end-to-end evaluation.

As mentioned earlier, we conducted a comparison between the IMMKF and the Fast-LIO2 on a larger scale

	Incremental update [ms]		<i>ikd-Tree</i> Size [num]		Pre-process time [ms]	
	ours	[13]	ours	[13]	ours	[13]
<i>nse_01</i>	2.503	2.532	213384	215817	1.187	1.317
<i>nse_02</i>	1.581	1.712	88244	88840	1.170	1.192
<i>nse_03</i>	2.327	2.362	142730	142807	1.103	1.118
<i>nse_04</i>	4.310	5.016	60019	70328	1.101	1.273
<i>nle_01</i>	2.501	2.573	196410	197506	1.142	1.326
<i>nle_02</i>	2.708	2.754	269891	270773	1.129	1.155
<i>nle_03</i>	3.838	3.947	269891	270773	1.129	1.155
<i>nle_04</i>	1.717	1.825	79317	79668	1.104	1.136

TABLE III: Average time consumption of the mapping procedures per scan using the *ikd-Tree*.

by running the New College Dataset [25]. The Fast-LIO2 encounters a challenge in obtaining a reliable initial guess in a campus scenario where the platform may not stay static long enough to acquire a dense point cloud. Instead of running all sequences in the New College dataset, we focused on an individual sequence where the movement was already dynamic and LiDAR scans were captured in campus areas with open fields, walls, and trees, resulting in fewer meaningful feature points. The random walk and non-optimal initialization estimation contribute to the degradation of the EKF filter-based in the FAST-LIO2, impacting its convergence and accuracy. It is clear that the FAST-LIO2 exhibits drift along the side of the 3D map, as depicted on a larger scale within the white box in Fig. 5. This highlights the benefits of the IMMKF filter in providing smoother and more accurate values.

IV. CONCLUSION

This paper introduces the IMMKF-LIO, a framework for tightly-coupled LiDAR-inertial odometry through localiza-

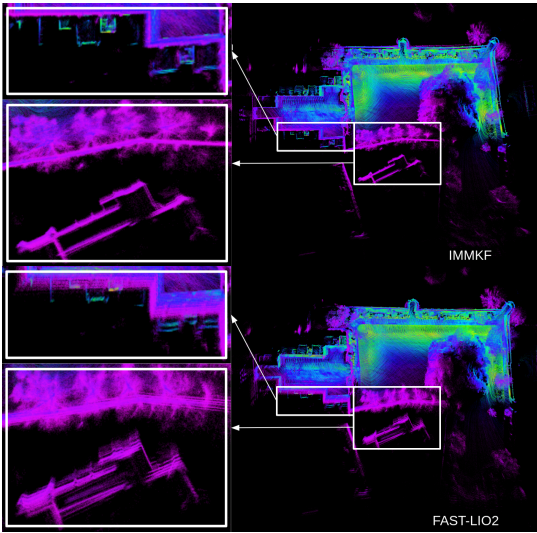


Fig. 5: The mapping results of the IMMKE, and the FAST-LIO2 in bird's eye view using the New College Dataset [25]. Zooming out from the white box in the mapping results, it is evident that our results exhibit less drift than the FAST-LIO2.

tion and mapping, demonstrating computational efficiency compared to the current state-of-the-art LIO algorithm. The key innovation lies in its support for various estimation models to compensate for LiDAR distortion and ensure consistent mapping. The implemented results demonstrate that the IMMKE exhibits, on average, a 10% reduction in computational complexity compared to other common non-linear models, such as the EKF. The results indicate that the IMMKE achieves similar or better accuracy with lower computation compared to the FAST-LIO2 [13]. However, our work still has some shortcomings. In certain indoor scenes where feature scarcity is common due to aggressive altitude changes, the LIO system may fail to extract enough features to match the global map. To address this issue, our future work will focus on fusing additional sensors, such as millimeter-wave radar, to enhance feature detection and improve system robustness.

REFERENCES

- [1] A. Singandhupe and H. M. La, "A review of slam techniques and security in autonomous driving," in *2019 third IEEE international conference on robotic computing (IRC)*. IEEE, 2019, pp. 602–607.
- [2] A. Singandhupe, H. M. La, and Q. P. Ha, "Single frame lidar-camera calibration using registration of 3d planes," in *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2022, pp. 395–402.
- [3] A. Sehgal, A. Singandhupe, H. M. La, A. Tavakkoli, and S. J. Louis, "Lidar-monocular visual odometry with genetic algorithm for parameter optimization," in *International Symposium on Visual Computing*. Springer, 2019, pp. 358–370.
- [4] A. Singandhupe and H. M. La, "Single frame lidar and stereo camera calibration using registration of 3d planes," in *2021 Fifth IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2021, pp. 115–118.
- [5] J. Zhang, S. Singh *et al.*, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [6] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 6240–6245.
- [7] Y. Balazadegan Sarvrood, S. Hosseinyalamdary, and Y. Gao, "Visual-lidar odometry aided by reduced imu," *ISPRS international journal of geo-information*, vol. 5, no. 1, p. 3, 2016.
- [8] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 3126–3131.
- [9] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.
- [10] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3144–3150.
- [11] 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)*. IEEE, 2020, pp. 5135–5142.
- [12] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.
- [13] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [14] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 8899–8906.
- [15] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.
- [16] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis, "A laser-aided inertial navigation system (l-ins) for human localization in unknown indoor environments," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5376–5382.
- [17] Z. Cheng, D. Liu, Y. Yang, T. Ling, X. Chen, L. Zhang, J. Bai, Y. Shen, L. Miao, and W. Huang, "Practical phase unwrapping of interferometric fringes based on unscented kalman filter technique," *Optics express*, vol. 23, no. 25, pp. 32 337–32 349, 2015.
- [18] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 8899–8906.
- [19] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.
- [20] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidar-inertial-camera odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5848–5854.
- [21] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.
- [22] R. ARA Lab, University of Nevada, "Imm lio project," <https://github.com/ara-lab-unr/IMM.LIO>, 2024, accessed: 2024-09-07.
- [23] N. J. Higham, *Accuracy and stability of numerical algorithms*. SIAM, 2002.
- [24] 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.
- [25] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld lidar, inertial and vision with ground truth," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4353–4360.
- [26] L.-T. Hsu, F. Huang, H.-F. Ng, G. Zhang, Y. Zhong, X. Bai, and W. Wen, "Hong kong urbannav: An open-source multisensory dataset for benchmarking urban navigation algorithms," *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 4, 2023.
- [27] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022.