







LIMOT: A Tightly-Coupled System for LiDAR-Inertial Odometry and Multi-Object Tracking

Zhongyang Zhu , Junqiao Zhao , *Member, IEEE*, Kai Huang , *Graduate Student Member, IEEE*, Xuebo Tian , Jiaye Lin , and Chen Ye , *Member, IEEE*

Abstract—Simultaneous localization and mapping (SLAM) is essential for autonomous driving. Most LiDAR-inertial SLAM algorithms assume a static environment, leading to unreliable localization in dynamic environments. Moreover, the accurate tracking of moving objects is of great significance for the control and planning of autonomous vehicles. This letter proposes LIMOT, a tightly-coupled multi-object tracking and LiDAR-inertial odometry system that is capable of accurately estimating the poses of both ego-vehicle and objects. Based on the historical trajectories of tracked objects in a sliding window, we perform robust data association. We propose a trajectory-based dynamic feature filtering method, which leverages tracking results to filter out features belonging to moving objects before scan-matching. Factor graph-based optimization is then conducted to optimize the bias of the IMU and the poses of both the ego-vehicle and surrounding objects in a sliding window. Experiments conducted on the KITTI tracking dataset and self-collected dataset show that our method achieves better pose and tracking accuracy than our previous work DL-SLOT and other baseline methods.

Index Terms—SLAM, LiDAR-inertial odometry, multi-object tracking.

I. INTRODUCTION

SIMULTANEOUS localization and mapping (SLAM) is essential for autonomous vehicles, especially in Global Navigation Satellite System (GNSS)-denied environments. In previous studies, LiDAR-inertial SLAM has been shown to achieve great success [1], [2], [3], [4], [5]. Nevertheless, most SLAM

systems, including LiDAR-inertial SLAM, degrade in dynamic environments because they rely heavily on the assumption of a static environment, which is usually invalidated in reality. Therefore, it is crucial for SLAM to use reasonable approaches to deal with dynamic objects in real-world applications. At the same time, multi-object tracking in complex dynamic scenes is crucial for the decision-making and planning of autonomous vehicles. Combining SLAM and multi-object tracking becomes an important task and many related approaches have emerged in recent years [6], [7], [8].

The vast majority of methods perform multi-object tracking and SLAM separately, i.e., loosely-coupled. This results in that the tracking accuracy highly depends on ego-pose estimation which is, however, not reliable in dynamic environments. Recently, tightly-coupled multi-object tracking and vision-based SLAM systems have gained extensive attention [6], [9], [10], [11]. These systems generally construct a unified optimization framework, in which the poses of both the ego-vehicle and moving objects are jointly optimized. However, their performance is nonetheless limited by inaccurate 3D object detection and sensitivity to illumination change and rapid motion.

In a previous study, we developed DL-SLOT [7], a tightly-coupled LiDAR SLAM and multi-object tracking system; although effective, it is still subject to the limitations of the single LiDAR sensor described above. Therefore, it is necessary to develop the tight coupling of multi-object tracking and LiDAR-inertial SLAM. In this letter, we propose LIMOT, a tightly-coupled multi-object tracking and LiDAR-inertial odometry system capable of accurately estimating the poses of both the ego-vehicle and surrounding objects. First, all movable objects are represented by 3D bounding boxes generated by an object detector. Simultaneously, inertial measurement unit (IMU) pre-integration is utilized to de-skew LiDAR scans and provide an initial guess for scan-matching. Then, similar to DL-SLOT [7], a combination of trajectory approximation of tracked objects in a sliding window and the successive shortest path algorithm [12] is employed to perform data association. Object states (stationary or dynamic) can then be determined through tracking results.

DL-SLOT filters out all feature points belonging to movable objects and suffers from feature sparsity when many movable objects are actually static. However, based on the approximated object trajectories and the estimated motion from IMU pre-integration, feature points belonging to moving objects can be precisely filtered out before scan-matching in LIMOT. Finally,

Manuscript received 20 March 2024; accepted 13 May 2024. Date of publication 24 May 2024; date of current version 10 June 2024. This letter was recommended for publication by Associate Editor J. Zhang and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported by the National Key Research and Development Program of China under Grant 2021YFB2501104. (*Corresponding author: Junqiao Zhao.*)

Zhongyang Zhu, Xuebo Tian, Jiaye Lin, and Chen Ye are with the Department of Computer Science and Technology, School of Electronics and Information Engineering, MOE Key Lab of Embedded System and Service Computing, Tongji University, Shanghai 200070, China (e-mail: 2233057@tongji.edu.cn; 1930773@tongji.edu.cn; 2310920@tongji.edu.cn; yechen@tongji.edu.cn).

Junqiao Zhao is with the Department of Computer Science and Technology, School of Electronics and Information Engineering, MOE Key Lab of Embedded System and Service Computing, Tongji University, Shanghai 200070, China, and also with the Institute of Intelligent Vehicles, Tongji University, Shanghai 200070, China (e-mail: zhaojunqiao@tongji.edu.cn).

Kai Huang is with the School of Surveying and Geo-Informatics, Tongji University, Shanghai 200070, China (e-mail: 1911202@tongji.edu.cn).

Our open-source implementation is available at <https://github.com/tiev-tongji/LIMOT>.

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3405385>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3405385

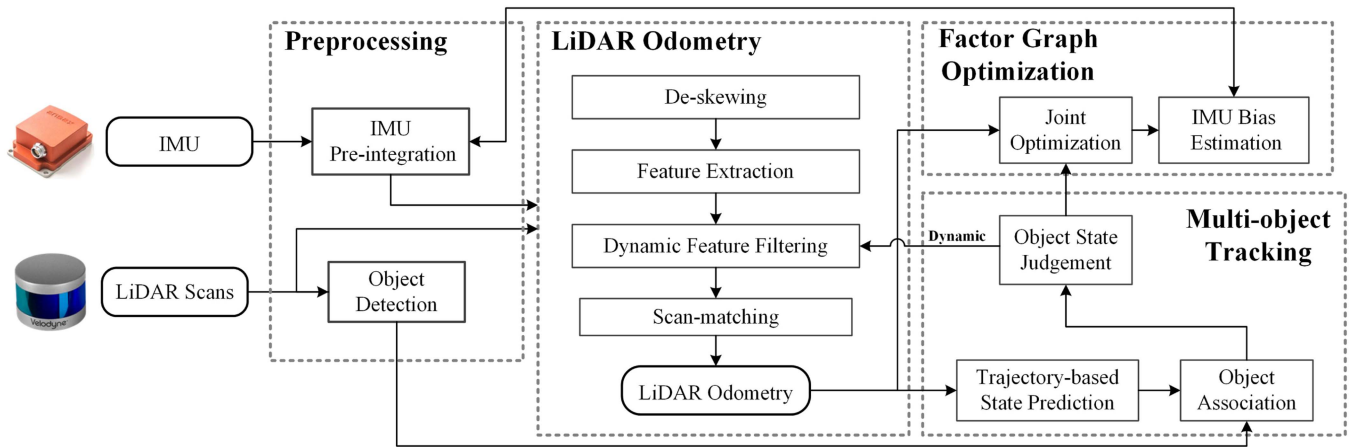


Fig. 1. System architecture of LIMOT. The system consists of the preprocessing, LiDAR odometry, multi-object tracking, and factor graph optimization modules.

a factor graph optimization framework is conducted to optimize the bias of the IMU and the poses of both the ego-vehicle and objects in a sliding window. The main contributions of this work can be summarized as follows:

- A tightly-coupled multi-object tracking and LiDAR-inertial odometry system capable of jointly estimating the poses of the ego-vehicle and surrounding objects.
- A dynamic feature filtering method that uses the approximated object trajectories to identify and exclude feature points belonging to moving objects, while still utilizing feature points on static movable objects to provide constraints for scan-matching.
- Extensive experiments on different datasets demonstrate the advantages of our system compared to other methods. To benefit the community, our implementation of this work is open-source at <https://github.com/tiev-tongji/LIMOT>.

II. RELATED WORKS

A. Lidar-Inertial Slam

Schemes relating to the fusion of LiDAR and IMU measurements can be classified into two categories: loosely-coupled fusion [1], [2], [13], [14], [15] and tightly-coupled fusion [3], [4], [5], [16], [17]. In LOAM [1] and LeGO-LOAM [2], IMU measurements are used to de-skew LiDAR scans and provide a motion prior to scan registration. In [13], [14], [15], the extended Kalman filter is employed to loosely fuse LiDAR, IMU, and optionally GNSS for robot pose estimation. However, the IMU bias is not optimized in these loosely-coupled algorithms, thereby reducing their precision. LINS [16] proposed a tightly-coupled LiDAR-inertial odometry method based on an iterated Kalman filter. Similar to [16], FAST-LIO [4] also utilizes the iterated Kalman filter but introduces a new Kalman gain to reduce the complexity of computation. Based on [4], FAST-LIO2 [5] directly registers raw points without extracting features and introduces the iKD-Tree to increase mapping rate and odometry accuracy. LIOM [17] jointly optimizes LiDAR-IMU extrinsic parameters, IMU bias, and robot pose using graph-based optimization. It achieves better accuracy when compared with LOAM; however, the algorithm suffers from computational

inefficiency. LIO-SAM [3] estimates IMU bias in the factor graph exhibiting better real-time performance than LIOM and achieving accurate and robust state estimation. However, all of these LiDAR-inertial methods depend on the assumption of the static environment and are easily disturbed by moving objects.

B. Dynamic SLAM and Multi-Object Tracking

During the past decade, researchers have increasingly focused on the study of SLAM for dynamic environments. The traditional dynamic SLAM approaches mainly reject information from all movable objects to ensure the robustness of the system, which might cause feature sparsity and thus degrade the SLAM performance [18], [19], [20]. Some methods reject information drawn from moving objects but do not perform object tracking [21], [22]. The loss of dynamic information can also degrade the localization accuracy.

In recent years, the multi-object tracking and SLAM system in dynamic environments has been increasingly investigated. The existing methods can be classified into two types: loosely-coupled and tightly-coupled.

In the former, multi-object tracking and SLAM are performed independently [23], [24], [25]. SLAMMOT [23] first proposed simultaneously estimating ego-motion and multi-object motion, establishing a mathematical framework to decompose the estimation problem into two separated filter-based estimators. MaskFusion [24], an RGB-D SLAM system, uses both photometric and geometric information to track the detected objects in the scene. MLO [25] estimates ego-motion based on the static background and achieves object tracking through the least-squares method by fusing point clouds and 3D detection. However, the accuracy of multi-object tracking in these methods is heavily dependent on ego-pose estimation, which likely fails in complex dynamic environments.

Tightly-coupled multi-object tracking and SLAM methods are primarily proposed in visual SLAM systems [6], [9], [10], [11]. Specifically, CubeSLAM [6] firstly proposed the use of a multi-view bundle adjustment (BA) to jointly optimize ego-pose, states of objects, and feature points using only a monocular camera. Dynamic objects are tracked by a 2D visual object

tracking algorithm [26]. VDO-SLAM [9] tracks feature points on objects by leveraging dense optical flow. Motion model constraints are added to the factor graph to effectively enable the combined optimization of ego-pose and object states. DynaSLAM II [10] makes use of instance semantic segmentation and ORB feature [27] correspondences to track moving objects, jointly optimizing the static structure and trajectories of the camera and moving objects in a local-temporal window. TwistSLAM [11] achieves data association using optical flow estimation and performs a novel joint optimization by defining inter-cluster constraints modeled by mechanical joints. These vision-based methods mostly perform tracking in 2D image space and thus suffer from inaccurate 3D object detection, vulnerability to textureless or low-illumination environments, and occlusion.

To the best of our knowledge, our previous work DL-SLOT [7] is the first tightly-coupled LiDAR SLAM and multi-object tracking system without IMU. Moreover, DL-SLOT filters out all feature points belonging to movable objects, and easily suffers from feature sparsity when many movable objects are actually static in reality. The experiments shown in this letter indicate that the present method substantially outperforms DL-SLOT in terms of pose accuracy. The recent work LIO-SEGMENT [8] proposed an optimization framework similar to ours, but without using a sliding window. As a result, it is computationally expensive when there are multiple objects for tightly-coupled optimization. Additionally, this method does not remove the dynamic points, which results in only a slight improvement in pose accuracy compared to LIO-SAM [3]. In contrast, LIMOT achieves more accurate localization and higher computational efficiency by limiting the cost of joint optimization of the poses of the ego-vehicle and objects.

III. METHODS

A. Notation and System Overview

We consider W as the world frame and L_k as the LiDAR frames, related to the k -th LiDAR scan at time t_k , respectively. We denote $\mathbf{T}_b^a \in \text{SE}(3)$ as the pose of b in frame a . We also assume that the LiDAR frame coincides with the ego-vehicle frame for convenience. Therefore, the pose of the ego-vehicle in frame W at t_k is represented as $\mathbf{T}_{L_k}^W$ and the pose transformation from t_{k-1} to t_k is represented as $\mathbf{T}_{L_k}^{L_{k-1}}$. For simplification, we denote $\mathbf{T}_{L_k}^W$ as \mathbf{T}_k^W and $\mathbf{T}_{L_k}^{L_{k-1}}$ as \mathbf{T}_k^{k-1} . In addition, the pose of j -th object in W and the ego-vehicle frame at t_k are represented as \mathbf{T}_{k,O_j}^W and \mathbf{T}_{k,O_j}^L , respectively. Each can be converted into the other as follows:

$$\mathbf{T}_{k,O_j}^W = \mathbf{T}_k^W \cdot \mathbf{T}_{k,O_j}^L \quad (1)$$

We choose LIO-SAM as our baseline method since its optimization process is based on the factor graph framework, which is relatively easy to extend [28]. An overview of the proposed system is shown in Fig. 1. The system consists of four modules, Preprocessing, LiDAR Odometry, Factor Graph Optimization, and Multi-object Tracking. These modules are expounded upon in subsequent sections. The LiDAR Odometry module and IMU

pre-integration in the Preprocessing module of LIMOT are mainly inherited from LIO-SAM. However, there are significant differences between LIMOT and LIO-SAM. Firstly, LIMOT employs a trajectory-based dynamic feature filtering method that leverages tracking results to filter out features belonging to moving objects before scan-matching. Secondly, a factor graph optimization framework is utilized in LIMOT to optimize the bias of the IMU and the poses of both ego-vehicle and objects in a sliding window.

B. Preprocessing

1) *Object Detection*: To simplify the object observation model, we utilize CenterPoint [29], an open source real-time 3D LiDAR object detector, to generate the 3D bounding box and pose \mathbf{T}_{k,O_j}^L of an object in LiDAR frame.

2) *IMU Pre-Integration*: We perform IMU pre-integration to aggregate raw IMU measurements in the local frame, following [30]. We refer the reader to [30] for the detailed description.

C. Multi-Object Tracking

We use the approach in DL-SLOT [7] to predict the position of the tracked object by fitting its trajectory in a sliding window with a cubic order polynomial. Then, a M by N matching matrix Ψ_k is generated by calculating the distances between the M detected objects in the current scan and the predicted positions of the N tracked objects in the previous scan. The element $\psi_k^{i,j}$ in Ψ_k indicates the matching score between the i -th detected object and the j -th tracked object. This matching score can be calculated by (2).

$$\psi_k^{i,j} = \begin{cases} 1 - \frac{d_{i,j}}{\alpha} & (d_{i,j} < d_{\text{thres}}) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where α is a constant, $d_{i,j}$ is the distance between the detected object and the predicted position of the tracked object, and d_{thres} is the association distance threshold. The continuous shortest path algorithm [12] is employed to perform data association based on the matching matrix.

After completing the data association, the average velocities of the objects within the sliding window are calculated. An object is determined to be dynamic when its average velocity is greater than a given threshold v_{thres} .

D. LiDAR Odometry

1) *De-Skewing, Feature Extraction and Scan-Matching*: For point cloud de-skewing, a nonlinear motion model is utilized with the estimated motion from the IMU, which is more precise than using a linear motion model [17]. Edge and planar feature points are then extracted based on the local roughness of the point cloud. After removing feature points belonging to moving objects (see Section III-D2), point-to-edge and point-to-plane scan-matching is conducted between the current scan at t_k and the submap composed of a fixed-size set of history scans to obtain ego-pose \mathbf{T}_k^W . The initial transformation guess is obtained using the predicted ego-motion, $\tilde{\mathbf{T}}_k^W$, from IMU pre-integration.

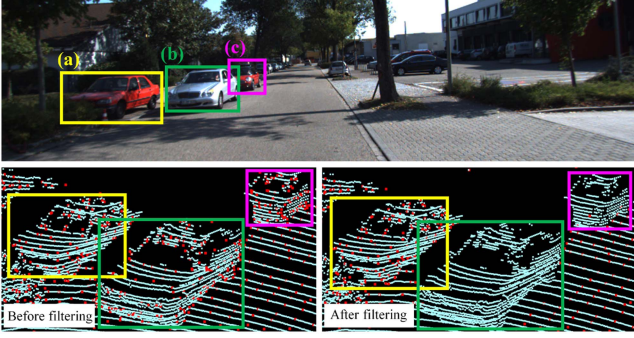


Fig. 2. Example of filtering out dynamic feature points. The light blue points denote the original point cloud and the red points denote the feature points. The feature points on the moving cars (b) and (c) are removed exactly by LIMOT, while the feature points on the static car (a) are remained.

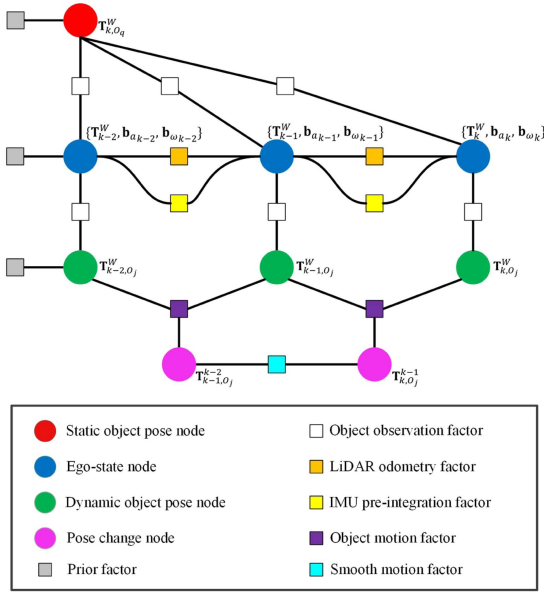


Fig. 3. Factor graph framework of LIMOT for joint optimization.

2) *Trajectory-Based Dynamic Feature Filtering*: For dynamic objects at t_k , we use their fitted trajectories generated by the Multi-object tracking module to predict their positions at t_{k+1} . Based on the predicted ego-pose $\hat{\mathbf{T}}_{k+1}^W$ at t_{k+1} , feature points located within the 3D bounding boxes of dynamic objects can be filtered out from the $k+1$ -th point cloud before scan-matching. An illustration is shown in Fig. 2. There are three cars (labeled by (a), (b), and (c)) in this scene. Car (a) is parked on the side of the road, and car (b) and car (c) are traveling on the road. It can be seen that our method can filter out the feature points (shown in red) extracted from the moving objects and remain the feature points on static objects.

E. Joint Factor Graph Optimization

The joint factor graph optimization framework is shown in Fig. 3. It consists of the factors providing constraints for optimization and variable nodes including the states of the ego-vehicle and surrounding objects. An object is deemed initialized

when it has been continuously tracked for 5 frames. Only the pose nodes of the initialized objects are added to the factor graph, which avoids false detections. We add pose nodes for initialized dynamic objects at each time t_k but maintain only one pose node in the factor graph when the object is judged as stationary to ensure the uniqueness of its global pose. This provides reliable static observation constraints. The residual formulation of each factor is given below.

Given the ego-poses at t_{k-1} and t_k estimated by LiDAR scan-matching, the residual of the LiDAR odometry factor can be defined as follows:

$$\mathbf{r}_{odo}^k(\mathbf{T}_{k-1}^W, \mathbf{T}_k^W) = (\mathbf{T}_{k-1}^{W^{-1}} \cdot \mathbf{T}_k^W) \cdot \mathbf{T}_k^{k-1^{-1}} \quad (3)$$

Then, based on the object detection results, the residual of the object observation factor is defined as follows:

$$\mathbf{r}_{obs}^{k,j}(\mathbf{T}_k^W, \mathbf{T}_{k,O_j}^W) = (\mathbf{T}_k^{W^{-1}} \cdot \mathbf{T}_{k,O_j}^W) \cdot \mathbf{T}_{k,O_j}^{L^{-1}} \quad (4)$$

The pose change of object \mathbf{T}_{k,O_j}^{k-1} between t_{k-1} and t_k can be calculated using (5), which also indicates the velocity of the object.

The object motion factor is a ternary factor associated with three nodes: two dynamic object pose nodes \mathbf{T}_{k-1,O_j}^W , \mathbf{T}_{k,O_j}^W and one pose change node \mathbf{T}_{k,O_j}^{k-1} . These are continually updated throughout the optimization process, but they must always satisfy (5). Thus, the residual of the object motion factor is defined by (6).

$$\mathbf{T}_{k,O_j}^{k-1} = \mathbf{T}_{k-1,O_j}^W^{-1} \cdot \mathbf{T}_{k,O_j}^W \quad (5)$$

$$\begin{aligned} & \times \mathbf{r}_{moti}^{k,j}(\mathbf{T}_{k-1,O_j}^W, \mathbf{T}_{k,O_j}^W, \mathbf{T}_{k,O_j}^{k-1}) \\ & = (\mathbf{T}_{k-1,O_j}^{W^{-1}} \cdot \mathbf{T}_{k,O_j}^W) \cdot \mathbf{T}_{k,O_j}^{k-1^{-1}} \end{aligned} \quad (6)$$

We assume that a dynamic object moves at a constant velocity over a short period of time, such that the pose changes of the object at successive times should be almost identical. Therefore, the residual of the smooth motion factor is defined as:

$$\mathbf{r}_{smoo}^{k,j}(\mathbf{T}_{k-1,O_j}^{k-2}, \mathbf{T}_{k,O_j}^{k-1}) = \mathbf{T}_{k-1,O_j}^{k-2^{-1}} \cdot \mathbf{T}_{k,O_j}^{k-1} \quad (7)$$

Finally, the optimization problem can be denoted as:

$$\begin{aligned} \mathcal{X}^* = \operatorname{argmin}_{\mathcal{X}} & \left\{ \|\mathbf{r}_P(\mathcal{X})\|^2 + \sum_{k \in \mathcal{I}} \|\mathbf{r}_I^k(\mathcal{X})\|_{Q_I}^2 \right. \\ & + \sum_{k \in \mathcal{L}} \left(\rho \|\log(\mathbf{r}_{odo}^k(\mathcal{X}))\|_{Q_{odo}}^2 + \sum_{j \in \mathcal{O}_k} \left(\rho \|\log(\mathbf{r}_{obs}^{k,j}(\mathcal{X}))\|_{Q_{obs}}^2 \right. \right. \\ & \left. \left. + \rho \|\log(\mathbf{r}_{moti}^{k,j}(\mathcal{X}))\|_{Q_{moti}}^2 + \rho \|\log(\mathbf{r}_{smoo}^{k,j}(\mathcal{X}))\|_{Q_{smoo}}^2 \right) \right) \left. \right\} \end{aligned} \quad (8)$$

where \mathcal{X} is the set of all variables, $\|\mathbf{r}_P(\mathcal{X})\|^2$ is the prior from marginalization, \mathcal{I} is the set of all IMU measurements, \mathcal{L} is the set of LiDAR scans in the sliding window, and Q represents the covariance matrix. We refer the reader to [30] for the formula of the IMU pre-integration factor residual $\mathbf{r}_I^k(\mathcal{X})$.

The operation $\log()$ represents the transformation from $SE(3)$ to $se(3)$. ρ is the Cauchy robust function [31]. Each LiDAR scan is associated with the tracked object set \mathcal{O}_k . For practical implementation, the states of the ego-vehicle and the objects are first jointly optimized with the IMU bias fixed. Subsequently, the IMU bias is estimated based on the optimized ego-vehicle poses. This process ensures the robustness and efficiency of the optimization.

IV. EXPERIMENTS

We conducted experiments using the public KITTI tracking dataset [32] and the self-collected dataset to evaluate the performance of the proposed LIMOT. To demonstrate the robustness of LIMOT to the detector, all experiments were based on real-time detection results from CenterPoint, whose weights are trained on the Argoverse 2 dataset [33]. For all the experiments, we set the sliding window size to 5, v_{thres} to 1 m/s, and α to 100. Additionally, the d_{thres} for initialized object association was set to 2 m. Experiments were carried out on a workstation with Ubuntu 20.04, equipped with an Intel Core Xeon(R) Gold 6248R 3.00 GHz processor, 32 G RAM, and a NVIDIA RTX A4000 16 GB graphic card.

A. Evaluation Metrics

The root-mean-square error (RMSE) of the translational and the rotational absolute trajectory errors (ATE_T [m] and ATE_R [rad]) are adopted to assess the accuracy of ego-poses [34]. Multi-object tracking performance is evaluated in two ways following [10]. The pose accuracy of the objects is also evaluated using the RMSE of ATE_T [m] and ATE_R [rad]. Tracking precision is evaluated using the multi-object tracking precision (MOTP) metric [35]. MOTP results are only given on the KITTI tracking dataset since it provides the ground truth tracking results for evaluation.

B. Baselines

The baseline multi-object tracking and SLAM methods are DL-SLOT [7] and LIO-SEGMOT [8]. LIO-SAM [3] is also compared since it is one of the state-of-the-art (SOTA) LiDAR-inertial SLAM methods, on which we build LIMOT. When only the proposed dynamic feature point removal is performed without joint optimization, the method is referred to as LIO-Dynaflit. In addition, we also provide experimental results of removing all feature points on movable objects without joint optimization, LIO-Allflit. All the methods above do not enable loop closure.

We selected two popular 3D online multi-object tracking methods, AB3DMOT [36] and PC3T [37], as well as DL-SLOT and LIO-SEGMOT as the baseline multi-object tracking methods. AB3DMOT first proposed to directly evaluate multi-object tracking in 3D space, which is suitable for LiDAR-based approaches. PC3T is a SOTA online multi-object tracking method that requires ground truth ego-poses as input.

C. KITTI Tracking Dataset

1) *Ego-Pose Evaluation*: The KITTI tracking dataset was collected in urban areas and along highways. All sequences in the KITTI tracking dataset were selected, except those without ego-motion. The comparative results are shown in Table I. LIO-Allflit shows a large translational error in sequence 01 because parked cars are representative in this sequence. The filtering of all their feature points impedes scan-matching. The pose accuracy of LIO-Dynaflit is higher than those of LIO-SAM and LIO-Allflit, demonstrating the effectiveness of our proposed dynamic feature filtering approach. Compared to DL-SLOT, all other methods achieve better performance, demonstrating that coupling IMU assists in localization. LIMOT achieves the best pose estimation performance in many of the evaluated sequences, confirming that jointly optimizing the poses of the ego-vehicle and objects is beneficial. LIO-SEGMOT does not improve the pose accuracy much compared to LIO-SAM because it does not consider the effect of dynamic features on the scan-matching. It is important to note that sequence 20 is a highway scene containing a large number of moving objects, so LIMOT, LIO-Dynaflit, and LIO-Dynaflit all have a large improvement compared to LIO-SAM and LIO-SEGMOT. Moreover, since the highway is a typical scene for LiDAR SLAM degradation, DL-SLOT shows the largest translational error.

The comparison of ego-trajectories of sequence 04 is shown in Fig. 4(a). The point cloud maps of sequence 01 generated by LIO-SAM and LIMOT are shown in Fig. 4(b). In the red ellipse of the former, there is a very obvious ghosting caused by the moving objects, which, however, is eliminated from the point cloud generated by LIMOT.

2) *Tracking Precision Evaluation*: The comparison results are shown in Table II. For each tracked object, its Intersection over Union (IoU) with ground truth label should exceed a threshold IoU_{thres} to be considered as a successful match. It can be found that our method shows the best results in terms of MOTP, indicating that our method achieves better tracking precision for all tracked objects.

3) *Object Pose Evaluation*: We further evaluated the pose accuracy of the individual objects. We selected 10 objects with the longest tracking frame length in Table I for evaluation. The experimental results are shown in Table III. Here TP [%] stands for the ratio of the number of tracked frames to the number of the frames of the object's ground truth trajectory. The IoU threshold for evaluating TP is taken as 0.25. Since we adopt the tracking method in DL-SLOT, the TP results of these two methods are the same on most sequences. However, the pose accuracy of objects is significantly improved in LIMOT, primarily attributed to the reduction of the ego-pose rotation error through coupling IMU measurements. Additionally, while LIMOT and LIO-SEGMOT track objects for nearly the same number of frames, LIMOT achieves superior pose accuracy for these objects.

The trajectories of the tracked object (id 0) and ego-vehicle in sequence 10 are demonstrated in Fig. 4(c). The blue solid line and the green dotted line represent the trajectories of the ego-vehicle and the tracked object, respectively. We further

TABLE I
RMSE OF ATE_T [m] AND ATE_R [rad] RESULTS OF EGO-POSE ESTIMATION COMPARISON ON THE KITTI TRACKING DATASET

Seq	LIO-SAM [3]		DL-SLOT [7]		LIO-SEGMOT [8]		LIO-Allfilt		LIO-Dynaflt		LIMOT	
	ATE_T [m]	ATE_R [rad]	ATE_T [m]	ATE_R [rad]	ATE_T [m]	ATE_R [rad]	ATE_T [m]	ATE_R [rad]	ATE_T [m]	ATE_R [rad]	ATE_T [m]	ATE_R [rad]
00	0.856	0.024	0.984	0.031	0.867	0.020	0.854	0.024	0.852	0.024	0.846	0.024
01	1.683	0.044	1.764	0.193	1.704	0.043	1.824	0.043	1.686	0.044	1.681	0.045
02	0.274	0.013	0.362	0.023	0.253	0.016	0.280	0.014	0.274	0.012	0.269	0.013
03	0.245	0.031	1.920	0.050	0.269	0.038	0.247	0.028	0.246	0.028	0.245	0.030
04	0.660	0.104	1.050	0.276	0.657	0.114	0.659	0.116	0.634	0.099	0.629	0.107
05	0.347	0.134	1.209	0.100	0.313	0.124	0.346	0.127	0.342	0.122	0.355	0.112
06	0.266	0.013	1.925	0.081	0.202	0.012	0.199	0.012	0.206	0.012	0.190	0.012
07	1.319	0.025	1.759	0.272	1.376	0.032	1.331	0.025	1.308	0.024	1.278	0.024
08	1.274	0.307	2.807	0.284	1.120	0.284	1.253	0.328	1.257	0.310	1.237	0.311
09	1.201	0.024	1.597	1.123	1.135	0.025	1.181	0.024	1.220	0.025	1.175	0.030
10	0.438	0.108	0.860	0.137	0.495	0.106	0.447	0.118	0.438	0.106	0.425	0.106
11	0.262	0.202	0.303	0.214	0.282	0.283	0.260	0.182	0.262	0.173	0.266	0.181
13	0.260	0.044	0.331	0.051	0.258	0.040	0.263	0.043	0.262	0.043	0.256	0.045
14	0.105	0.013	0.151	0.197	0.095	0.013	0.111	0.014	0.107	0.013	0.108	0.012
15	0.275	0.066	0.385	0.012	0.291	0.076	0.274	0.065	0.273	0.066	0.274	0.066
18	0.385	0.141	0.851	0.166	0.371	0.168	0.374	0.157	0.376	0.159	0.373	0.160
19	0.916	0.122	1.006	0.358	0.955	0.136	0.916	0.131	0.916	0.135	0.915	0.120
20	9.488	0.021	22.349	0.043	9.313	0.023	2.656	0.018	2.480	0.024	1.710	0.027
mean	1.125	0.080	2.312	0.201	1.109	0.086	0.749	0.082	0.730	0.079	0.680	0.079

Bold and underlined text indicate the best and the suboptimal result, respectively

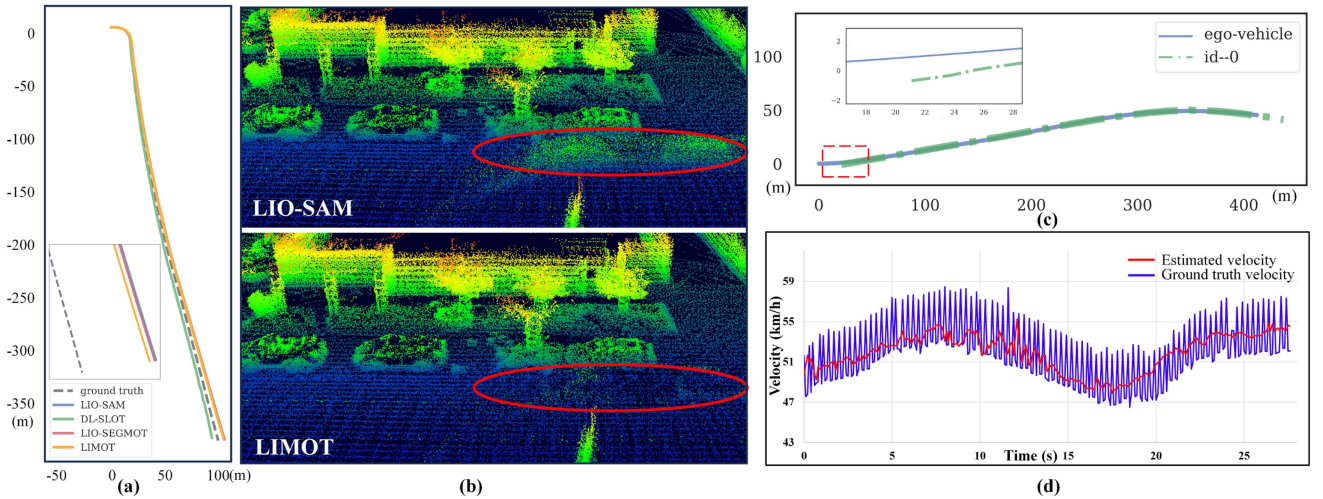


Fig. 4. Qualitative results of LIMOT on the KITTI tracking dataset. (a) Comparison of ego-trajectories in sequence 04 of the KITTI tracking dataset. (b) Comparison of point cloud maps generated by LIO-SAM and LIMOT. (c) Trajectories of the main tracked object (id 0) and ego-vehicle in sequence 10 of the KITTI tracking dataset. (d) Comparison between the ground truth and estimated instantaneous velocity of the tracked object in (c).

TABLE II
MOTP [%] RESULT COMPARISON OF DIFFERENT MULTI-OBJECT TRACKING ALGORITHMS ON THE KITTI TRACKING DATASET

Method	MOTP [%]	
	($IOU_{thres} = 0.25$)	($IOU_{thres} = 0.5$)
AB3DMOT	61.37	64.17
PC3T	61.26	64.07
DL-SLOT	60.13	63.60
LIO-SEGMOT	61.45	64.22
LIMOT	62.25	64.26

compare the estimated instantaneous velocity of the object (id 0) in sequence 10 with the ground truth. As shown in Fig. 4(d), the

ground truth instantaneous velocities (blue line) exhibit significant fluctuation due to the imprecise object annotations within the dataset. In contrast, the estimated instantaneous velocities by LIMOT (red line) are smoother and in the middle of the fluctuation range of the ground truth. This indicates that it is appropriate to add the constant velocity constraint to the factor graph optimization.

D. Self-Collected Dataset

1) *Data Collection:* We collected this dataset using the TIEV platform [38] on the North Jiasong Road, Jiading District, Shanghai. Fig. 5 shows an overview of this dataset and it can be seen that this scene is rich in dynamic objects. TIEV is equipped

TABLE III
RESULTS OF OBJECT POSE ESTIMATION COMPARISON ON THE KITTI TRACKING DATASET

Seq / Obj.id	DL-SLOT [7]			LIO-SEGMOT [8]			LIMOT		
	TP [%]	ATE _T [m]	ATE _R [rad]	TP [%]	ATE _T [m]	ATE _R [rad]	TP [%]	ATE _T [m]	ATE _R [rad]
04 / 2	27.07	2.799	1.760	27.71	0.711	0.241	27.07	0.697	0.233
05 / 31	40.40	0.924	0.223	40.07	0.304	0.175	40.40	0.302	0.163
08 / 8	23.33	0.661	0.346	24.36	0.430	0.122	23.08	0.384	0.130
08 / 13	48.85	0.827	0.116	50.76	0.656	0.104	48.85	0.641	0.105
10 / 0	93.88	0.943	0.168	88.44	0.439	0.156	93.88	0.368	0.156
11 / 0	51.21	0.445	0.129	50.40	0.292	0.186	51.21	0.277	0.189
18 / 2	29.17	0.566	0.878	31.44	0.246	0.580	30.68	0.261	0.602
18 / 3	46.67	0.493	0.456	45.61	0.386	0.560	47.02	0.372	0.459
20 / 12	37.79	10.342	0.106	45.10	1.701	0.105	46.66	0.826	0.102
20 / 122	30.20	1.005	0.117	29.41	0.258	0.127	30.20	0.292	0.131
mean	42.86	1.901	0.430	43.33	0.542	0.236	43.90	0.442	0.227

Bold and underlined text indicate the best and the suboptimal result, respectively

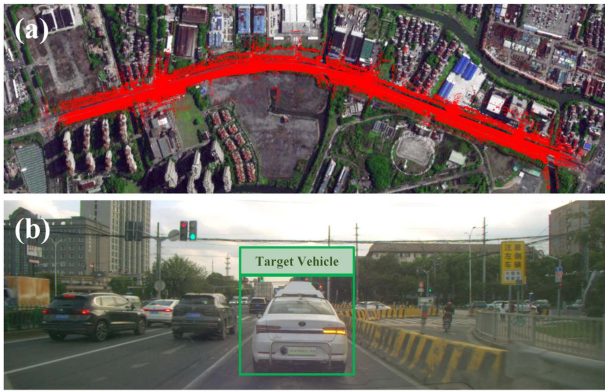


Fig. 5. Overview of the self-collected dataset. (a) LIMOT mapping result and the satellite map of the self-collected dataset. They are aligned manually to better illustrate the environment of the dataset. (b) A representative front view image of the self-collected dataset.

TABLE IV
RESULTS OF POSE ESTIMATION COMPARISON ON THE SELF-COLLECTED DATASET

Method	Ego-pose		Target object pose		
	ATE _T [m]	ATE _R [rad]	TP [%]	ATE _T [m]	ATE _R [rad]
LIO-SAM	3.344	0.049	—	—	—
DL-SLOT	5.188	0.118	68.61	0.524	0.092
LIO-SEGMOT	3.243	0.046	85.99	2.593	0.079
LIO-Allfilt	3.358	0.046	—	—	—
LIO-Dynafilt	3.314	0.047	—	—	—
LIMOT	3.005	0.046	71.73	0.212	0.092

Bold and underlined text indicate the best and the suboptimal result, respectively

with a Velodyne HDL-64E LiDAR and an RTK-Inertial Navigation System (INS), which provides IMU data at 100 HZ as well as high-precision ground truth. In addition, we used a target vehicle, playing the role of the tracked object, equipped with the same INS, so we can obtain its ground truth trajectory.

2) *Pose Evaluation*: The pose evaluation results for both the ego-vehicle and the target vehicle are presented in Table IV. Compared to LIO-SAM, the ego-pose result of LIMOT presents in an average improvement of 10.14% and 6.12% in terms of

TABLE V
AVERAGE TIME-CONSUMING OF THE MAIN FUNCTIONAL MODULES FOR PROCESSING ONE SCAN

Module	Average Runtime (ms)		
LiDAR Odometry	78.4 ± 0.4		
Multi-object Tracking	0.9		
Factor Graph Optimization	1.5 ± 7.4		
	LIO-SAM	LIO-SEGMOT	LIMOT
FPS	12.6	7.5	11.3

ATE_T and ATE_R, respectively. As for the target object pose result, although LIO-SEGMOT tracks the object for a longer period of time, it shows a significant translational error. LIMOT obtains the most accurate target object position.

E. Running Time Analysis

We calculated the average time-consumption of the main functional modules of LIMOT on the KITTI tracking dataset. The results are shown in Table V. The detector CenterPoint can achieve real-time operation, processing around 20 frames per second (FPS), with a detection range from 50 m in front of the ego-vehicle to 30 m behind. Compared with LIO-SAM, the time consumption of LiDAR odometry and factor graph optimization increase by 0.4 ms and 7.4 ms, respectively, which correspond to the runtime of the dynamic feature filtering and joint optimization of the states of objects. LIO-SEGMOT performs significantly slower than LIMOT because of its computationally intensive factor graph structure.

V. CONCLUSION

We present LIMOT, a tightly-coupled system for LiDAR-inertial SLAM and multi-object tracking capable of jointly optimizing the poses of the ego-vehicle and surrounding objects in a sliding window. Furthermore, this method can filter out feature points belonging to moving objects based on the approximated object trajectories, while the remaining feature points on static objects are used to provide constraints for scan-matching, which enhances the robustness of the system and improves its

performance in dynamic environments. Experimental results show that our method improves the pose accuracy of ego-vehicle and objects, as well as the tracking accuracy, which demonstrates that LiDAR-inertial SLAM and multi-object tracking can exhibit mutual benefits on each other.

In the future, it will be advantageous to introduce a dynamics model for moving objects in the environment to obtain more accurate object states. Future work could also involve the utilization of sensor data from the road-test unit to further improve system performance.

REFERENCES

- [1] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robo.: Sci. Syst. Conf.*, 2014, pp. 1–9.
- [2] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
- [3] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5135–5142.
- [4] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.
- [5] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [6] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.
- [7] X. Tian, Z. Zhu, J. Zhao, G. Tian, and C. Ye, "DL-SLOT: Tightly-coupled dynamic LiDAR SLAM and 3D object tracking based on collaborative graph optimization," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1017–1027, Jan. 2024.
- [8] Y.-K. Lin, W.-C. Lin, and C.-C. Wang, "Asynchronous state estimation of simultaneous EGO-motion estimation and multiple object tracking for LiDAR-inertial odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10616–10622.
- [9] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: A visual dynamic object-aware SLAM system," 2020, *arXiv:2005.11052*.
- [10] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5191–5198, Jul. 2021.
- [11] M. Gonzalez, E. Marchand, A. Kacete, and J. Royan, "TwistSLAM: Constrained SLAM in dynamic environment," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6846–6853, Jul. 2022.
- [12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. NJ, USA: Prentice-Hall, 1993, ch. 9, pp. 320–323.
- [13] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3923–3929.
- [14] M. Demir and K. Fujimura, "Robust localization with low-mounted multiple Lidars in urban environments," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 3288–3293.
- [15] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma, "A robust pose graph approach for city scale LiDAR mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1175–1182.
- [16] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A Lidar-inertial state estimator for robust and efficient navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8899–8906.
- [17] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 3144–3150.
- [18] B. Bescos, J. M. Fàcil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.
- [19] V. Vaquero, K. Fischer, F. Moreno-Noguer, A. Sanfeliu, and S. Milz, "Improving map re-localization with deep movable objects segmentation on 3D LiDAR point clouds," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 942–949.
- [20] S. Zhao, Z. Fang, H. Li, and S. Scherer, "A robust laser-inertial odometry and mapping method for large-scale highway environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1285–1292.
- [21] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 4421–4428.
- [22] Q. Li et al., "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8473–8482.
- [23] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, 2007.
- [24] M. Runz, M. Buffier, and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2018, pp. 10–20.
- [25] T. Ma, G. Jiang, Y. Ou, and S. Xu, "Semantic geometric fusion multi-object tracking and LiDAR odometry in dynamic environment," *Robotica*, vol. 42, no. 3, pp. 891–910, 2024.
- [26] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
- [27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
- [28] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5692–5698.
- [29] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11784–11793.
- [30] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.
- [31] J. Wang, D. Lyu, Z. He, H. Zhou, and D. Wang, "Cauchy kernel-based maximum correntropy Kalman filter," *Int. J. Syst. Sci.*, vol. 51, no. 16, pp. 3523–3538, 2020.
- [32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [33] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, J. Vanschoren and S. Yeung, Eds. 2021, vol. 1. [Online]. Available: https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/4734ba6f3de83d861c3176a6273cac6dPaper-round2.pdf
- [34] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7244–7251.
- [35] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.
- [36] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 10359–10366.
- [37] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, "3D multi-object tracking in point clouds based on prediction confidence-guided data association," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5668–5677, Jun. 2022.
- [38] J. Zhao et al., "TiEV: The tongji intelligent electric vehicle in the intelligent vehicle future challenge of China," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 1303–1309.