LIO-Track: Tightly-Coupled LiDAR-Inertial Odometry with Multi-Object Tracking

Zhengcheng Yu Hongbo Duan Xueqian Wang Bin Liang

Abstract— **In this paper, we propose LIO-Track, a tightly-coupled LiDAR-Inertial Odometry (LIO) with Multiple Object tracking (MOT) system that achieves object tracking together with Simultaneous Localization And Mapping (SLAM) and obtains better pose estimation accuracy. We formulate the whole LIO-MOT problem in a hierarchical factor graph optimization and adopt different association and optimization strategies for objects moving quickly and slowly to hold the uniqueness of the same tracking object. We design a novel Error State Kalman Filter (ESKF) for successfully associated objects that take observation from ego pose and detection results. Considering the spatial-temporal correlation of dynamic objects, we build a spatial-temporal corridor for each object and use this corridor to filter dynamic object points. Meanwhile, we represent object spatial information as a normal distribution and use divergence between normal distributions to measure the difference between objects. Experiments on the KITTI dataset verify the strength of our approach compared to the open-source state-of-the-art approach. We also validate our proposed framework in real-world environments.**

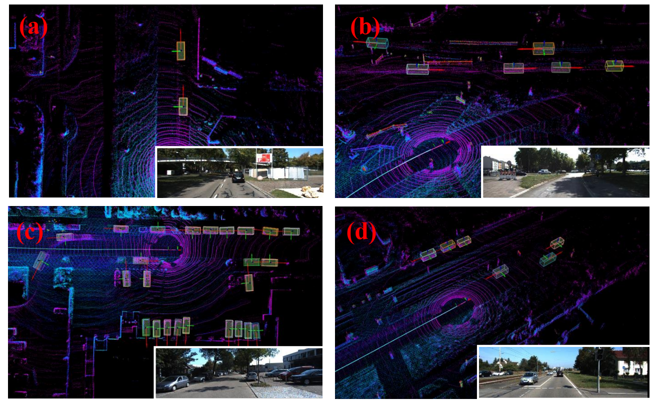

Fig. 1. The correspondence of mapping and tracking results with front view camera images on KITTI sequences 0926-0013(a), 0926-0051(b), 0926-0009(c), and 0926-0015(d). The green bounding boxes with coordinates are detected objects with their orientations.

I. Introduction

With the development of robot navigation and autonomous driving, Simultaneous Localization and Mapping (SLAM) have become increasingly important for robots and self-driving cars to perceive and record surrounding information [1]–[3]. The performance of SLAM directly determines the upper limit of the system's capability. However, most SLAM methods [1], [4], [5] are based on the hypothesis that the surrounding environment is completely static, which is not matched in high-dynamic scenes such as urban environment and will lead to pose estimation bias. Therefore, a more common perception scheme that applies to both static and dynamic environments is required. In recent years, as deep learning technology evolves, 3D object detection [6] provides more and more accurate 6-DoF relative pose of every object in the scene. The process of associating detected objects between different frames is called Multiple Object Tracking (MOT). Through 3D MOT, robots and autonomous cars can monitor the change of environment [7] and predict the future trajectories of objects. This information will be available for the planning and decision-making for robots and self-driving cars in complex scenarios.

In recent years, the academic community has seen the combination of SLAM and MOT gaining more and more attention [7]–[9]. In the majority of previous SLAM-MOT works, like VDO-SLAM, LIMOT and LIO-SEGMOT, the object state is only computed by factor graph optimization, and the initial value of the object state is given by constant linear and angular velocity (CLAV) kinematic model which is unfit in the situation of object acceleration or deceleration. The other question is that the detection breaks due to occlusions or poor ability of the 6-DoF detector will lead to unknown changes in object state which make it hard for using the CLAV model to update the object state. The wrong initial object state will degrade the overall graph optimization result. To acquire a more reliable initial object state, we propose a novel Error State Kalman Filter (ESKF) specifically designed for object tracking. We take ESKF for objects that are successfully associated with current observations to provide a credible initial object state for bundle adjustment between ego and objects.

Although a lot of previous works [10] [11] have noticed that there are spatial and temporal relationships between objects from different LiDAR frames, they only consider the object motion between two consecutive frames and do not build the entire knowledge of object spatial-temporal information. In this paper, we establish a spatial-temporal corridor for each object. This corridor can be used to filter out points belonging to dynamic objects and restore the static environment structure at the same time. Through exploiting the relationship between objects from different LiDAR frames, it suits well in dealing with situations where objects alternate between dynamic and static states.

While other works mainly focus on enhancing SLAM functionality, we also pay attention to holding the uniqueness of tracking objects which means the same object has the

All authors are with Tsinghua University

This work was supported by the Shenzhen Science and Technology Program (Grant No. RCJC20210706091946001) and the Shenzhen Science and Technology Program (Grant No. ZDCY20250901104207008).

same tracking index. Considering that the most common and serious mistake that the 6-DoF detectors make is miscalculating the yaw angle θ to $\theta \pm 180°$, the association error computed by pose will be unstable and will result in mismatching. For this reason, we propose a probability representation based on Gaussian distribution for object spatial information and use divergence to measure the discrepancy between different objects. This kind of representation avoids including reverse rotation error thus making the process of association more robust while reducing the complexity and computation compared with 3D IoU. In contrast to LIO-SEGMOT [7] and LIMOT [10], when dealing with objects moving at high and low speeds, we add different kinds of factors to the optimization graph to carry out association. Experiments on both the KITTI dataset and the real world demonstrate the effectiveness of this proposed method.

The main contributions of this paper could be summarized as follows:

1) We design a novel Error State Kalman Filter specifically designed for object tracking.
2) We build a corridor for each object to acquire its spatial-temporal correlation between different LiDAR frames and use this corridor to filter out points belonging to dynamic objects.
3) We use the normal distribution to represent object spatial information and take divergence to measure the difference between objects.
4) For fast-moving and slow-moving objects, we adopt different association strategies while adding different kinds of factors to carry out graph optimization.

II. Related work

A. Tightly-Coupled LiDAR Inertial Odometry

During the past decade, the fusion of data from the Inertial Measurement Unit (IMU) and LiDAR has become the mainstream of LiDAR SLAM research. In the LIO system, the tightly-coupled [2] [12] [13] method takes the inner properties of different sensors into consideration and fuses all residuals directly from observations in a unified state estimator rather than only fusing the output from each component modularly which is done by the loose-coupled method. In the field of tightly-coupled LIO systems, LIOM [14] project points from all LiDAR sweep to pivot LiDAR sweep to build a sliding window. LiLi-OM proposes a novel LIO scheme for both solid-state and mechanical LiDARs and directly fusing IMU and LiDAR measurements. LINS [15] use Iterated Extended Kalman Filter (IEKF) to update state iteratively. Fast-LIO [2] considers motion compensation in back propagation and adjusts the update formula of IEKF from the observation dimension to the state dimension to reduce computation. Following Fast-LIO, Fast-LIO2 [12] uses incremental kd-tree [16] to save the local map thus reducing the time cost of querying the nearest neighbor of laser points and removes the hand-engineered feature extraction module to make the algorithm more applicable to emerging LiDAR types and different scanning patterns. Furthermore, Faster-LIO [13] leverages incremental voxel structure as its point cloud container to support incremental insertion and parallel KNN querying in substitution of complicated tree-based method. LIO-SAM [3] formulates the question of LiDAR-Inertial Odometry atop a single factor graph while marginalizing old laser scans for pose optimization and matching current scan to local map rather than global map to achieve real-time functionality.

B. SLAM with Multi-Object Tracking

Most of the traditional LiDAR and Visual SLAM systems [3] are based on the assumption of an unchanged environment. However, a large number of real-world dynamic scenarios violate this assumption, thus resulting in the degeneration and drift of those systems.

To tackle this problem and improve the robustness of SLAM, in the field of visual SLAM, the early work DS-SLAM [17] uses object detection, semantic segmentation and moving consistency check to filter out dynamic objects. In contrast to DS-SLAM, MaskFusion [18] takes instance-level semantic segmentation to track and reconstruct independent dynamic or static objects in real time. CubeSLAM [11] generates high-quality cuboid proposals from both 2D bounding box and vanishing points sampling for single image object detection and takes a multi-view bundle adjustment framework to optimize the pose of the camera and objects jointly. The most distinguishable work that CubeSLAM has done is that instead of treating dynamic objects as outliers, it uses them to provide large-range geometry and scale constraints to reduce monocular drift and improve estimation of camera pose thus demonstrating that the parts of 3D object detection and multi-view SLAM are mutually beneficial. VDO-SLAM [8], as the first full SLAM system to achieve object detection, semantic segmentation, depth and optical flow prediction as well as estimation of pose and velocity for every rigid object in the scene, exploits semantic information to track dynamic objects and estimate their motion without prior knowledge of objects' geometry shape. VDO-SLAM also provides a joint optimization framework to estimate the pose and velocities of the camera and objects, which is widely used by subsequent researchers.

The combination of LiDAR SLAM and object tracking appears later than the visual SLAM field [9] [7] [10]. Recent works LIO-SEGMOT [7] and LIMOT [10] are based on LIO-SAM [3], one of the most popular LIO frameworks, to formulate the whole question of estimation of ego-motion as well as pose and velocity of object in a unified factor graph. LIMOT [10] employs a trajectory-based filtering method to remove features belonging to dynamic objects before scan matching. Given that the keyframe selecting mechanism in LIO-SAM will lead to the desynchronization between results of detection and tracking, LIO-SEGMOT [7] proposes a mock detection method to align tracking objects from both keyframe and non-keyframe, in other words, finish asynchronous state estimation. Additionally, LIO-SEGMOT uses a hierarchical criterion to determine the coupling mode of odometry and tracking objects. For more stable tracking, TRLO [9] designs Unscented Kalman Filter (UKF) [19] as

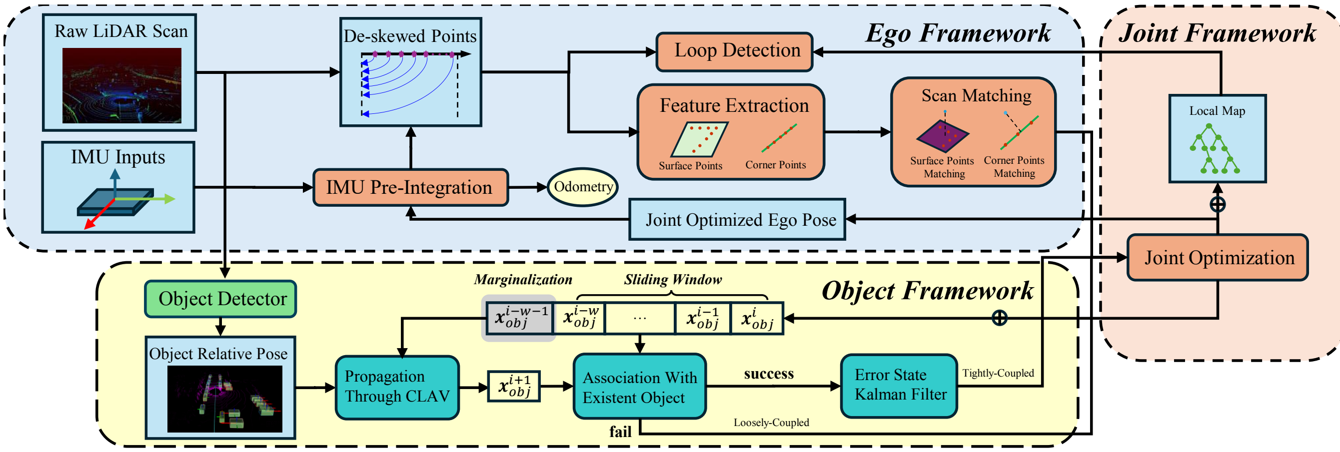

Fig. 2. The structure of the LIO-Track system. The system consists of three main sub-frameworks: ego-motion estimation framework, object tracking framework, and joint optimization framework. The details for each part will be discussed in section III

its object tracker and distinguishes between dynamic objects and static objects using estimated velocity so as to identify and filter out all dynamic points. Based only on the LiDAR sensor, TRLO takes the Generalized Iterative Closest Point (GICP) algorithm as DLO [20] to carry out scan-to-scan matching using static points. LIO-LOT [21] is the most recent LIO system with the MOT framework which uses a sliding window to retain previous frames while marginalizing the oldest state. To address the detector's lack of accuracy, LIO-LOT introduces feature-level residuals for associated objects.

III. Methodology

In this paper, we denote ${}^{w}\mathbf{x}_l = [{}^{w}\mathbf{R}_l; {}^{w}\mathbf{p}_l; {}^{w}\mathbf{v}_l; \mathbf{b}_a; \mathbf{b}_g]$ as ego state, where ${}^{w}\mathbf{T}_l = \{{}^{w}\mathbf{R}_l, {}^{w}\mathbf{p}_l\} \in SE(3)$ is ego pose. We write ${}^{w}\mathbf{x}_{obj} = [{}^{w}\mathbf{T}_{obj}, {}^{w}\boldsymbol{\xi}_{obj}]$ as object state, where object pose ${}^{w}\mathbf{T}_{obj}$ and velocity ${}^{w}\boldsymbol{\xi}_{obj}$ are both represented in $SE(3)$ space.

A. Tightly-Coupled LiDAR-Inertial Odometry via Smoothing and Mapping

Tightly-Coupled LiDAR-Inertial Odometry (LIO) fuse raw measurement from both LiDAR and Inertial Measurement Unit (IMU) to achieve the task of Simultaneous Localization and Mapping (SLAM). LIO-SAM is one of the best tight-coupled LIO systems. It formulates the SLAM question in a unified factor graph optimization framework and is convenient to extend so we use it as our SLAM subsystem.

LIO-SAM adopts isam2 [22] [23] to incrementally build the Bayes tree when the new factors arrive and solve the optimization problem through backward substitution of square root information matrix obtained through variable elimination. This kind of smoothing and mapping method brings effectiveness and efficiency to LIO-SAM in large-scale scenes. At the beginning of LIO-SAM, The raw distorted LiDAR points are de-skewed by IMU measurements and then be extracted as surface points or corner points to carry out scan matching, and the result is taken as a constraint of odom factor in the optimization graph, which also includes the three other kinds of factors, namely IMU pre-integration factor, loop factor and GPS factor.

B. Error State Kalman Filter Designed for Tracking Object

We propose a novel Error State Kalman Filter for successfully associated tracking objects. In the constant linear and angular velocity (CLAV) model, it propagates object pose from time i to time $i+1$ based on constant velocity assumption where object velocity is given by the last step graph optimization as:

$$ {}^{w}\mathbf{T}^{i+1}_{obj} = {}^{w}\mathbf{T}^{i}_{obj} \oplus \mathrm{Exp}(\mathrm{Log}({}^{w}\boldsymbol{\xi}^{i}_{obj})\Delta t_{i+1,i}) \tag{1} $$

where Log and Exp are logarithm and exponential mapping between manifold space and vector space. $\Delta t_{i+1,i}$ is the time interval between time indices i and $i+1$. The object's pose is set to identity matrix while its velocity is set to zero vector at initialization.

However, the CLAV assumption is not always matched when it comes to object acceleration or deceleration so we take ESKF as our state estimator before graph estimation.Considering that in traditional ESKF [24] there are two white Gaussian noises added into acceleration and angular velocity while object poses between LiDAR frames lack of high-frequency intermediate measurement like IMU, we add white Gaussian noises to simplified velocity

$$ \mathrm{Log}({}^{w}\boldsymbol{\xi}_{obj}) = [{}^{w}\mathbf{R}^{T}_{obj}\,{}^{w}\mathbf{v}_{obj}; {}^{w}\boldsymbol{\omega}_{obj}] \tag{2} $$

where ${}^{w}\mathbf{v}_{obj}$ and ${}^{w}\boldsymbol{\omega}_{obj}$ is linear velocity and angular velocity of the object respectively. The kinematic equation of true state in continuous time can be written as:

$$ {}^{w}_{\mathrm{t}}\dot{\mathbf{p}}_{obj} = {}^{w}\mathbf{v}_{obj} - \boldsymbol{\eta}_v \tag{3} $$

$$ {}^{w}_{\mathrm{t}}\dot{\mathbf{R}}_{obj} = {}^{w}_{\mathrm{t}}\mathbf{R}_{obj}({}^{w}\boldsymbol{\omega}_{obj} - \boldsymbol{\eta}_\omega)^\wedge \tag{4} $$

where $\boldsymbol{\eta}_\omega$ and $\boldsymbol{\eta}_v$ are both zero-mean white Gaussian noises for angular velocity and linear velocity. Symbol $^\wedge$ denotes mapping a three-dimensional vector to a skew-symmetric matrix. The kinematic equation of nominal object state ${}^{w}_{\mathrm{n}}\dot{\mathbf{p}}_{obj}$, ${}^{w}_{\mathrm{n}}\dot{\mathbf{R}}_{obj}$ is the same as true state but without white

Gaussian noise. The corresponding relationship between the nominal state and the true state is defined as:

$$ {}^{w}_{\mathrm{t}}\mathbf{p}_{obj} = {}^{w}_{\mathrm{n}}\mathbf{p}_{obj} + \delta\mathbf{p} \tag{5}$$
$$ {}^{w}_{\mathrm{t}}\mathbf{R}_{obj} = {}^{w}_{\mathrm{n}}\mathbf{R}_{obj}\mathrm{Exp}(\delta\boldsymbol{\theta}) \tag{6}$$

where $\delta\mathbf{T} = [\delta\mathbf{p}; \delta\boldsymbol{\theta}]$ is error state of position and rotation.

We derive the kinematic equation of error state in continuous time and get its discrete-time form through Euler integration:

$$\delta\mathbf{T}(t + \Delta t) = \mathbf{F}\delta\mathbf{T}(t) + \mathbf{w} \tag{7}$$

where $\mathbf{F}$ is the linearized Jacobian matrix of error state kinematic equation at time t and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ the integration of the white Gaussian noises. The process of shifting the object state from the current time to the next timestamp is called prediction. The predicted error state and its covariance of error state will be $\check{\delta\mathbf{T}} = \mathbf{F}\delta\mathbf{T}$ and $\check{\mathbf{P}} = \mathbf{F}\mathbf{P}\mathbf{F}^T + \mathbf{Q}$.

During update phase we take raw observation from current ego pose ${}^{w}\mathbf{T}_l$ and object relative pose ${}^{l}\mathbf{T}_{obj}$ given by detector, that is ${}^{w}_{\mathrm{obs}}\mathbf{T}_{obj} = \{{}^{w}_{\mathrm{obs}}\mathbf{R}_{obj}, {}^{w}_{\mathrm{obs}}\mathbf{p}_{obj}\} = {}^{w}\mathbf{T}_l * {}^{l}\mathbf{T}_{obj}$. Then we use raw observation object state and propagated object state ${}^{w}\check{\mathbf{T}}_{obj} = \{{}^{w}\check{\mathbf{R}}_{obj}, {}^{w}\check{\mathbf{p}}_{obj}\}$ to build Kalman observation $\mathbf{z} = [{}^{w}_{\mathrm{obs}}\mathbf{p}_{obj} - {}^{w}\check{\mathbf{p}}_{obj}; \mathrm{Log}({}^{w}\check{\mathbf{R}}_{obj}^{T}\,{}^{w}_{\mathrm{obs}}\mathbf{R}_{obj})]$ and write error state observation equation:

$$\mathbf{z} = \mathbf{h}(\delta\mathbf{T}) + \mathbf{v} \tag{8}$$

where $\mathbf{h}$ is the observation function of error state and $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{V})$ is observation noise. It is obvious to see that the Jacobian matrix of observation function is $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \delta\mathbf{T}}\big|_{\delta\mathbf{T}=\mathbf{0}} = \mathbf{I}_{6\times6}$. The reason we make expansion around $\delta\mathbf{T} = \mathbf{0}$ is that the error state will be reset to zero after every update.

According to the above derivation, we follow the formula of Kalman Filter [25] to update the error state and add it to the object's nominal state.

$$\mathbf{K} = \check{\mathbf{P}}\mathbf{H}^T(\mathbf{H}\check{\mathbf{P}}\mathbf{H}^T + \mathbf{V})^{-1} \tag{9}$$
$$\hat{\delta\mathbf{T}} = \mathbf{K}(\mathbf{z} - \mathbf{h}(\check{\delta\mathbf{T}})) \tag{10}$$
$${}^{w}\hat{\mathbf{T}}_{obj} = {}^{w}\check{\mathbf{T}}_{obj} \oplus \hat{\delta\mathbf{T}} \tag{11}$$
$$\hat{\mathbf{P}} = (\mathbf{I} - \mathbf{K}\mathbf{H})\check{\mathbf{P}} \tag{12}$$

where $\delta\mathbf{T} \sim \mathcal{N}(\hat{\delta\mathbf{T}}, \hat{\mathbf{P}})$ is distribution of updated error state and the symbol $\oplus$ denotes generalized addition of vector and manifold. To make sure the distribution of tracking object state remains unchanged before and after the update, which means:

$$({}^{w}\check{\mathbf{T}}_{obj} \oplus \hat{\delta\mathbf{T}}) \oplus \delta\mathbf{T}^{+} = {}^{w}\check{\mathbf{T}}_{obj} \oplus \delta\mathbf{T} \tag{13}$$

the error state after reset will be $\delta\mathbf{T}^{+} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}^{+})$, where $\mathbf{P}^{+} = \mathbf{J}\hat{\mathbf{P}}\mathbf{J}^T$ and $\mathbf{J} = \mathrm{diag}(\mathbf{I}_{3\times3}, \mathbf{I}_{3\times3} - \frac{1}{2}\hat{\delta\boldsymbol{\theta}}^{\wedge})$. The reset error state will be used for next iteration.

It needs to be mentioned that if a LIO system itself suffers from drastic vibration, the use of ESKF may enlarge the system's error because the observation is taken partially from the ego pose which is given by scan matching.

C. Dynamic Points Filtering Using Object Spatial-Temporal Corridor

The dynamic objects violating the assumption of a static environment will lead to unexpected drift and degeneration of the SLAM system especially in highly dynamic scenarios. So, removing points belonging to dynamic objects is a valuable practice for SLAM applications. Nevertheless, the traditional filtering methods [9] just leverage velocity estimated by consecutive frames to determine whether an object is dynamic or static, then decide whether to carry out filtering based on that result. Therefore, in the cases that objects alternate between dynamic and static states, these methods will fail to filter out points that belong to temporarily stopping objects.

Different from [9], [26], [27], We add bounding box information to every object trajectory point to form a spatial-temporal corridor

$$\mathcal{T} = \{{}^{w}\mathbf{T}^{i}_{obj}, \mathrm{BBox}_i, t_i\}_{i=1,\cdots,n} \tag{14}$$

Once the object is determined in motion for larger than k steps, we exclude out point cloud within the bounding box corresponding to each spatial-temporal corridor point. After filtering, these dynamic points can be effectively separated from the static environment, reducing the influence of errors caused by the movement of dynamic objects. The spatial-temporal corridor method is especially suitable for the case where objects alternates between dynamic and static states, which is very common in cases such as vehicle brake and start at the intersection and can be used online or offline, which means carrying out filtering during localization and mapping or after the built of the map.

D. 3D Bounding Box Representation Based on Normal Distribution

If the association criterion and the corresponding factor are not the same, when the results given by different criteria are not the same, it will arise re-linearization, which means a significant adjustment of the Bayes tree and increase computation sharply, even disturb the whole system. Therefore, the association criterion and error computation should be the same if we want to update the association during optimization. Therefore, it is in urgent requirement to design a criterion for both association and optimization. However, the pose difference based criterion which is adopted by most LIO-MOT method sometimes leads to mis-association and an amplified optimization error because of the false detection.

We use CenterPoint [6], one of the concise and lightweight point cloud object detection networks, as our detector while taking TensorRT to quantize the model in order to improve the inference speed. We also take SE-SSD as candidate detector for fair comparison. However, for most of the 3D point cloud object detectors, owing to the spatial sparsity of point cloud, the most common error detectors they make is detecting objects as facing the opposite orientation, which numerically means that the real yaw angle θ is miscalculated to $\theta \pm 180°$, though they can give relatively accurate value of bounding boxes' position and scale. This case often

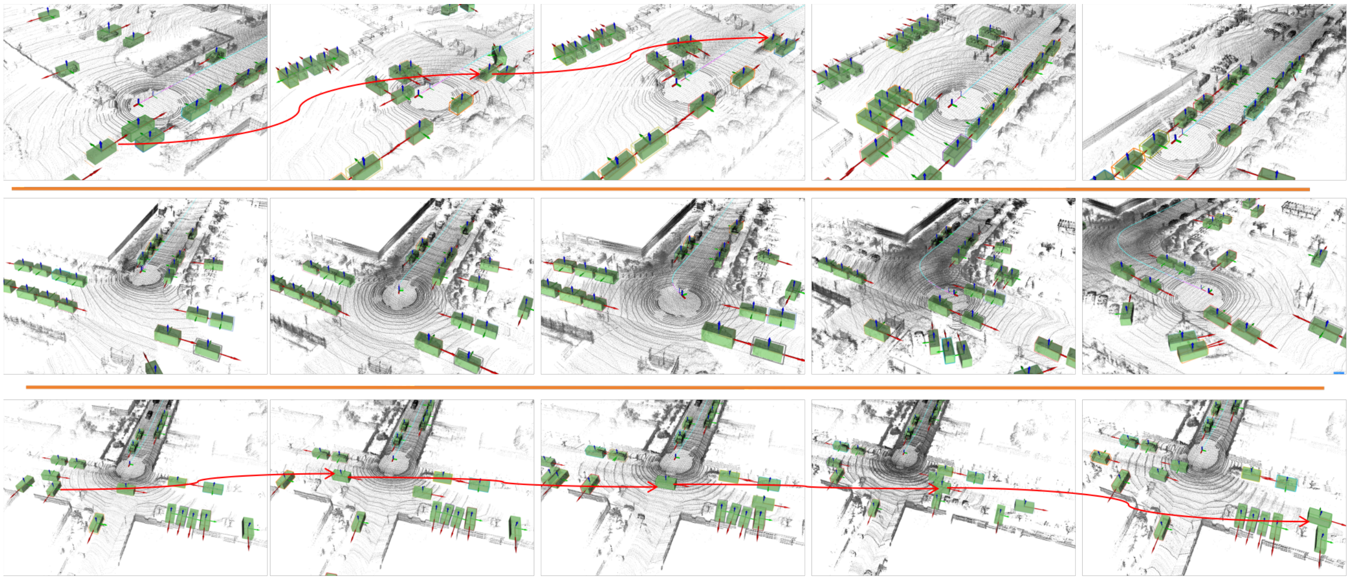

Fig. 3. The figure illustrates three distinct traffic scenarios. In the top panel, both the ego vehicle and surrounding vehicles are in motion on a highway. The middle panel depicts an intersection scenario where the ego vehicle is moving while the surrounding vehicles remain stationary. In the bottom panel, the ego vehicle is stationary, whereas the surrounding vehicles are in motion. The red arrows indicate the association of dynamic objects across different frames. The details of association tracking will be discussed in section III

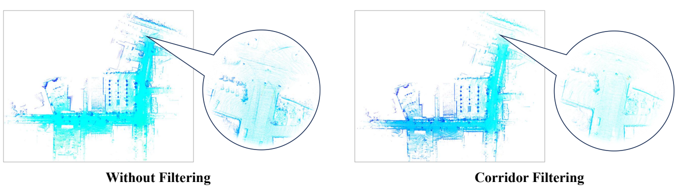

Fig. 4. The comparison of dynamic scene with(right) or without(left) spatial-temporal corridor filtering

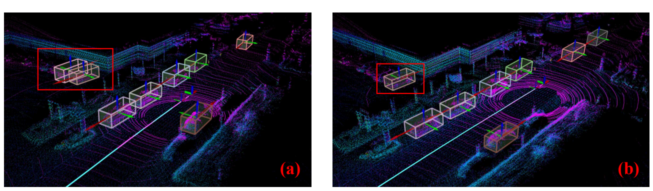

Fig. 5. The degeneration of detector in a parked car. In red boxes, it shows that the orientation of a car in (a) is opposite to that of its counterpart in (b) with the orientations of other reference objects remaining unchanged.

occurs when it comes to autonomous driving situations where the head and rear parts of surrounding cars are prone to confusion, as shown in Fig. 5.

The majority of multi-object tracking methods use IoU [28] [29] [30] [31] as their association standard. IoU-based methods naturally neglect the error caused by opposite orientation as they calculate the intersection over union independent of whether the yaw angle is θ or $\theta \pm 180°$. However, the computation of 3D IoU, which is equivalent to solving the polyhedron volume, is challenging to optimize even through simplification [32]. So we leverage multivariate normal normal distribution to represent 3D bounding box $\mathrm{BBox} = (x, y, z, l, w, h, \mathrm{yaw})$.

Following the three-sigma rule, in the object frame, the probability distribution object point $\mathbf{t}$ can be written as:

$$p(\mathbf{t}) = \mathcal{N}(\mathbf{0}, \mathrm{diag}(\frac{l}{3}, \frac{w}{3}, \frac{h}{3})) \tag{15}$$

After transformation to world frame using $[{}^{w}\mathbf{R}_{obj}, {}^{w}\mathbf{t}_{obj}] = [\mathrm{Exp}(\mathrm{yaw}), (x, y, z)]$, the distribution of points belonging to two different objects are changed to:

$$\begin{aligned} p_1(\mathbf{t}) &= \mathcal{N}({}^{w}_{1}\mathbf{t}_{obj}, {}^{w}_{1}\mathbf{R}_{obj}\mathrm{diag}(\frac{l_1}{3}, \frac{w_1}{3}, \frac{h_1}{3})\, {}^{w}_{1}\mathbf{R}^{T}_{obj}) \\ &= \mathcal{N}(\mathbf{t}_1, \mathbf{\Sigma}_1) \\ p_2(\mathbf{t}) &= \mathcal{N}({}^{w}_{2}\mathbf{t}_{obj}, {}^{w}_{2}\mathbf{R}_{obj}\mathrm{diag}(\frac{l_2}{3}, \frac{w_2}{3}, \frac{h_2}{3})\, {}^{w}_{2}\mathbf{R}^{T}_{obj}) \\ &= \mathcal{N}(\mathbf{t}_2, \mathbf{\Sigma}_2) \end{aligned} \tag{16}$$

In the world frame, we use divergence to measure the difference between these two normal distributions (NDD):

$$\begin{aligned} \mathrm{diff} &= \int [p_1(\mathbf{t}) - p_2(\mathbf{t})] \ln \frac{p_1(\mathbf{t})}{p_2(\mathbf{t})} d\mathbf{t} \\ &= \frac{1}{2}\mathrm{tr}(\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_2 + \mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1 - 2\mathbf{I}) \\ &\quad + \frac{1}{2}(\mathbf{t}_1 - \mathbf{t}_2)^T(\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1})(\mathbf{t}_1 - \mathbf{t}_2) \end{aligned} \tag{17}$$

Through this association strategy and error analytic expression, we make it efficient to associate objects between frames and perform optimization. The normal distribution representation method is mainly used for static object tracking since it is strict as association standard. As for high-speed objects, we still take pose difference based standard as association and optimization strategy in implementation.

E. Joint Optimization

The local structure of the joint factor graph is as Fig. 4. In the graph we add IMU pre-integration and LiDAR odometry factors like [3], as well as constant velocity factor, motion factor, dynamic and static object detection factor for object tracking. The optimization graph is shown as Fig. 7.

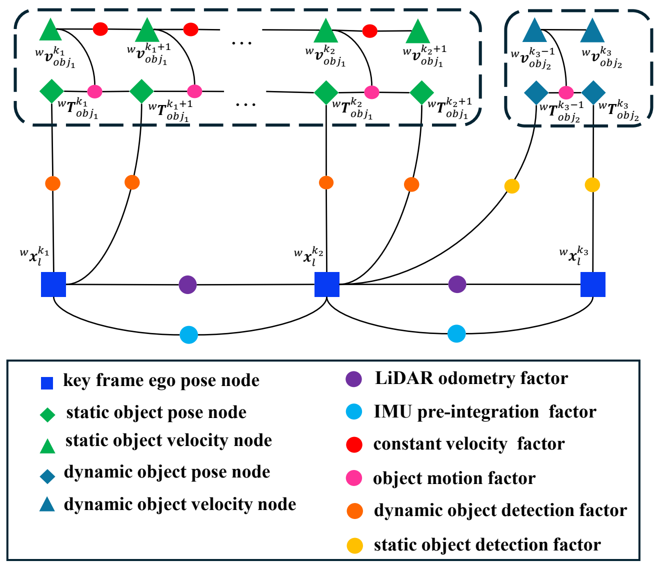

Fig. 6. The structure of factor graph. The GPS factors, loop closure factors and prior factors are omitted while the loosely-coupled and tightly-coupled factors are not distinguished in the graph showing.

According to Fig. 7, the formula of graph optimization can be mathematically expressed as Eq. 18, where the IMU pre-integration $\mathbf{r}_I$ factor and odometry factor $\mathbf{r}_O$ is added for self-localization. $\mathcal{I}, \mathcal{O}, \mathcal{V}, \mathcal{M}, \mathcal{D}_F, \mathcal{D}_S$ is edge sets relevant to factors. For factors related to object tracking, motion factor $\mathbf{r}_M$ constrain the motion of objects to follow the CLAV model, while constant velocity factors $\mathbf{r}_V$ ask that the velocity is unchanged in short time. Different from [7], we classify the detection factors as slow object detection factor $\mathbf{r}_S$ and fast object detection factor $\mathbf{r}_F$. The choice of these two detection factors is decided by object velocity provided by ESKF. The error calculation method corresponding to these two factors is explained in III-D. The covariance matrices $\mathbf{\Sigma}_O, \mathbf{\Sigma}_V, \mathbf{\Sigma}_M, \mathbf{\Sigma}_F$ are given as hyper-parameters while $\mathbf{\Sigma}_I$ is computed by the pre-integration forward propagation.

IV. Experiments

A. Ego Pose Estimation Results

We take translation and rotation parts of Absolute Pose Error (APE) $\mathrm{APE_T}$, $\mathrm{APE_R}$ to evaluate our system's global performance on ego pose estimation. In Table. I, the ground truth is given by the high precision OXTS integrated navigation module in KITTI. Both CenterPoint and SE-SSD object detectors are used to finish our experiments. The final result is the best value chosen from the results based on these two detectors.

It can be seen in Table. I that LIO-Track outperforms other methods in most sequence in both $\mathrm{APE_T}$ and $\mathrm{APE_R}$ standard. Quantitatively, LIO-Track is 3.72% and 2.0% better than LIO-SAM and LIO-SEGMOT on average in terms of $\mathrm{APE_T}$. On the other hand, LIO-Track suffers from sightly

TABLE I

The comparison of global ego pose estimation results on KITTI sequences.

Sequence	LIO-SAM		LIO-SEGMOT		LIO-Track	
	$\mathrm{APE_T}$	$\mathrm{APE_R}$	$\mathrm{APE_T}$	$\mathrm{APE_R}$	$\mathrm{APE_T}$	$\mathrm{APE_R}$
0926-0009	0.608	0.393	0.666	**0.391**	**0.566**	0.399
0926-0013	0.191	2.254	0.192	2.254	**0.178**	**2.253**
0926-0015	0.303	1.088	0.319	1.088	**0.294**	**1.083**
0926-0051	0.352	1.586	0.353	1.586	**0.352**	**1.585**
0926-0101	8.010	2.781	8.144	**2.771**	**7.868**	2.781
0930-0027	0.702	1.894	**0.550**	1.843	0.653	**1.839**

degradation in Seq. 0926-0051 and 0930-0027, which might account for the deterioration of the object detector.

While the Absolute Pose Error depicts the global error of pose, the Relative Pose Error (RPE) focus more on local accuracy and is more suitable for measuring the results of ego pose estimation with multi-object tracking as the above factor graph optimization can always be viewed as a local bundle adjustment question. As the full RPE results shown in Table. II, through robust local association with ego and tracking objects, LIO-Track obtains a precise local increment comparable to the state-of-the-art method LIO-SEGMOT [7] on average.

TABLE II

The comparison of local ego pose increment estimation results on KITTI sequences.

Sequence	LIO-SAM	LIO-SEGMOT	LIO-Track
0926-0009	1.114	**1.089**	1.112
0926-0013	1.594	1.760	**1.594**
0926-0015	1.834	**1.827**	1.836
0926-0051	0.722	0.721	**0.720**
0926-0101	2.114	**2.108**	2.116
0930-0027	1.594	0.779	**0.777**
Average	1.495	**1.381**	1.388

B. Multi-Object Tracking Results

We adopt index maps to qualitatively evaluate the performance of object tracking in large-scale scenes. An index map is constructed by accumulating the trajectories of tracked objects over time, with their corresponding tracking indices attached to each trajectory point. In this way, the index map provides a compact yet informative visualization that encodes both spatial and temporal consistency of the tracking process. By assigning distinct colors to different indices, the index map can clearly illustrate whether the trajectory associated with each object remains continuous and unique throughout the entire sequence, thereby serving as an intuitive diagnostic tool for assessing re-identification errors or trajectory fragmentation. For a more objective and quantitative evaluation, Table III reports the multi-object tracking performance under different module configurations using self-defined reference-free metrics. Since object-level ground-truth trajectories are not available for all evaluated sequences, we evaluate the temporal consistency and stability of tracking results by the average track length, detection-track matching ratio,

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \sum_{(\mathbf{x}_l^i, \mathbf{x}_l^{i+1}) \in \mathcal{I}} \| \mathbf{r}_I(\mathbf{x}_l^i, \mathbf{x}_l^{i+1}) \|^2_{\mathbf{\Sigma}_I} + \sum_{(\mathbf{T}_l^i, \mathbf{T}_l^{i+1}) \in \mathcal{O}} \| \mathbf{r}_O(\mathbf{T}_l^i, \mathbf{T}_l^{i+1}) \|^2_{\mathbf{\Sigma}_O} + \sum_{(\mathbf{v}_{obj}^i, \mathbf{v}_{obj}^{i+1}) \in \mathcal{V}} \| \mathbf{r}_V(\mathbf{v}_{obj}^i, \mathbf{v}_{obj}^{i+1}) \|^2_{\mathbf{\Sigma}_V}$$
$$+ \sum_{(\mathbf{T}_{obj}^i, \mathbf{T}_{obj}^{i+1}, \mathbf{v}_{obj}^i) \in \mathcal{M}} \| \mathbf{r}_M(\mathbf{T}_{obj}^i, \mathbf{T}_{obj}^{i+1}, \mathbf{v}_{obj}^i) \|^2_{\mathbf{\Sigma}_M} + \sum_{(\mathbf{T}_l^i, \mathbf{T}_{obj}^j) \in \mathcal{D}_F} \| \mathbf{r}_F(\mathbf{T}_l^i, \mathbf{T}_{obj}^j) \|^2_{\mathbf{\Sigma}_F} + \sum_{(\mathbf{T}_l^i, \mathbf{T}_{obj}^j) \in \mathcal{D}_S} r_S(\mathbf{T}_l^i, \mathbf{T}_{obj}^j) \quad (18)$$

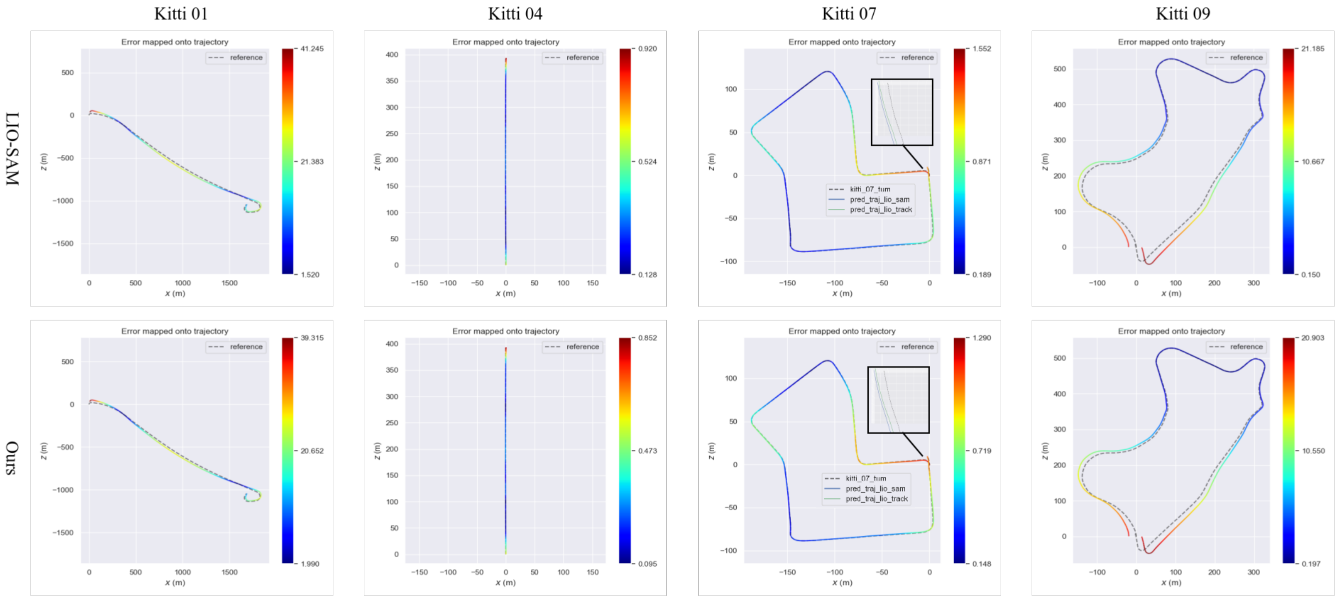

Fig. 7. As illustrated in the figure, since LIO-Track is built upon the same underlying framework, its trajectory accuracy does not show an absolute advantage over frameworks such as LIO-SAM and LIO-SEGMOT. However, by explicitly tracking neighboring objects and tightly integrating them into the factor graph, our method enhances the robustness of the localization algorithm in most scenarios and achieves smoother trajectory estimates.

TABLE III
ABLATION EVALUATION OF MULTI-OBJECT TRACKING

Methods	Seq.	Avg. Track Length ↑	Match Ratio ↑	Frag. Rate ↓	Jump Rate ↓	Size Instab. ↓
w/o ESKF, w/o NDD	0013	29.4	0.431	0.189	0.014	0.035
	0015	33.6	0.459	0.171	0.012	0.032
	0051	27.8	0.404	0.214	0.016	0.038
w/o ESKF, w/ NDD	0013	27.8	0.505	0.142	0.010	0.026
	0015	31.5	0.531	0.128	0.009	0.024
	0051	26.1	0.482	0.171	0.012	0.030
w/ ESKF, w/o NDD	0013	**35.6**	0.472	0.165	0.011	0.031
	0015	**39.4**	0.498	0.151	0.010	0.029
	0051	**33.2**	0.448	0.195	0.014	0.035
w/ ESKF, w/ NDD	0013	32.8	**0.563**	**0.124**	**0.008**	**0.022**
	0015	37.4	**0.591**	**0.109**	**0.007**	**0.019**
	0051	31.5	**0.527**	**0.151**	**0.010**	**0.026**

fragmentation rate, motion jump rate, and bounding-box size instability. Four variants are compared, including LIO-Track without ESKF and NDD, LIO-Track with only NDD, LIO-Track with only ESKF, and the full LIO-Track using both ESKF and NDD. As shown in Table III, ESKF tends to improve track continuity by providing better object-state initialization, while NDD improves association reliability by suppressing unstable detection associations. The full model achieves the best overall performance, with higher matching ratios, fewer fragmented trajectories, fewer motion jumps, and more stable bounding boxes across most sequences, although the average track length can be slightly shorter than the variant without NDD due to stricter association constraints.

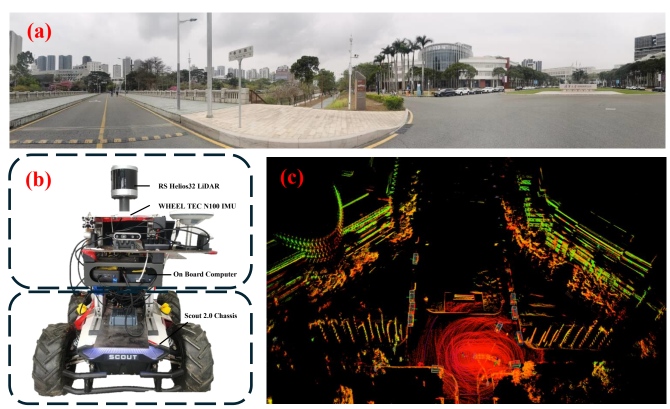

Fig. 8. The real-world experiment sketch. The panorama (a) is the scene where we collect data. (b) and (c) show our collecting device and the result of mapping and tracking.

C. Real-World Experiment

We validate the effectiveness of the proposed method on a self-collected dataset acquired in a campus environment. The data were collected on-site using a Robosense Helios 32-beam LiDAR integrated with a WHEEL TEC N100 IMU, which jointly provide high-resolution 3D spatial information and precise motion measurements. This campus dataset is characterized by a rich mixture of static and dynamic elements, including buildings, parked vehicles, vegetation, traffic signs, and moving pedestrians, thereby offering a challenging environment for simultaneous localization and mapping (SLAM). As illustrated in Fig. 8(a), the dataset captures diverse real-world conditions with significant structural complexity and object variability. The mapping and tracking results presented in Fig. 8(c) demonstrate that our method can faithfully reconstruct both static and dynamic structures, while maintaining consistent object associations across frames. These results highlight not only the robustness of our approach in handling dynamic environments but also its strong generalization capability to real-world scenarios beyond synthetic or benchmark datasets..

V. Conclusions

In this paper, we propose LIO-Track, a tightly-coupled LiDAR-Inertial Odometry with multi-object tracking. By taking dynamic objects into consideration, we develop a robust system to recognize and track objects while achieving accurate localization and mapping. We first design a novel ESKF for tracking objects. Considering there are the cases that tracking objects alternates between dynamic and static, we build a spatial-temporal corridor for each object to filter out dynamic points based on spatial and temporal information. To get a more stable association criterion, we take normal distribution to represent object spatial occupancy and leverage divergence to calculate the similarity between objects. We classify the tracking objects as slow or fast ones and add them to the hierarchical factor graph to perform optimization. Experiments on both simulation and the real world demonstrate the effectiveness of our proposed method. The framework and source code will be released to the community to facilitate further research and development.

References

[1] 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.

[2] 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.

[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 *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE transactions on robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[5] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[6] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[7] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 616–10 622.

[8] J. Zhang, M. Henein, R. Mahony, and V. Ila, "Vdo-slam: A visual dynamic object-aware slam system," *arXiv preprint arXiv:2005.11052*, 2020.

[9] Y. Jia, T. Wang, X. Chen, and S. Shao, "Trlo: An efficient lidar odometry with 3d dynamic object tracking and removal," *arXiv preprint arXiv:2410.13240*, 2024.

[10] Z. Zhu, J. Zhao, K. Huang, X. Tian, J. Lin, and C. Ye, "Limot: a tightly-coupled system for lidar-inertial odometry and multi-object tracking," *IEEE Robotics and Automation Letters*, 2024.

[11] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Y. Cai, W. Xu, and F. Zhang, "ikd-tree: An incremental kd tree for robotic applications," *arXiv preprint arXiv:2102.10808*, 2021.

[17] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[18] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE international symposium on mixed and augmented reality (ISMAR)*. IEEE, 2018, pp. 10–20.

[19] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*. Ieee, 2000, pp. 153–158.

[20] L. Sun, J. Zhao, X. He, and C. Ye, "Dlo: Direct lidar odometry for 2.5 d outdoor environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–5.

[21] X. Li, Z. Yan, S. Feng, C. Xia, S. Li, and Y. Zhou, "Lio-lot: Tightly-coupled multi-object tracking and lidar-inertial odometry," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[22] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

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

[25] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," 1995.

[26] H. Lim, S. Hwang, and H. Myung, "Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021.

[27] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4530–4537.

[28] X. Weng, J. Wang, D. Held, and K. Kitani, "Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics," *arXiv preprint arXiv:2008.08063*, 2020.

[29] X. Li, T. Xie, D. Liu, J. Gao, K. Dai, Z. Jiang, L. Zhao, and K. Wang, "Poly-mot: A polyhedral framework for 3d multi-object tracking," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9391–9398.

[30] K. Pauwels and D. Kragic, "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1300–1307.

[31] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *European conference on computer vision*. Springer, 2022, pp. 680–696.

[32] J. Xu, Y. Ma, S. He, and J. Zhu, "3d-giou: 3d generalized intersection over union for object detection in point cloud," *Sensors*, vol. 19, no. 19, p. 4093, 2019.