2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Semantic-LiDAR-Inertial-Wheel Odometry Fusion for Robust Localization in Large-Scale Dynamic Environments

Haoxuan Jiang, Peicong Qian, Yusen Xie, Linwei Zheng, Xiaocong Li,
Ming Liu, *Fellow, IEEE,* and Jun Ma, *Senior Member, IEEE*

***Abstract*— Reliable, drift-free global localization presents significant challenges yet remains crucial for autonomous navigation in large-scale dynamic environments. In this paper, we introduce a tightly-coupled Semantic-LiDAR-Inertial-Wheel Odometry fusion framework, which is specifically designed to provide high-precision state estimation and robust localization in large-scale dynamic environments. Our framework leverages an efficient semantic-voxel map representation and employs an improved scan matching algorithm, which utilizes global semantic information to significantly reduce long-term trajectory drift. Furthermore, it seamlessly fuses data from LiDAR, IMU, and wheel odometry using a tightly-coupled multi-sensor fusion Iterative Error-State Kalman Filter (iESKF). This ensures reliable localization without experiencing abnormal drift. Moreover, to tackle the challenges posed by terrain variations and dynamic movements, we introduce a 3D adaptive scaling strategy that allows for flexible adjustments to wheel odometry measurement weights, thereby enhancing localization precision. This study presents extensive real-world experiments conducted in a one-million-square-meter automated port, encompassing 3,575 hours of operational data from 35 Intelligent Guided Vehicles (IGVs). The results consistently demonstrate that our system outperforms state-of-the-art LiDAR-based localization methods in large-scale dynamic environments, highlighting the framework's reliability and practical value.**

I. INTRODUCTION

Global localization in large-scale dynamic environments remains a formidable challenge, particularly in settings like ports and industrial parks. Traditional localization systems often depend on signal-based methods like the Global Positioning System (GPS) or WiFi. However, these approaches can be unreliable in scenarios where stable signal delivery is disrupted, such as during signal loss or transmission interruptions. In recent years, Simultaneous Localization and Mapping (SLAM) [1], [2] has witnessed rapid advancements in high-precision localization, which becomes indispensable for industrial applications of robots and autonomous systems.

Sensor fusion-based SLAM methods typically achieve localization by aligning scanned frames with pre-constructed maps, but they often suffer from abnormal localization drift. To mitigate this, some advanced approaches incorporate continuous odometry estimation within localization frameworks. These methods integrate IMU data, LiDAR scans, and wheel odometry into a joint localization framework, yielding improved results. However, the absence of semantic information still makes them vulnerable to long-term drift. In conventional map representation, they typically rely on explicit feature representations like point clouds [3], [4], surfels [5] , and voxels [6], [7]. While effective in static environments, these methods struggle with sparse observations, occlusions, and dynamic scenes, frequently failing to distinguish dynamic objects from static backgrounds. Moreover, they lack semantic information, leading to challenges such as long-term drift and difficulties in re-localization upon system restart. Additionally, traditional wheel odometry [8], [9] encounters difficulties in complex terrains, such as slopes or slippery surfaces, where tire slippage or poor ground contact results in inaccurate motion estimation.

To address aforementioned issues, this paper proposes a tightly-coupled semantic-LiDAR-inertial-wheel odometry fusion framework designed to significantly enhance localization accuracy and robustness in dynamic, large-scale environments. The framework incorporates LiDAR, IMU, and wheel odometry data within a tightly coupled multi-sensor fusion scheme based on the iterative error state Kalman filter (iESKF) [10], [11], which effectively mitigates motion distortion and abnormal localization drift. Our framework employs a voxel-based semantic matching algorithm that tightly integrates semantic information with spatial geometric features to enhance environmental understanding and reduce long-term trajectory drift. By extracting semantic labels from LiDAR point clouds and mapping them onto a voxelized map, the algorithm effectively distinguishes dynamic objects from static environmental features, minimizing the impact of dynamic objects on localization results while maximizing the contribution of static objects to improve localization accuracy and stability. Furthermore, a 3D adaptive scaling strategy is proposed to address errors due to complex terrains. This strategy dynamically adjusts the weight of wheel speed observations based on motion states and terrain features, optimizing wheel odometry performance and ensuring improved adaptability across diverse and challenging terrains.

Haoxuan Jiang, Yusen Xie, and Linwei Zheng are with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China (e-mail: hjiangax@connect.hkust-gz.edu.cn; yxie827@connect.hk-gz.edu.cn; lzhengad@connect.ust.hk).

Peicong Qian is with Shenzhen Unity Drive Innovation Technology Co., Ltd., Shenzhen 518063, China (e-mail: epsilonjohn9527@gmail.com).

Xiaocong Li is with the College of Information Science and Technology, Eastern Institute of Technology, Ningbo, Ningbo 315200, China (e-mail: xiaocongli@eitech.edu.cn).

Ming Liu is with the Research & Development Institute of Northwestern Polytechnical University, Shenzhen 518063, China (e-mail: liu.ming.prc@gmail.com).

Jun Ma is with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China, and also with the Cheng Kar-Shun Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong SAR, China (e-mail: jun.ma@ust.hk). *(Corresponding Author: Jun Ma.)*

In summary, the main contributions of this paper are as follows:

- We propose an iESKF-based semantic-LiDAR-inertial-wheel odometry fusion framework, effectively integrating LiDAR, IMU, and wheel odometry data to mitigate motion distortion and abnormal localization drift.
- We introduce a semantic voxel-based matching algorithm that integrates LiDAR semantic labels with spatial geometric features to distinguish diverse dynamic objects from static environments, effectively reducing the long-term trajectory drift.
- We present a 3D adaptive scaling strategy to address errors caused by complex terrains, which dynamically adjusts wheel speed observation weights based on motion states and terrain features.
- Our algorithm has been successfully deployed in a large-scale automated port, delivering precise and stable localization for 35 IGVs, demonstrating its reliability and effectiveness in complex real-world applications.

II. RELATED WORK

A. Efficient Map Representation

Map representation and scan registration serve as the fundamental basis for ensuring reliable localization in a LiDAR-based odometry or SLAM system. Explicit mapping directly utilize geometric data collected by sensors for map construction. Point cloud representations (e.g., CT-ICP [3], KISS-ICP [4]) directly utilize sensor data, offering high precision and intuitive geometric characteristics. However, they require significant storage and have low query efficiency, limiting their deployment in large-scale or real-time applications. Surfel representations (e.g., SuMa++ [5]) store local surface information, such as points and normals, reducing storage requirements and improving query efficiency while retaining some geometric details. However, they struggle with adaptability in sparse or dynamic environments, often losing fine details. Voxel-based representations (e.g., LiTA-MIN2 [6], Voxel-SLAM [7]) leverage spatial partitioning for efficient querying and updating, making them suitable for dynamic and large-scale environments. However, their accuracy is determined by voxel resolution: low resolutions result in a loss of detail, while high resolutions greatly increase computational costs.

To enhance the efficiency of processing and storing explicit map representations, various advanced data structures have been developed. Incremental KD-Tree (iKD-Tree) [11], [12] enables fast point cloud updates and efficient nearest neighbor queries in dynamic environments. Incremental Voxel Map (iVox) from Faster-LIO [13] updates voxel grids incrementally, integrating probabilistic and geometric data for accurate, robust mapping and efficient storage management. Voxelized Generalized Iterative Closest Point (GICP) [14] and Voxel-based Surface Covariance Estimator (VSCE) from iG-LIO [15] leverage voxelized point distributions to robustly estimate surface covariances, avoiding costly nearest neighbor searches. Voxelized GICP enhances the capabilities of GICP [16] through voxelization, maintaining reliability in sparse data scenarios and supporting efficient parallel optimization. Similarly, VSCE improves efficiency in processing dense scans while maintaining high accuracy in sparse and small field-of-view scenarios, and also enabling parallel optimization. However, these methods primarily focus on geometric information and struggle to effectively distinguish between static and dynamic objects, often leading to localization drift in dynamic environments.

B. Multi-Sensor Fusion based Localization

Based on reliable and efficient mapping, LiDAR-based odometry and SLAM methods utilize sensors like LiDAR, IMU, and wheel odometry to achieve highly precise localization. Within the LiDAR-inertial odometry (LIO) framework, FAST-LIO [10] employs a tightly-coupled Kalman filter to reduce computational overhead and correct motion distortion, enabling robust navigation in dynamic environments. Building on this, FAST-LIO2 [11] improves accuracy through raw point-to-map registration and boosts efficiency by utilizing iKD-Tree [12] for efficient map management and querying. Faster-LIO [13] further improves performance by replacing iKD-Tree [12] with iVox for faster updates and approximate k-NN queries, avoiding complex tree operations. Point-LIO [17] adopts a point-by-point framework for high-frequency odometry updates, effectively removing motion distortion, and introduces a stochastic process-augmented kinematic model for accurate localization during aggressive motions, even in scenarios with IMU saturation. Beyond standard LIO frameworks, methods like EKF-LOAM [8], LIWO [9], and LIWOM-GD [18] integrate additional wheel odometry data to further enhance state estimation. EKF-LOAM enhances LeGO-LOAM [19] by employing an adaptive Extended Kalman Filter (EKF) with a lightweight covariance scheme to improve path estimation in feature-sparse environments, while LIWO utilizes a tightly-coupled bundle adjustment (BA) framework to achieve more accurate velocity estimation and effectively mitigate IMU drift. LIWOM-GD [18] further incorporates an advanced dynamic point removal technique, global plane constraints and loop closure, all within a refined factor graph optimization framework. These works show satisfactory performance in static environments. However, their effectiveness declines considerably in dynamic and repetitive settings, particularly when encountering sparse observations and unforeseen collisions.

III. METHODOLOGY

As illustrated in Fig. 1, our framework comprises several key modules: system backbone in Sec. III-A, semantic voxel mapping pipeline in Sec. III-B, motion dynamics prediction model in Sec. III-C, LiDAR pose estimation model in Sec. III-D, IMU correction model in Sec. III-E, and wheel odometry correction model in Sec. III-F.

A. System Description

Our localization system, built on the mapping module, uses a tightly coupled iESKF-based multi-sensor fusion approach [10], [11] to enhance accuracy and reliability.

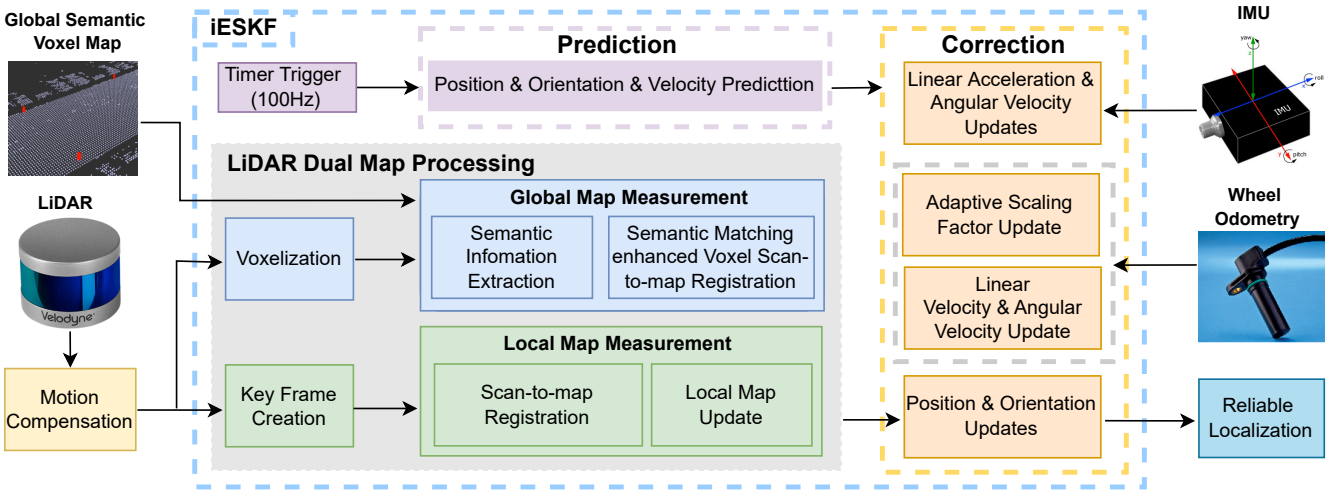


Fig. 1: The diagram provides a detailed overview of our high-precision localization system architecture based on the **iESKF** filter, which integrates data from LiDAR, IMU, and wheel odometry, alongside a global voxel map enriched with semantic information for robust state estimation. The workflow is divided into four key modules: **LiDAR Dual Map Processing**, **Prediction**, and **Correction**.

Firstly, we define the state vector $\mathbf{x}$ as

$$\mathbf{x} \triangleq \begin{bmatrix} {}^{W}\mathbf{p}_I & {}^{W}\mathbf{v}_I & {}^{W}\mathbf{R}_I & \mathbf{b_a} & \mathbf{b}_{\boldsymbol{\omega}} & {}^{B}\mathbf{a} & {}^{B}\boldsymbol{\omega} \\ & {}^{v}\mathbf{S} & {}^{W}\mathbf{g} & {}^{I}\mathbf{R}_L & {}^{I}\mathbf{p}_L & {}^{I}\mathbf{R}_B & {}^{I}\mathbf{p}_B \end{bmatrix} \tag{1}$$

where ${}^{W}\mathbf{p}_I$, ${}^{W}\mathbf{v}_I$, and ${}^{W}\mathbf{R}_I$ denote the IMU position, velocity and orientation in the world frame W. ${}^{B}\mathbf{a}$ and ${}^{B}\boldsymbol{\omega}$ are the linear acceleration and angular velocity in the wheel odometry frame B, while $\mathbf{b_a}$ and $\mathbf{b}_{\boldsymbol{\omega}}$ are the corresponding biases. ${}^{v}\mathbf{S}$ represents the adaptive scaling factor for linear velocity part of wheel odometry. ${}^{W}\mathbf{g}$ is the gravity vector in the world frame W. ${}^{I}\mathbf{R}_L$ and ${}^{I}\mathbf{p}_L$ are the extrinsic parameters between IMU frame I and LiDAR frame L, respectively. ${}^{I}\mathbf{R}_B$ and ${}^{I}\mathbf{p}_B$ are the extrinsic parameters between IMU frame I and wheel odometry frame B, respectively.

Using the notations $\boxplus$ and $\boxminus$ as defined in FAST-LIO [10], the state transition model in the Lie algebra space can be defined as:

$$\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus (\Delta t\, \mathbf{f}(\mathbf{x}_i, \mathbf{w}_i)) \tag{2}$$

with the function $\mathbf{f}$ in forward process derived as

$$\mathbf{f}(\mathbf{x}, \mathbf{w}) \triangleq \begin{bmatrix} {}^{W}\mathbf{v}_I + \frac{1}{2}\left({}^{W}\mathbf{R}_I {}^{I}\mathbf{R}_B {}^{B}\mathbf{a} + {}^{W}\mathbf{g}\right)\Delta t \\ {}^{W}\mathbf{R}_I {}^{I}\mathbf{R}_B {}^{B}\mathbf{a} + {}^{W}\mathbf{g} \\ {}^{I}\mathbf{R}_B {}^{B}\boldsymbol{\omega} \\ \mathbf{n_{ba}} \\ \mathbf{n}_{\mathbf{b}\boldsymbol{\omega}} \\ \mathbf{n_a} \\ \mathbf{n}_{\boldsymbol{\omega}} \\ \mathbf{0}_{18\times 1} \end{bmatrix} \tag{3}$$

Within the system, IMU observation data is transformed from the IMU frame I to the wheel odometry frame B before being processed by the filter. Consequently, the IMU's angular velocity and linear acceleration measurement process noise, denoted as ${}^{B}\mathbf{w}$, is directly expressed in the wheel odometry frame B, as shown in (4). Here, $\mathbf{n_a}$ and $\mathbf{n}_{\boldsymbol{\omega}}$ denote the measurement noise of the IMU's linear acceleration and angular velocity, respectively, in the wheel odometry frame B, both of which can be modeled as Gaussian white noise. Meanwhile, the IMU measurement bias noise terms $\mathbf{n_{ba}}$ and $\mathbf{n}_{\mathbf{b}\boldsymbol{\omega}}$, corresponding to the noise components of linear acceleration bias $\mathbf{b_a}$ and angular velocity bias $\mathbf{b}_{\boldsymbol{\omega}}$ in the wheel odometry frame B, can be modeled as random walk processes using the following equations:

$$\begin{aligned} {}^{B}\mathbf{w} &\triangleq \begin{bmatrix} \mathbf{n_a} & \mathbf{n}_{\boldsymbol{\omega}} & \mathbf{n_{ba}} & \mathbf{n}_{\mathbf{b}\boldsymbol{\omega}} \end{bmatrix} \\ \mathbf{n}_a &\sim \mathcal{N}(0, \sigma_a^2), \quad \mathbf{n}_\omega \sim \mathcal{N}(0, \sigma_\omega^2). \\ \mathbf{n}_{ba} &\sim \mathcal{N}(0, \sigma_{b_a}^2), \quad \mathbf{n}_{b\omega} \sim \mathcal{N}(0, \sigma_{b_\omega}^2). \end{aligned} \tag{4}$$

B. Map Construction and Update

The mapping module utilizes a dual-map architecture, as demonstrated in Fig. 1. The global map provides a stable long-term reference for consistent and reliable localization, while the local dynamic map focuses on rapid adaptation to environmental changes, offering real-time support in dynamic scenarios. This hierarchical structure combines coarse alignment via the global map and fine alignment with the local map, ensuring accurate, efficient, and reliable localization across diverse conditions.

Map Construction. In constructing pre-existing global maps, Fast-LIO2 [11] incorporates loop closure and GPS factors for back-end optimization (inspired by the LIO-SAM [20] framework), ensuring precise alignment and high consistency of point cloud data. By detecting and removing dynamic elements while retaining only static point cloud data, it effectively prevents localization drift caused by environmental changes or moving objects. For local dynamic maps, key frames capturing core point cloud data are generated based on significant environmental changes to reduce redundancy and improve update efficiency. Then, corrected LiDAR poses are used to integrate multi-frame key frames, incrementally updating the map based on map size and distance thresholds to quickly adapt to environmental changes and ensure short-term localization accuracy.

Voxel-Based Storage. A voxel-based point cloud storage format (referencing VSCE in iG-LIO [15]) divides the space

into fixed 0.5-meter voxel grids, with each voxel retaining a representative point, characterized by the mean coordinates and distribution variance of points within the voxel, to significantly reduce data redundancy, improve registration efficiency, and support fast neighborhood queries and dynamic updates.

Global Semantic Information Integration. Building on dynamic object removal and voxelization, semantic features are extracted from the voxelized global map using Algorithm 1. This enables the system to leverages high-level semantic environmental information for more reliable global point cloud registration and alignment. Meanwhile, to enhance the robustness and accuracy of point cloud registration, especially in handling degenerate scenarios or noisy point clouds, we introduced specific constraint strategies for different semantic feature voxels. For cylinder semantic feature voxels, a minimum eigenvalue constraint is applied to improve numerical stability and prevent optimization failure caused by degenerate directions. For plane and other semantic feature voxels, a plane eigenvalue constraint is introduced to enhance registration accuracy in planar regions and mitigate errors caused by planar feature degeneration.

C. Model Prediction

In our prediction model, the updated state $\bar{\mathbf{x}}_i$ at time step i can be propagated to the next time step $i+1$ using the state transition model described in (2), with $\mathbf{w}_i$ set to zero:

$$\hat{\mathbf{x}}_{i+1} = \bar{\mathbf{x}}_i \boxplus (\Delta t\, \mathbf{f}(\bar{\mathbf{x}}_i, 0)) \tag{5}$$

In greater detail, the state can be propagated using the following equations:

$$\begin{aligned} {}^{W}\hat{\mathbf{p}}_I^{\,i+1} &= {}^{W}\bar{\mathbf{p}}_I^{\,i} + {}^{W}\bar{\mathbf{v}}_I^{\,i}\,\Delta t + \frac{1}{2}\left({}^{W}\bar{\mathbf{R}}_I^{\,i}\,{}^{I}\bar{\mathbf{R}}_B^{\,i}\,{}^{B}\bar{\mathbf{a}}^i + {}^{W}\bar{\mathbf{g}}^i\right)\Delta t^2 \\ {}^{W}\hat{\mathbf{v}}_I^{\,i+1} &= {}^{W}\bar{\mathbf{v}}_I^{\,i} + \left({}^{W}\bar{\mathbf{R}}_I^{\,i}\,{}^{I}\bar{\mathbf{R}}_B^{\,i}\,{}^{B}\bar{\mathbf{a}}^i + {}^{W}\bar{\mathbf{g}}^i\right)\Delta t \\ {}^{W}\hat{\mathbf{R}}_I^{\,i+1} &= {}^{W}\bar{\mathbf{R}}_I^{\,i}\,\mathbf{Exp}\left({}^{I}\bar{\mathbf{R}}_B^{\,i}\,{}^{B}\bar{\boldsymbol{\omega}}^i\,\Delta t\right) \end{aligned} \tag{6}$$

Rather than relying on raw IMU measurements, state propagation can be achieved by leveraging estimated linear acceleration and angular velocity and assuming these values remain constant between observation frames. In contrast to pure IMU pre-integration methods, which depend on continuous high-frequency inertial data and are vulnerable to filter instability or divergence during data interruptions, this approach demonstrates strong resilience against IMU failures, data gaps, and synchronization delays.

Then, the propagation of covariance $\bar{\mathbf{P}}_i$ from time step i to $i+1$ can be implemented as follows:

$$\hat{\mathbf{P}}_{i+1} = \mathbf{F}_{\mathbf{x}_i}\bar{\mathbf{P}}_i\mathbf{F}_{\mathbf{x}_i}^{\top} + \mathbf{F}_{\mathbf{w}_i}\mathbf{Q}_i\mathbf{F}_{\mathbf{w}_i}^{\top} \tag{7}$$

where $\mathbf{Q}_i$ is the covariance of the IMU measurement process noise $\mathbf{w}_i$, and the matrices $\mathbf{F}_{\bar{\mathbf{x}}_i}$ and $\mathbf{F}_{\mathbf{w}_i}$ can be calculated as below:

$$\begin{aligned} \mathbf{F}_{\mathbf{x}_i} &= \left.\frac{\partial(\mathbf{x}_{i+1} \boxminus \hat{\mathbf{x}}_{i+1})}{\partial \delta\mathbf{x}_i}\right|_{\delta\mathbf{x}_i=0,\mathbf{w}_i=0} \\ \mathbf{F}_{\mathbf{w}_i} &= \left.\frac{\partial(\mathbf{x}_{i+1} \boxminus \hat{\mathbf{x}}_{i+1})}{\partial \mathbf{w}_i}\right|_{\delta\mathbf{x}_i=0,\mathbf{w}_i=0} \\ &\text{where} \quad \delta\mathbf{x}_i = \mathbf{x}_i \boxminus \hat{\mathbf{x}}_i \end{aligned} \tag{8}$$

Algorithm 1 Voxel-based Semantic Information Extraction and Covariance Refinement

Input: Voxel map $\mathcal{M}$ with mean μ_g and covariance Σ_g for each grid, angle threshold θ_{thc} for **cylinder**, angle threshold θ_{thp} for **plane**.
Output: Voxel map $\mathcal{M}$ with semantic information **cylinder**, **plane**, or **other** for each grid.

for each grid $g \in \mathcal{M}$ **do**
  Perform SVD on Σ_g: $\Sigma_g = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$;
  Extract the singular values σ_1, σ_2, σ_3 from $\mathbf{\Sigma}$, along with their corresponding singular vectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$ from $\mathbf{U}$, ensuring the singular values satisfy the condition $\sigma_1 \geq \sigma_2 \geq \sigma_3$;
  Set grid g as **other** by default;
  if $\sigma_1 \gg \sigma_2$ **and** $\sigma_2 \approx \sigma_3$ **then**
    Compute angle $\theta = \arccos\left(\frac{\mathbf{u}_1 \cdot \mathbf{z}}{\|\mathbf{u}_1\|\|\mathbf{z}\|}\right)$, where $\mathbf{z}$ is the unit vector of the z-axis;
    if $\theta < \theta_{thc}$ **then**
      Classify grid g as **cylinder**;
    end if
  else if $\sigma_1 \approx \sigma_2$ **and** $\sigma_3 \ll \sigma_1$ **then**
    Compute angle $\theta = \arccos\left(\frac{\mathbf{u}_3 \cdot \mathbf{z}}{\|\mathbf{u}_3\|\|\mathbf{z}\|}\right)$, where $\mathbf{z}$ is the unit vector of the z-axis;
    if $\theta < \theta_{thp}$ **then**
      Classify grid g as **plane**;
    end if
  end if
  if grid g is **cylinder then**
    Compute $\sigma_i = \max(\sigma_i, 1\text{e-}3)$ for the singular values σ_1, σ_2, σ_3
    Set the diagonal matrix $\mathbf{D} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)$;
  else if grid g is **plane or** grid g is **other then**
    Set the diagonal matrix $\mathbf{D} = \mathrm{diag}(1, 1, 1\text{e-}3)$;
  end if
  Update the covariance of grid g: $\Sigma_g = \mathbf{U}\mathbf{D}\mathbf{V}^{\top}$;
end for

D. LiDAR Dual Map Processing

The LiDAR dual-map processing module consists of global map measurement and local map measurement, which handle global and local point cloud data respectively, working together to achieve precise and efficient localization. First, we define some symbols. ${}^{L_i}\mathbf{p}_j$ and ${}^{L_i}\mathbf{n}_j$ represent the LiDAR point j in current i-th frame of point cloud and its noise, and we assume that this noise is affected by zero-mean Gaussian white noise. ${}^{\mathcal{M}}\mathbf{T}_I{}^{i}$ is the i-th estimate of transformation between the global map frame $\mathcal{M}$ and the IMU frame $\mathbf{I}$. ${}^{I}\mathbf{T}_L{}^{i}$ is the i-th estimate of transformation between the IMU frame $\mathbf{I}$ and the LiDAR frame $\mathbf{L}$.

Global Map Measurement. The global pose can be estimated by matching the current point cloud with the global voxel-based prior map $\mathcal{M}_{global}$, ensuring long-term localization stability. First, the current point cloud is voxelized, dividing it into 0.5-meter voxel units to effectively reduce data volume while preserving the spatial structure of the

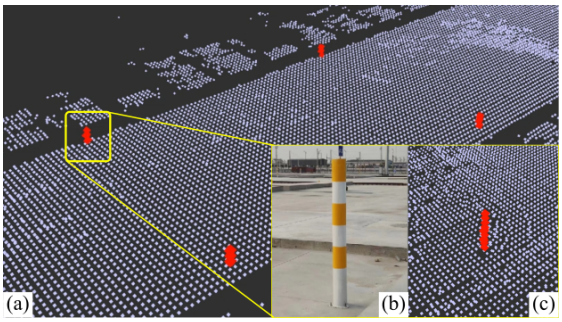


Fig. 2: (a) Example of deployment of pillars. Pillars are strategically deployed along both sides of the lane, with a spacing interval of 35 m between each pillar. (b) The pillar features a diameter of at least 20 cm and a minimum height of 1.5 m. (c) The point cloud representation of the pillars is highlighted in red on the map, while the background point cloud is shown in white.

environment, thereby improving computational efficiency. During the voxel-based scan-to-map registration phase, the voxelized current point cloud is aligned with the global map using the K-nearest neighbor voxel search method and GICP registration [16], as described below, for pose estimation:

$$ {}^{\mathcal{M}_{global}}\mathbf{R}_j\left(\mathbf{x}_i, {}^{L_i}\mathbf{p}_j, {}^{L_i}\mathbf{n}_j\right) = \mathbf{Vox}\left({}^{\mathcal{M}}\mathbf{q}_j\right) - \mathbf{Vox}\left({}^{\mathcal{M}}\mathbf{T}_I\, {}^{I}\mathbf{T}_L\, {}^{i}\left({}^{L_i}\mathbf{p}_j + {}^{L_i}\mathbf{n}_j\right)\right) \tag{9} $$

where **Vox** represents the voxelization operation applied to a point in the map, returning the mean value of the corresponding voxel. ${}^{\mathcal{M}}\mathbf{q}_j$ is the associated point in the global map.

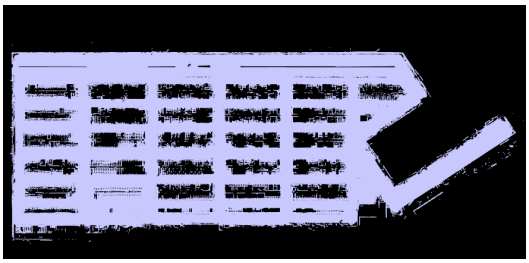
Since voxels with semantic feature labels are extracted from the pre-existing map using Algorithm 1, different weights are assigned to these voxels during the computation of constraints. Specifically, w_{cylinder}, w_{plane}, and w_{other} correspond to voxels with cylinder, plane, and other semantic features, respectively, and the weights satisfy the relationship $w_{\text{cylinder}} > w_{\text{plane}} > w_{\text{other}}$. Then, based on the semantic label of the **voxel** that the LiDAR point j in the current i-th frame of the point cloud belongs to, the weight w_j for this point j can be determined using an indicator function $\mathbb{I}$, as shown below:

$$ \begin{aligned} w_j = {} & w_{\text{cylinder}} \cdot \mathbb{I}\{\mathbf{voxel} \in cylinder\} \\ & + w_{\text{plane}} \cdot \mathbb{I}\{\mathbf{voxel} \in plane\} \\ & + w_{\text{other}} \cdot \mathbb{I}\{\mathbf{voxel} \in other\} \end{aligned} \tag{10} $$

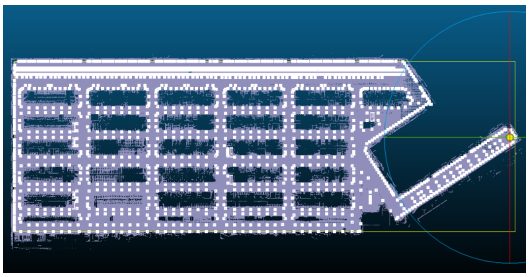
$$ \text{where } \mathbb{I}\{\text{condition}\} = \begin{cases} 1, & \text{if condition is true,} \\ 0, & \text{otherwise.} \end{cases} $$

Local Map Measurement. By concentrating on the immediate vicinity, the local map enables faster and more efficient scan-to-map registration, reducing the computational load of the global map. Meanwhile, the local map is dynamically updated to capture changes like moving objects or temporary obstacles, enhancing robustness in cluttered environments. The current point cloud is matched to this dynamically updated local voxel map $\mathcal{M}_{local}$ for relative pose calculation using the KNN search algorithm and point-to-plane registration, as detailed below:

$$ {}^{\mathcal{M}_{local}}\mathbf{R}_j\left(\mathbf{x}_i, {}^{L_i}\mathbf{p}_j, {}^{L_i}\mathbf{n}_j\right) = {}^{\mathcal{M}}\mathbf{u}_j^T\left({}^{\mathcal{M}}\mathbf{T}_I\, {}^{I}\mathbf{T}_L\, {}^{i}\left({}^{L_i}\mathbf{p}_j + {}^{L_i}\mathbf{n}_j\right) - {}^{\mathcal{M}}\mathbf{q}_j\right) \tag{11} $$



(a)



(b)

Fig. 3: Global voxelized point cloud map with semantic information. (a) The generated point cloud map spans an area of 1538 m × 596 m. (b) The global semantic point cloud highlights pillar points, which are projected onto the global map with an enlarged view to emphasize their positions. The pillar points are represented in white, while the background global point cloud map is displayed in light purple.

where ${}^{\mathcal{M}}\mathbf{u}_j$ is the normal vector of the associated plane fitted using neighboring points of ${}^{L_i}\mathbf{p}_j$ in the local map. ${}^{\mathcal{M}}\mathbf{q}_j$ is another point on the associated fitted plane in the local map.

Overall Map Constraints. Rather than directly registering point clouds with the map, we integrate feature alignment into the iESKF measurement update through corresponding constraints. This process is expressed by the following equation:

$$ \mathbf{0} = w_j \cdot {}^{\mathcal{M}_{global}}\mathbf{R}_j\left(\mathbf{x}_i, {}^{L_i}\mathbf{p}_j, {}^{L_i}\mathbf{n}_j\right) + {}^{\mathcal{M}_{local}}\mathbf{R}_j\left(\mathbf{x}_i, {}^{L_i}\mathbf{p}_j, {}^{L_i}\mathbf{n}_j\right) \tag{12} $$

This approach combines high-accuracy registration of the global map $\mathcal{M}_{\text{global}}$ with the real-time matching of the local map $\mathcal{M}_{\text{local}}$ within a single frame. The weights w_j are calculated using (10) and assigned to voxels representing cylinders, planes, and other semantic features.

E. IMU-based Correction

The IMU is an indispensable sensor in the system, providing high-frequency linear acceleration and angular velocity data. In this system, we treat IMU data as a measurement for filter correction in the wheel odometry frame B. Then, for the i-th frame of IMU measurement data, we can denote the constraints as:

$$ \begin{aligned} \mathbf{R}\left(\mathbf{x}_i, {}^{I}\boldsymbol{\omega}_m^i, \mathbf{n}_{\boldsymbol{\omega}}^i\right) &= \left({}^{I}\mathbf{R}_B^i\right)^{-1}\, {}^{I}\boldsymbol{\omega}_m^i - \mathbf{n}_{\boldsymbol{\omega}}^i - \mathbf{b}_{\boldsymbol{\omega}}^i - {}^{B}\boldsymbol{\omega}^i \\ \mathbf{R}\left(\mathbf{x}_i, {}^{I}\mathbf{a}_m^i, \mathbf{n}_{\mathbf{a}}^i\right) &= \left({}^{I}\mathbf{R}_B^i\right)^{-1}\, {}^{I}\mathbf{a}_m^i - \mathbf{n}_{\mathbf{a}}^i - \mathbf{b}_{\mathbf{a}}^i - {}^{B}\mathbf{a}^i \end{aligned} \tag{13} $$

where ${}^{I}\boldsymbol{\omega}_m^i$ and ${}^{I}\mathbf{a}_m^i$ are the i-th frame of measurements of IMU angular velocity and linear acceleration.

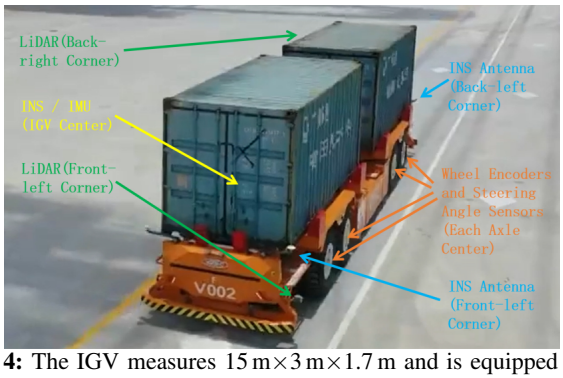


Fig. 4: The IGV measures 15 m×3 m×1.7 m and is equipped with two 16-line LiDARs, four wheel encoders, four steering angle sensors and an INS (featuring two antennas and a 6-axis IMU).

These IMU constraints can be integrated in the following form for the observation update:

$$\mathbf{0} = \mathbf{R}\left(\mathbf{x}_i, {}^{I}\boldsymbol{\omega}_m^i, \mathbf{n}_{\boldsymbol{\omega}}^i\right) + \mathbf{R}\left(\mathbf{x}_i, {}^{I}\mathbf{a}_m^i, \mathbf{n}_{\mathbf{a}}^i\right) \tag{14}$$

F. Wheel Odometry-based Correction with An Adaptive Scaling Model

Wheel odometry provides low-drift measurements, complementing LiDAR and IMU in scenarios with poor visibility, dynamic obstacles, or high-frequency vibrations, and enhances robustness in low-feature environments. In addition, to address challenges like tire slippage on complex terrains, a 3D adaptive scaling factor model dynamically adjusts linear velocity observation weights, compensating errors and improving robustness and precision in dynamic and diverse terrains. Then, upon receiving the i-th frame of wheel odometry measurement data, the constraints for the filter observation update can be expressed as:

$$\begin{aligned}\mathbf{R}\left(\mathbf{x}_i, {}^{B}\mathbf{v}_m^i, \mathbf{0}\right) =& {}^{W}\mathbf{R}_I^i \; {}^{I}\mathbf{R}_B^i \; {}^{B}\mathbf{v}_m^i \odot {}^{v}\mathbf{S}^i \\ & - {}^{W}\mathbf{R}_I^i \left[{}^{I}\mathbf{R}_B^i \, {}^{B}\boldsymbol{\omega}^i\right]_\times {}^{I}\mathbf{p}_B^i \\ & - {}^{W}\mathbf{v}_I^i \\ \mathbf{R}\left(\mathbf{x}_i, {}^{B}\boldsymbol{\omega}_m^i, \mathbf{0}\right) =& {}^{B}\boldsymbol{\omega}_m^i - {}^{B}\boldsymbol{\omega}^i\end{aligned} \tag{15}$$

where ${}^{B}\boldsymbol{\omega}_m^i$ and ${}^{B}\mathbf{v}_m^i$ are the i-th frame of the measured angular velocity and linear velocity from the wheel odometry. The operator $\odot$ denotes the element-wise product.

These constraints for wheel odometry can be integrated in the following form for the observation update:

$$\mathbf{0} = \mathbf{R}\left(\mathbf{x}_i, {}^{B}\mathbf{v}_m^i, \mathbf{0}\right) + \mathbf{R}\left(\mathbf{x}_i, {}^{B}\boldsymbol{\omega}_m^i, \mathbf{0}\right) \tag{16}$$

IV. EXPERIMENTS

To evaluate the performance of the proposed method, we have gathered real-world data from a typical dynamic and open port terminal to build a dataset for testing. Details of the experimental setup are provided in Sec. IV-A, and the localization results are discussed in Sec. IV-B.

A. Experiment Setup

Global Prior Map Construction. Port scenario is large-scale, highly dynamic, and unbounded, with sparse yet repetitive environment features. Frequent movements of mechanical equipment, dense vehicle occlusions, and complex logistics operations result in significant temporal variations

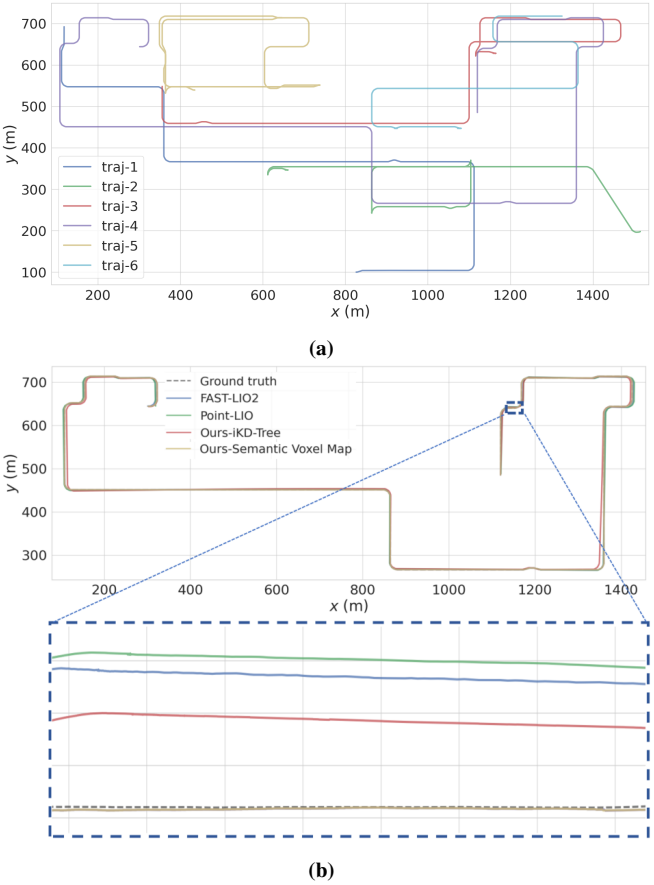


Fig. 5: (a) The trajectories on our private dataset are demonstrated. These 6 trajectories cover multiple distinct motion scenarios, primarily traversing the main passageways of an industrial port. (b) Detailed comparison of the longest trajectory (**traj-4**) from (a), illustrating the performance of FAST-LIO2, Point-LIO, our primary proposed method (Ours-Semantic Voxel Map), and its variant (Ours-iKD-Tree). This comparison emphasizes long-term accuracy and robustness across varying levels of environmental complexity.

in environmental features. Besides, LiDAR data primarily captures ground information, posing challenges for stable map construction and high-precision localization. In global map construction, we focus on extracting static features (e.g., ground elements) and systematically eliminate potential dynamic objects from the map. Additionally, the structured static semantic features (such as pillars, as shown in Fig. 2) present in the environment are incorporated to enhance environmental understanding and ensure accurate and stable global pose constraints. Finally, a stable and reliable global prior map covering approximately 1 million square meters is generated, as illustrated in Fig. 3.

Datasets and Sensor Equipments. As illustrated in Fig. 4, each IGV is equipped with a multi-sensor layout that includes two diagonally mounted 16-line LiDARs (10 Hz) to maximize coverage for environmental perception, a centrally located INS/IMU (20 Hz) for attitude and motion information, two GNSS antennas mounted at the front-left and rear-left corners for high-precision positioning, and four wheel encoders (50 Hz) and four steering angle sensors (50 Hz) located at the center of each axle for vehicle motion estima-

TABLE I: Quantitative comparison for Light-LOAM [21], FAST-LIO2 [11], Point-LIO [17], our primary proposed method (Ours-Semantic Voxel Map), and its variant (Ours-iKD-Tree) across six datasets (corresponding to each trajectory in Fig. 5(a)). The dash " – " indicates that the method ***fails*** on this test trajectory.

Dataset	Mileage (km)	Case	Max absolute pose error (m)	Mean absolute pose error (m)	Max lateral error (m)	Mean lateral error (m)	Max longitudinal error (m)	Mean longitudinal error (m)
traj-1	1.843	Light-LOAM	–	–	–	–	–	–
		FAST-LIO2	4.255	2.379	3.893	1.189	4.097	**0.001**
		Point-LIO	21.283	1.676	2.585	0.829	21.257	0.013
		Ours-iKD-Tree	10.932	5.583	9.721	2.729	10.797	**0.001**
		Ours-Semantic Voxel Map	**0.278**	**0.129**	**0.198**	**0.039**	**0.265**	0.064
traj-2	1.770	Light-LOAM	–	–	–	–	–	–
		FAST-LIO2	3.421	0.638	3.239	0.283	2.647	0.013
		Point-LIO	10.681	0.377	2.266	0.119	10.450	0.094
		Ours-iKD-Tree	0.476	0.156	0.440	0.043	**0.370**	**0.001**
		Ours-Semantic Voxel Map	**0.374**	**0.121**	**0.346**	**0.027**	0.372	0.009
traj-3	1.914	Light-LOAM	**0.621**	0.400	0.319	0.090	**0.598**	**0.001**
		FAST-LIO2	5.053	2.554	4.595	1.194	4.965	1.240
		Point-LIO	6.606	2.029	3.823	0.958	6.394	0.866
		Ours-iKD-Tree	13.486	5.314	12.882	2.779	13.196	2.331
		Ours-Semantic Voxel Map	0.699	**0.132**	**0.302**	**0.041**	0.693	0.032
traj-4	2.950	Light-LOAM	–	–	–	–	–	–
		FAST-LIO2	4.869	2.654	4.346	0.562	4.864	1.439
		Point-LIO	14.312	3.242	5.065	0.741	14.276	1.778
		Ours-iKD-Tree	11.061	3.572	10.979	0.139	10.825	1.868
		Ours-Semantic Voxel Map	**0.669**	**0.164**	**0.309**	**0.044**	**0.669**	**0.092**
traj-5	1.840	Light-LOAM	–	–	–	–	–	–
		FAST-LIO2	1.202	0.378	0.721	0.087	1.133	0.245
		Point-LIO	3.619	0.825	1.231	0.044	3.619	0.399
		Ours-iKD-Tree	1.518	0.613	1.246	0.162	1.487	0.348
		Ours-Semantic Voxel Map	**0.483**	**0.129**	**0.201**	**0.018**	**0.479**	**0.090**
traj-6	1.368	Light-LOAM	–	–	–	–	–	–
		FAST-LIO2	1.044	0.431	1.042	0.098	0.999	0.047
		Point-LIO	18.224	0.535	1.092	0.101	18.214	0.113
		Ours-iKD-Tree	**0.397**	0.118	0.252	0.039	**0.322**	0.058
		Ours-Semantic Voxel Map	0.422	**0.076**	**0.142**	**0.027**	0.420	**0.042**

tion and odometry calculation. The LiDAR point clouds are synchronized, distortion-corrected, and then transformed into the vehicle's central coordinate system. Wheel odometry data (50 Hz) is generated by fusing inputs from wheel encoders, steering angle sensors, and motor feedback signals using a vehicle kinematic model. This data is further refined through corrections from IMU pre-integration measurements and transformed into the vehicle's central coordinate system. High-precision localization results, based on an RTK-aided multi-sensor fusion approach (50 Hz), are used as ground truth for quantitative performance evaluation, ensuring the reliability and accuracy of the system's localization and navigation. We manually construct a comprehensive dataset covering approximately 1 square kilometer, consisting of six test trajectories collected from six IGVs, with a total length of 11.685 kilometers (trajectories are showed in Fig. 5(a), while detailed mileage data is provided in the Table I).

Implementation Platform and Hardware. All our comparative experiments are conducted on a computer equipped with an Intel i7-8750H CPU (2.20 G Hz), 32 GB of RAM, and a GeForce GTX 1050 Ti GPU, running the Ubuntu 20.04 operating system. The experimental code is efficiently written in C++ and developed based on the Robot Operating System (ROS) to ensure modularity and flexibility, facilitating testing and system integration.

Baselines and Metrics. To validate the superiority of our proposed method, we conduct comparative experiments to comprehensively evaluate its advantages in terms of localization reliability. We have select several state-of-the-art and representative baseline methods, including FAST-LIO2 [11], Point-LIO [17], and Light-LOAM [21]. To ensure fairness in comparison, the experimental results of all baseline methods are obtained using the original source code provided by their authors, with only minor adjustments to the input data interface to adapt to our dataset. Additionally, to further demonstrate the advantages of our proposed map representation, we compare the conventional iKD-Tree-based map with our proposed formats: the voxelized map enriched with semantic information, referred to as Ours-iKD-Tree and Ours-Semantic Voxel Map, respectively, as shown in the Table I. Absolute Trajectory Error (ATE) is adopted as the primary metric for evaluating SLAM accuracy, and the evo toolkit is utilized to analyze and compare the localization trajectories of different algorithms.

B. Localization Results and Comparison

The full test trajectories are illustrated in Fig. 5(a). Light-LOAM [21] is found to fail on most port test datasets, displaying significant errors. Therefore, the analysis primarily focuses on the successful algorithms: Point-LIO [17], FAST-LIO2 [11], and our primary proposed method (Ours-Semantic Voxel Map), and its variant (Ours-iKD-Tree).

Trajectory Comparison. Our proposed methods demonstrate close alignment with ground truth trajectories across all 6 test datasets. This performance is exemplified by the longest trajectory traj-4 (over 2.950 km) in Fig. 5(b). In this large-scale scenario, FAST-LIO2 [11] and Point-LIO

[17] accumulate significant drift and exhibit instability over the long distance. Ours-iKD-Tree improves accuracy but lacks long-term stability compared to the semantic voxel-based approach. Ours-Semantic Voxel Map show exceptional performance, with enhanced stability, particularly in dynamic and cluttered environments, demonstrating the critical role of semantic information for robust, long-term localization.

Pose Error Quantitative Comparison. Table I highlights notable differences in localization errors across six algorithms. Light-LOAM exhibits the highest errors and poorest adaptability across various scenarios, underscoring its lack of robustness.

In simple scenarios, FAST-LIO2 achieves a maximum error of 1.044 m and an average of 0.431 m (**traj-6**), while Point-LIO shows a maximum error of 10.681 m and an average of 0.377 m (**traj-2**). In complex environments, FAST-LIO2's maximum error rises to 5.053 m (**traj-3**), and Point-LIO's increases to 14.312 m (**traj-4**), both showing significant robustness degradation.

Ours-iKD-Tree demonstrates significantly better performance compared to Light-LOAM, FAST-LIO2 and Point-LIO. For example, in **traj-2**, its maximum absolute error is 0.476 m, with an average error of 0.156 m, a substantial improvement over FAST-LIO2 (3.421 m), and Point-LIO (10.681 m). However, in certain scenarios, such as **traj-3** and **traj-4**, its maximum errors are relatively high (13.486 m and 11.061 m, respectively), indicating room for improvement in dynamic or long-distance tasks.

Ours-Semantic Voxel Map consistently outperforms Ours-iKD-Tree across all scenarios, demonstrating superior robustness. For instance, in **traj-1**, its maximum error is only 0.278 m, with an average error of 0.129 m, significantly better than Ours-iKD-Tree (10.932 m and 5.583 m). In **traj-4**, its maximum error is 0.669 m, with an average error of 0.164 m, compared to Ours-iKD-Tree (11.061 m and 3.572 m), showing remarkable improvement.

V. CONCLUSION

Building on the iESKF filter, this paper proposes a tightly-coupled semantic LiDAR-inertial-wheel odometry fusion framework. Within this framework, a semantic voxel-based matching algorithm is introduced to effectively distinguish and leverage diverse semantic features, thereby mitigating long-term trajectory drift. Additionally, a 3D adaptive scaling strategy is presented to optimize wheel odometry performance on complex terrains. Extensive experiments demonstrate the superiority of our method over state-of-the-art approaches in dynamic and large-scale environments. Successfully deployed in a one-million-square-meter automated port, the system delivers precise and stable localization for 35 IGVs, demonstrating reliability in real-world applications. Future work will focus on expanding semantic element detection, supporting objects with distinctive geometric shapes, and incorporating visual semantic information to further enrich the semantic content of the map and enhance the algorithm's performance in texture-sparse scenarios.

REFERENCES

[1] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards Fully Autonomous Driving: Systems and Algorithms," in *IEEE Int. Veh. Sym.*, 2011, pp. 163–168.

[2] D. Kumar and N. Muhammad, "A Survey on Localization for Autonomous Vehicles," *IEEE Acce.*, vol. 11, pp. 115 865–115 883, 2023.

[3] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time elastic LiDAR odometry with loop closure," in *IEEE Int. Conf. Robot. Autom.*, 2022, pp. 5580–5586.

[4] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 1029–1036, 2023.

[5] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based Semantic SLAM," in *IEEE Int. Conf. Intell. Rob. Syst.*, 2019, pp. 4530–4537.

[6] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN2: Ultra Light LiDAR-based SLAM using Geometric Approximation applied with KL-Divergence," in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11 619–11 625.

[7] Z. Liu, H. Li, C. Yuan, X. Liu, J. Lin, R. Li, C. Zheng, B. Zhou, W. Liu, and F. Zhang, "Voxel-SLAM: A Complete, Accurate, and Versatile LiDAR-Inertial SLAM System," *arXiv preprint arXiv:2410.08935*, 2024.

[8] G. P. C. Júnior, A. M. C. Rezende, V. R. F. Miranda, R. Fernandes, H. Azpúrua, A. A. Neto, G. Pessin, and G. M. Freitas, "EKF-LOAM: An Adaptive Fusion of LiDAR SLAM With Wheel Odometry and Inertial Data for Confined Spaces with Few Geometric Features," *IEEE Trans. on Auto. Sci. and Eng.*, vol. 19, no. 3, pp. 1458–1471, 2022.

[9] Z. Yuan, F. Lang, T. Xu, and X. Yang, "LIWO: LiDAR-Inertial-Wheel Odometry," in *IEEE Int. Conf. Intell. Rob. Syst.*, 2023, pp. 1481–1488.

[10] W. Xu and F. Zhang, "FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, 2021.

[11] 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, 2022.

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

[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 Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, 2022.

[14] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for Fast and Accurate 3D Point Cloud Registration," in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 11 054–11 059.

[15] Z. Chen, Y. Xu, S. Yuan, and L. Xie, "iG-LIO: An Incremental GICP-Based Tightly-Coupled LiDAR-Inertial Odometry," *IEEE Robot. Autom. Lett.*, vol. 9, no. 2, pp. 1883–1890, 2024.

[16] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robot. Sci. and Sys.*, vol. 2, no. 4, 2009, p. 435.

[17] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-LIO: Robust High-Bandwidth Light Detection and Ranging Inertial Odometry," *Adv. Intell. Sys.*, vol. 5, no. 7, p. 2200459, 2023.

[18] C. Zhang, M. Chen, G. Wang, Y. Lin, K. Li, M. Wu, Z. Li, and Q. Wang, "Liwom-gd: Enhanced lidar–inertial–wheel odometry and mapping by fusion with ground constraint and dynamic points elimination," *IEEE Sensors Journal*, vol. 24, no. 19, pp. 30 287–30 303, 2024.

[19] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized LiDAR Odometry and Mapping on Variable Terrain," in *IEEE Int. Conf. Intell. Rob. Syst.*, 2018, pp. 4758–4765.

[20] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

[21] S. Yi, Y. Lyu, L. Hua, Q. Pan, and C. Zhao, "Light-LOAM: A Lightweight LiDAR Odometry and Mapping Based on Graph-Matching," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3219–3226, 2024.