

Robust Localization for Autonomous Vehicles in Highway Scenes

Daqian Cheng¹, Xuchu Ding², Yujia Wu¹, Xiang Zhang¹, and Lei Wang^{3†}

Abstract—Localization for autonomous vehicles on highways remains under-explored compared to urban roads, and state-of-the-art methods for urban scenes degrade when directly applied to highways. We identify key challenges including environment changes under information homogeneity, heavy occlusion, degraded GNSS signals, and stringent downstream requirements on accuracy and latency. We propose a robust localization system to address highway challenges, which uses a dual-likelihood LiDAR front end that decouples 3D geometric structures and 2D road-texture cues to handle environment changes; a Control-EKF further leverages steering and acceleration commands to reduce lag and improve closed-loop behavior. An automated offline mapping and ground-truth pipeline keep maps fresh at high cadence for optimal localization performance. To catalyze progress, we release a public dataset covering both urban roads and highways while focusing on representative challenging highway clips, totaling 163 km; benchmarking is standardized using product-oriented accuracy metrics and certified ground truth. Compared to Apollo and Autoware, our system performs similarly on urban roads but shows superior robustness on challenging highway scenarios. The system has been validated by more than one million kilometers of road testing.

I. INTRODUCTION

Localization is a critical component in autonomous driving systems. Although many methods have been proposed and deployed, most prior work and public benchmarks focus on urban roads; by contrast, highway scenarios, especially long-haul autonomous trucking, remain under-explored. In our experience, when state-of-the-art urban methods are directly migrated to highways, their robustness and availability degrade substantially.

Why is localization different on highways? Based on more than one million kilometers of road testing, we categorize the challenges into external and internal factors. Externally, rapid environmental changes such as road construction and mowing disrupt map-observation consistency, particularly when environment information is highly homogeneous; side-by-side large trucks induce substantial occlusions; and limited mobile network coverage together with sparse Real-Time Kinematic (RTK) service often degrade Global Navigation Satellite System (GNSS) performance. Internally, the truck operates at speeds up to 75 mph (120 km/h), where tracking accuracy becomes highly sensitive to yaw-rate observation errors and delays. Under such conditions, the closed-loop



Fig. 1: Photo of autonomous truck. Designed for redundancy, the vehicle integrates four long-range 128-line spinning LiDARs (two on top, two on side panels), along with two IMUs and two VCUs installed inside the cabin.

system may exhibit degraded tracking performance or even oscillatory behavior.

Prior localization systems span GNSS & Inertial Navigation System (INS) integration, map-based LiDAR localization using geometry/intensity/semantics, and visual, inertial multi-sensor fusion. These methods excel on urban roads but rely heavily on stable, repeatable landmarks and, as our experiments indicate, are ill-suited for large-scale environmental changes and severe occlusion. Consequently, highway localization still lacks a robustness-oriented system that explicitly addresses these challenges.

To this end, we propose a robust yet simple highway-ready localization system. Environmental information is separated into two modes: 3D geometry (e.g., road structure and terrain shape) and 2D road-surface texture (e.g., lane markings). It is our experience that environmental changes typically affect one mode at a time; therefore, our LiDAR localizer employs two likelihood estimators: one for geometry structure and one for road texture, and fuses them in a Kalman filter, which also provides complementary robustness under heavy occlusion. An automated mapping pipeline keeps maps fresh, updating them daily to ensure map-observation inconsistencies are addressed; rare cases where geometry and texture change simultaneously are considered out of scope without materially harming availability. To improve tracking performance and mitigate oscillatory behavior, we introduce a Control EKF that integrates steering and acceleration commands into a data-driven truck model, reducing localization lag and aligning estimates with real dynamics. Despite these enhancements, the system remains simplistic with modest hardware requirements. Compared to Apollo [1] and Autoware [2], which both perform well in typical conditions, our

*This work was supported by Bot Auto.

¹Daqian Cheng, Yujia Wu, and Xiang Zhang are with Bot Auto daqian.cheng.162, yujia.wu.365, xiang.zhang.854@bot.auto

²Xuchu Ding is with Archer Aviation xding@archer.com

³Lei Wang is with Prexa AI wanglei07@gmail.com

†Corresponding author

system shows superior robustness under challenging highway scenarios.

To promote community progress, we release a public dataset focusing on highway challenges: construction, mowing, and heavy occlusion, alongside typical highway and urban segments for baseline comparison¹. Representative clips are visualized in Fig. 4. The benchmark includes 48 clips covering 163 km. Evaluation is standardized with certified ground-truth trajectories and product-oriented accuracy metrics, detailed in Sec. V-A.

We summarize our contributions as follows:

- **A robust localization system for highway challenges.** The system withstands environmental change and heavy occlusion, and achieves low lag for high-speed control by integrating vehicle control signals. This localization system has been proven through 1M km of road testing.
- **Highway localization dataset & evaluation protocol.** We identify key highway-specific challenges and release representative clips, together with standardized metrics and certified ground truth.

II. RELATED WORK

Modern localization for autonomous driving primarily adopts multi-sensor fusion frameworks that integrate GNSS/INS, LiDAR matching, and vision feature alignment. For LiDAR matching, existing methods can be broadly divided into geometry-based and learning-based approaches. Geometry-based LiDAR localization methods [1], [3]–[5] typically register online scans to prebuilt maps using variants of Iterative Closest Point (ICP) or Normal Distributions Transform (NDT) [6], as well as voxel- or grid-based likelihood estimations. Learning-based approaches, such as [7], [8], directly regress poses from raw or projected LiDAR measurements, demonstrating robustness against geometry degradation and perceptual aliasing. More recently, EgoVM [9] estimates poses via cross-modality matching between BEV perception features and lightweight vector maps, employing a robust histogram solver. For vision-based localization, feature alignment methods leverage image representations. Approaches such as PoseNet [10], hourglass network regressors [11], LSTM-based correlation models [12], and geometry-aware loss formulations [13] have enabled image-based 6 degree of freedom (DoF) localization. Among these, geometry-based LiDAR methods remain the most widely deployed in real-world systems, proving to be more robust. Furthermore, some localization frameworks [14]–[16] incorporate control commands to enhance fusion accuracy.

Generic LiDAR odometry methods—including KISS-ICP [17], CT-ICP [18], LIO-SAM [19], LOAM [20], and FAST-LIO2 [21]—have achieved impressive odometry accuracy on public datasets. However, adapting these methods for production-level localization requires additional effort, as they must address odometry drift that compromises global consistency.

¹Dataset will be made publicly available on Bot Auto website (bot.auto).

Compared to urban settings, research on highway localization is relatively limited. A comprehensive survey [22] highlights existing approaches primarily designed for driver-assistance systems, which fall short of meeting the stringent accuracy and robustness requirements of L4 autonomy. Other works rely on lane markings for localization [23], [24], but such methods are vulnerable to lane repavement and environmental changes.

Public localization benchmarks have significantly advanced the field, but remain biased toward urban environments. KITTI [25] continues to serve as a canonical odometry and localization benchmark, yet it contains limited highway scenarios. Pit30m [26] and Oxford RobotCar [27] provide larger-scale datasets but are still urban-centric. Apollo SouthBay [28] includes one freeway segment, but is located in a metropolitan area that is comparatively feature-rich and does not represent U.S. interstate corridors. In addition, this segment lacks the challenging highway scenarios explored in our work. These gaps motivate the development of our highway-centric dataset, which emphasizes challenging scenarios for robust localization.

III. LOCALIZATION METHOD

Our localization framework estimates the autonomous vehicle’s position, velocity, and attitude (PVA) jointly under the Universal Transverse Mercator (UTM) coordinate system. A standard Error State Kalman Filter (ESKF) fuses the 100 Hz PVA prediction from INS based on IMU readings and the 10 Hz pose (PA) estimation from the LiDAR Localizer using the 3D and intensity information of the LiDAR point cloud. Finally, to make the localization result smooth and low-lag enough for downstream, a Control EKF module takes the acceleration and steering command from Vehicle Control Unit (VCU) as prediction and takes the PVA estimation from ESKF as measurement to produce the final localization result. See Fig. 2 for our overall system design. We do not utilize the GNSS system other than for bootstrapping, because we find that it often fails sporadically due to limited RTK coverage and mobile network services.

We will cover the LiDAR Localization algorithms, the ESKF framework and the Control EKF in III-A, III-B and III-C, respectively.

A. LiDAR Localizer

LiDAR Localizer matches the pattern of onboard LiDAR point cloud with pre-built maps to provide pose estimations. We build the following two maps offline using the GNSS and LiDAR data collected:

- An NDT map which voxelizes the 3D space into 0.8 m cubes; each voxel stores the 3D mean and covariance of the LiDAR points’ positions within the voxel.
- An intensity map which partitions the space into a horizontal 2D grid of 0.125 m cells; each cell stores the mean and variance of the LiDAR intensities within.

Please find the details of our mapping process in IV.

Our LiDAR localization algorithm is a Maximum Likelihood Estimator (MLE). The idea is to sample poses around a

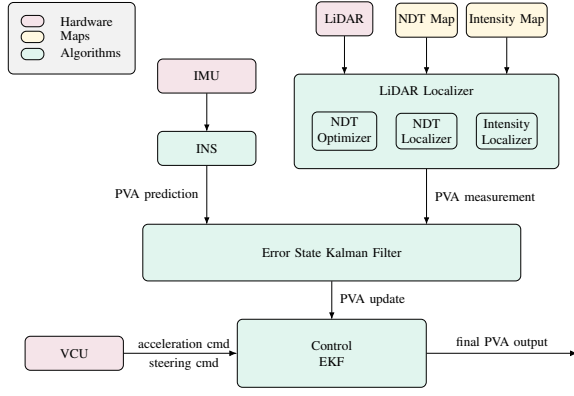


Fig. 2: Localization system overview. LiDAR localizer outputs pose measurement and covariance, ESKF fuses INS prediction and LiDAR measurement, and Control EKF consumes control commands plus ESKF output to produce final PVA estimation.

given prior provided by ESKF prediction, then compute the likelihood of each sampled pose based on the geometry information from the NDT map and the texture information from the intensity map, and output the pose with the maximum likelihood. To speed-up runtime, sample pose evaluations are performed on high-performing Graphics Processing Units (GPU) in parallel.

Algo. 1 summarizes our LiDAR localization algorithm. For notation, we use \mathbf{p} , \mathbf{R} to denote the UTM position and rotation matrix $\text{SO}(3)$. We use Z to denote the point cloud obtained onboard aggregated from all LiDARs at the localization time. Each point has an intensity value and a 3D position \mathbf{q} in the body frame.

Algorithm 1 LiDAR Localizer

Inputs: Prior pose $(\hat{\mathbf{p}}, \hat{\mathbf{R}})$; NDT map M_{NDT} ; Intensity map M_{Int} ; onboard LiDAR point cloud Z

```

 $\mathbf{p}^*, \mathbf{R}^* = \text{InitialOptimization}(\hat{\mathbf{p}}, \hat{\mathbf{R}}, M_{NDT}, Z)$ 
 $\{(\mathbf{p}^1, \mathbf{R}^1), \dots, (\mathbf{p}^N, \mathbf{R}^N)\} = \text{PoseSample}(\mathbf{p}^*, \mathbf{R}^*)$ 
 $Z_{NDT} = \text{SampleForNDT}(Z)$ 
 $Z_{Int} = \text{SampleForIntensity}(Z)$ 
for  $i = 1, 2, \dots, N$  do
   $L^i \leftarrow 0$ 
   $L^i \leftarrow L^i + \text{NDTLogLikelihood}(\mathbf{p}^i, \mathbf{R}^i, M_{NDT}, Z_{NDT})$ 
   $L^i \leftarrow L^i + \text{IntensityLogLikelihood}(\mathbf{p}^i, \mathbf{R}^i, M_{Int}, Z_{Int})$ 
   $\text{Prob}^i \propto e^{-L^i}$ 
end for
 $\mathbf{p}, \mathbf{R}, \mathbf{W} = \text{MeanAndCovariance}(\{\mathbf{p}^i\}, \{\mathbf{R}^i\}, \{\text{Prob}^i\})$ 
return  $\mathbf{p}, \mathbf{R}, \mathbf{W}$ 

```

1) *Initial Optimization:* Although the pose-wise likelihood calculations are executed on GPU, sampling poses on all six DoF is computationally intractable. Since the z , pitch, and roll can be readily and accurately estimated by LiDAR-map alignment because of the presence of the ground plane, our strategy is to decouple the 6 DoF: first use a lightweight optimizer to estimate z , pitch, and roll, then use MLE for the x , y , and yaw dimensions.

In Algo. 1, InitialOptimization is the algorithm to estimate z , pitch and roll. It is an NDT method that iteratively improves pose estimation. We sample M LiDAR points $Z_{NDT} = \{\mathbf{q}^1, \dots, \mathbf{q}^M\}$ from the onboard point cloud Z . In each iteration k , we transform the sampled points \mathbf{q}^j into the UTM frame denoted by $\mathbf{u}^j = \mathbf{R}_{k-1}\mathbf{q}^j + \mathbf{p}_{k-1}$ using the pose $(\mathbf{p}_{k-1}, \mathbf{R}_{k-1})$ from the previous iteration. We look up the voxel that \mathbf{u}^j falls in and get the corresponding 3D mean $\boldsymbol{\mu}^j$ and covariance $\boldsymbol{\Sigma}^j$ stored in the voxel in the NDT map M_{NDT} . Levenberg-Marquardt algorithm is used to iteratively optimize the pose estimation $\mathbf{p}_k, \mathbf{R}_k$ z , pitch, roll components to maximize the log likelihood:

$$-\sum_j (\mathbf{R}\mathbf{q}^j + \mathbf{p} - \boldsymbol{\mu}^j)^T (\boldsymbol{\Sigma}^j)^{-1} (\mathbf{R}\mathbf{q}^j + \mathbf{p} - \boldsymbol{\mu}^j) \quad (1)$$

2) *Pose and Point Cloud Sampling:* After the initial pose optimization step, we sample N candidate poses $(\mathbf{p}^1, \mathbf{R}^1), \dots, (\mathbf{p}^N, \mathbf{R}^N)$ around the optimized pose $\mathbf{p}^*, \mathbf{R}^*$ to be used for MLE. In our implementation we choose N to be 8000. We fix the z , pitch, and roll dimensions and sample the x, y , yaw dimensions uniformly with sampling range $0.5\text{ m}, 1.0\text{ m}, 0.5^\circ$ respectively, which achieves a good balance between convergence radius and sample efficiency.

To compute the NDT likelihood and intensity likelihood, we need to downsample the raw point cloud Z to reduce computation workload. Two different sampling strategies are used because the NDT evaluation captures the environment's geometry pattern which has more macroscale consideration compared to intensity evaluation which captures more microscale texture information. Therefore, for the NDT likelihood calculation, we first use a $(1\text{ m})^3$ sized 3D voxel-grid sampler to ensure a sparse and uniform sampling and then randomly down-select 5000 points to ensure computation efficiency. For the intensity likelihood calculation, we employ a $(0.1\text{ m})^2$ sized 2D grid sampler followed by a 30000 point random down-selection.

3) *NDT and Intensity Likelihood:* To calculate the NDT likelihood, we use the 5000 point samples Z_{NDT} . Let \mathbf{p}, \mathbf{R} denote the pose being evaluated. We transform each point sample $\mathbf{q}^j \in Z_{NDT}$ to the UTM frame as $\mathbf{u}^j = \mathbf{R}\mathbf{q}^j + \mathbf{p}$. We look up the voxel that \mathbf{u}^j falls in and get the corresponding 3D mean $\boldsymbol{\mu}^j$ and covariance $\boldsymbol{\Sigma}^j$ stored in the voxel in the NDT map M_{NDT} . We calculate the log likelihood as the pose \mathbf{p}, \mathbf{R} as

$$-\sum_j (\mathbf{u}^j - \boldsymbol{\mu}^j)^T (\boldsymbol{\Sigma}^j)^{-1} (\mathbf{u}^j - \boldsymbol{\mu}^j) \quad (2)$$

The intensity likelihood computation is similar. We just replace the 3D voxel matching by the 2D grid matching and use the intensity mean and variance in each matched grid for likelihood computation.

At last, we compute a distribution $\{\text{Prob}^i\}$ where $\text{Prob}^i \propto e^{-L^i}$ for the sampled poses, and return the mean \mathbf{p}, \mathbf{R} and covariance \mathbf{W} (under Lie algebra) of those poses as the final output of the LiDAR Localizer.

B. ESKF Framework

The localization state is denoted by $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \boldsymbol{\theta})$ where \mathbf{p}, \mathbf{v} are the UTM position and linear velocity; $\boldsymbol{\theta}$ is the rotation vector of the attitude. We also use the rotation matrix representation \mathbf{R} interchangeably with $\boldsymbol{\theta}$ to represent the attitude.

The error state is denoted by $\delta\mathbf{x} = (\delta\mathbf{p}, \delta\mathbf{v}, \delta\boldsymbol{\theta})$ and it is defined as $\delta\mathbf{x} := \mathbf{x} - \hat{\mathbf{x}}$, where \mathbf{x} and $\hat{\mathbf{x}}$ are the true and nominal states respectively.

In ESKF, we need to predict the nominal state and the error state; and we need measurement to update the error state and correct the nominal state by the error state.

1) *INS*: INS uses the following system dynamic equations for nominal state prediction at time k :

$$\begin{aligned}\hat{\mathbf{p}}_k &= \mathbf{p}_{k-1} + \mathbf{v}_{k-1}\Delta t \\ \hat{\mathbf{v}}_k &= \mathbf{v}_{k-1} + (\mathbf{R}_{k-1}\tilde{\mathbf{a}} + \mathbf{g})\Delta t \\ \hat{\mathbf{R}}_k &= \mathbf{R}_{k-1} \text{Exp}(\tilde{\boldsymbol{\omega}}\Delta t)\end{aligned}\quad (3)$$

Here Δt is the time difference between the time step k and $k-1$; and $\tilde{\mathbf{a}}, \tilde{\boldsymbol{\omega}}$ are the acceleration and angular velocity readings of the IMU in the body frame. For simplicity, we assume that the IMU biases have already been corrected.

The dynamic of the error state can easily be derived from the above system dynamics:

$$\begin{aligned}\delta\hat{\mathbf{p}}_k &= \delta\mathbf{p}_{k-1} + \delta\mathbf{v}_{k-1}\Delta t \\ \delta\hat{\mathbf{v}}_k &= \delta\mathbf{v}_{k-1} - [(\mathbf{R}_{k-1}\tilde{\mathbf{a}}) \times \delta\boldsymbol{\theta}_{k-1}]\Delta t \\ \delta\hat{\boldsymbol{\theta}}_k &= \text{Exp}(-\tilde{\boldsymbol{\omega}}\Delta t)\delta\boldsymbol{\theta}_{k-1}\end{aligned}\quad (4)$$

We apply standard error state covariance matrix prediction based on 4; details are omitted for brevity.

2) *ESKF Update*: LiDAR Localizer takes the ESKF's prediction as input and returns the its pose estimation and covariance matrix as the measurement update in ESKF. Therefore, the measurement model is $\mathbf{z} = h(\mathbf{x}) = (\mathbf{p}, \boldsymbol{\theta})$. Note that the Jacobian of h in ESKF is defined as

$$\mathbf{H} = \left. \frac{\partial h(\hat{\mathbf{x}} + \delta\mathbf{x})}{\partial \delta\mathbf{x}} \right|_{\delta\mathbf{x}=0} \quad (5)$$

where $\hat{\mathbf{x}}$ is the nominal state (predicted state) and $\delta\mathbf{x}$ is the error state.

Using Lie algebra (see [29]), we have

$$\left. \frac{\partial \text{Log}(\text{Exp}(\hat{\boldsymbol{\theta}})\text{Exp}(\delta\boldsymbol{\theta}))}{\partial \delta\boldsymbol{\theta}} \right|_{\delta\boldsymbol{\theta}=0} = \mathbf{J}_r^{-1}(\hat{\boldsymbol{\theta}}) \quad (6)$$

Therefore, in our case,

$$\mathbf{H}(\hat{\mathbf{x}}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{J}_r^{-1}(\hat{\boldsymbol{\theta}}) \end{bmatrix} \quad (7)$$

where \mathbf{I}_3 and $\mathbf{0}_3$ are the 3×3 identity and zero matrices. We apply the standard ESKF update using (7).

C. Control EKF

Control EKF is used to smooth the ESKF output to meet the smoothness requirement posted by the downstream control module. Unlike a conventional INS-based EKF that is driven directly by sensor outputs, the Control EKF uses steering and acceleration commands as inputs. This accounts for the intrinsic lags introduced by combustion and chassis dynamics, enabling the prediction step to better capture vehicle behavior and improve estimation accuracy.

A physics based truck model is inherently complicated, even under ideal assumptions [30], [31]. We use *SINDy* [32] to capture a set of simple governing equation for the normal driving data. The resulting dynamics can be expressed in the discrete-time state-space form,

$$\begin{aligned}x_{k+1} &= x_k - v_k \sin(\phi_k)\Delta t, \\ y_{k+1} &= y_k + v_k \cos(\phi_k)\Delta t, \\ v_{k+1} &= v_k + a_k\Delta t, \\ \phi_{k+1} &= \phi_k + r_k\Delta t, \\ r_{k+1} &= r_k + [(w_1/v_k + w_2v_k)r_k + w_3\sigma_k^{\text{cmd}}]\Delta t, \\ a_{k+1} &= a_k + w_4(-a_k + a_k^{\text{cmd}})\Delta t,\end{aligned}\quad (8)$$

where $(x_k, y_k) \in \mathbb{R}^2$ denote the UTM position, $v_k \in \mathbb{R}$ is the vehicle speed, $a_k \in \mathbb{R}$ the longitudinal acceleration, $\phi_k \in \mathbb{R}$ the yaw angle in the UTM frame, and $r_k \in \mathbb{R}$ the yaw rate. The system inputs are the steering angle $\sigma_k^{\text{cmd}} \in \mathbb{R}$ and the commanded acceleration $a_k^{\text{cmd}} \in \mathbb{R}$. We assume the process disturbances are applied to control inputs. Based on this state-space model (8), the design of the Extended Kalman Filter follows directly from the standard textbook formulation.

IV. GROUND TRUTH GENERATION AND MAPPING

In this section, we describe the offline processes to generate localization ground-truth (GT) trajectory for each of our trips without relying on high-cost GT-purpose systems. Our GT framework is based on Simultaneous Localization and Mapping (SLAM). It serves two purposes. One is to generate localization ground-truth for general trips which enables our fully automated pipeline to validate localization in these trips. The other is to generate mapping ground truth for candidates trips for high-definition (HD) mapping. Reliable mapping infrastructure is a critical prerequisite for robust highway-grade autonomous localization. In this section we also summarize the design strategies and implementation details of our mapping pipeline.

A key feature of ground-truth and mapping framework is that it is fully automated and low cost, which enables us to build HD maps in high cadence and be highly scalable as we scale up our fleet. Another is that we do not fully depend on GNSS systems for our ground-truth framework (i.e., GNSS data are used only when available). This is useful in the case of GNSS outages, which can occur due to bad weather, structural occlusions, signal interference, weak coverage, and a multitude of other factors that can arise in practice as we accumulate many miles of driving data.

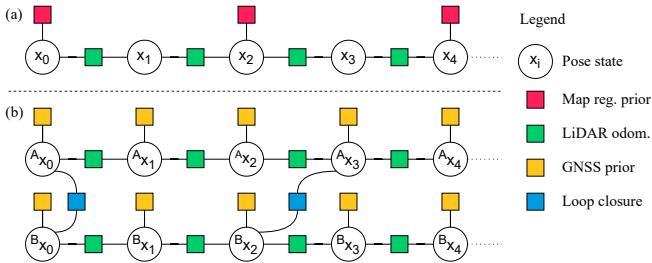


Fig. 3: Factor-graphs of the proposed ground-truth framework. (a) shows localization GT formulation, where map prior factors are obtained using point cloud registration with the trip’s corresponding mapping trips. (b) shows mapping GT formulation, where loop closures are generated between two or more overlapping mapping sessions.

We use one consistent framework to generate both types ground-truth mentioned above, with the only difference being how the SLAM system is set up. We use a well-known factor graph framework to organize constraints. A factor graph provides a flexible representation for fusing heterogeneous sensor modalities and constraints into a unified optimization problem, and have been widely used in state-of-the-art SLAM backends [33]. In this framework, robot poses are modeled as nodes in the graph, while constraints are represented as factors connecting these nodes. The Maximum A Posteriori (MAP) estimate of the trajectory is obtained by solving a nonlinear least-squares problem over this graph structure. Ceres Solver [34] is used to model and solve the MAP optimization problem.

Fig. 3 illustrates the factor graph design for localization and mapping GT generation. For localization GT, we treat the map as reference, as the goal of localization is to produce a trajectory estimate under which the registered onboard point cloud aligns well with the mapping point cloud. In our implementation, we form prior factors using G-ICP [35] point cloud registration between onboard point cloud frames and a dense offline LiDAR point cloud map derived from the mapping process. We employ KISS-ICP LiDAR odometry [17] to generate odometry factors between each two consecutive frames to ensure local smoothness. The localization GT has superior accuracy to onboard LiDAR localization by leveraging future information and a full-density point cloud map.

For mapping, the goal is to generate trajectories that produce complete, consistent, and sharp point cloud maps. Therefore, to produce one map, we use more than one trips to alleviate occlusion-induced map incompleteness of any single pass. All overlapping mapping sessions must be aligned to produce one consistent point cloud map; therefore, we form loop closure factors among co-visible mapping sessions via G-ICP point cloud registration. We employ the same LiDAR odometry factor to ensure local point cloud consistency. Global accuracy is ensured by GNSS prior factors (if they are available).

V. EXPERIMENTS

A. Dataset with Diverse Highway Challenges

We identify three major external challenges for localization on highway: road construction, terrain shift, and occlusion. Highway environments can be highly homogeneous; on some stretches, lane markings are the only texture cue and the terrain elevation is the only 3D geometry structure. Therefore, environment change like roadwork and terrain shift can induce rapid and extensive map-observation inconsistencies. Specifically, road repavement may shift lane markings, misleading lane marking-based localization methods; road-side grass mowing can drastically reduce terrain elevation by 1 meter. Heavy occlusion by side-by-side trucks can block LiDAR’s field of view, causing degraded localization accuracy and sometimes degeneration. Some clips are presented in detail in Sec. V-B.

We release a highway-centric dataset collected on autonomous trucks in Fig. 1. The dataset consists of 48 mapping-testing clip pairs, including 9 urban and 39 highway segments. Instead of providing lengthy but unremarkable recordings, 23 of the highway clips are deliberately selected to capture challenging scenarios, comprising 9 construction zones, 7 terrain shifts, and 7 occlusions. Fig. 4 visualizes some challenging clips. The remaining 16 clips are categorized as typical highway, representing general highway conditions. The dataset totals 163 km and collected on Interstate highway corridors in Texas, U.S., which are some of the most freight-concentrated corridors. The dataset has diverse weather conditions, seasons, and time of day.

The evaluation is standardized with product-oriented metrics. In contrast to academic convention of evaluating statistical accuracy, we prioritize robustness and deem performance as pass-or-fail. Specifically, standard highway lane width and truck width dictates a lateral accuracy of 20 cm; empirically, downstream object detection module requires a yaw accuracy of 0.3° and a longitudinal accuracy of 50 cm in order to correctly associate detection with HD-map, e.g., a larger yaw error may result in incorrect vehicle-lane associations.

B. Localization Benchmark

We compare the proposed localization method, including NDT-only and intensity-only variants, to two public and open-source baselines, Apollo [1] and Autoware [2]. Although more recent methods exist, these two frameworks have been widely deployed in real-world and exhibit adequate accuracy and strong robustness. Both baselines share similar LiDAR localization and EKF framework design as ours. For LiDAR localization, Autoware uses an NDT variant; Apollo mainly utilizes the LiDAR intensity information via a 2-D intensity map, but also weakly incorporates geometric information using a 2-D elevation map. In contrast, our method takes advantage of geometry information more heavily with a 3D voxelized NDT map. For a fair comparison, we utilize Apollo’s built-in option to only use GNSS for bootstrapping, which is consistent with Autoware and our methods. Our method achieves 46 ms mean runtime on AMD EPYC 7643 CPU and RTX 4090 GPU.

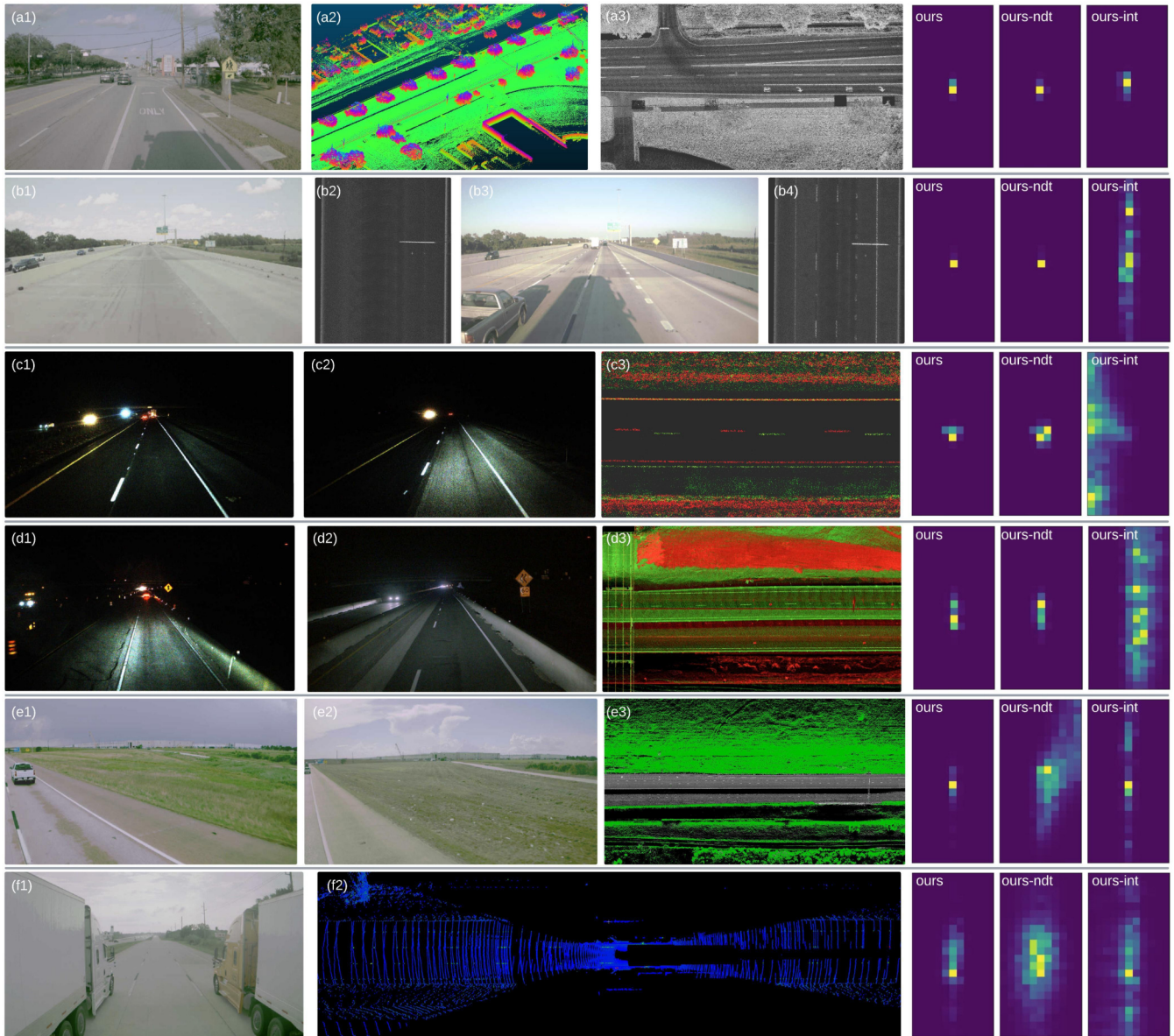


Fig. 4: Case study on urban road and challenging scenarios. All cases show a sample frame's LiDAR localizer distribution marginalized to the lateral ($\pm 0.5\text{m}$) and longitudinal ($\pm 1.0\text{m}$) dimensions, including NDT-only and intensity-only variants. (a) shows an urban road segment; LiDAR point clouds colored by height (a2) and by intensity (a3, in bird's-eye view) show abundant geometric and road texture information respectively. (b) A repavement from Botts' Dots (b1, b2) to painted markings (b3, b4) makes intensity localizer highly uncertain, and fusion relies on NDT localizer. (c) A repavement with shifted painted markings in both lateral and longitudinal direction (c3, from red to green) results in a biased and uncertain intensity localization, but NDT remains accurate and confident due to no geometric change. (d) A repavement with road deviation; the road surface was widened with large structure change (d3, from red to green), resulting in insufficient intensity information match; but NDT remains reliable thanks to terrain and unchanged road structures (e.g., the overhead bridge). (e) Roadside mowing shifts terrain elevation and affects NDT because the grass terrain can occupy most of LiDAR scan, as shown in the LiDAR semantic segmentation visualization (e3), while intensity localization helps keep fusion result accurate. (f) When ego is "sandwiched" by two other trucks, LiDAR's view becomes heavily occluded (f2, in bird's-eye view), causing uncertain NDT localization; but intensity localization helps maintain accurate and certain lateral localization, which is safety-critical in this scenario.

Both baselines are tuned for optimal performance adaptation to our platform. In particular, we find Autoware less production-ready for highway operation for the following reasons: (1) Autoware uses dense LiDAR point clouds as map, which demands impractically large onboard storage space for our 400 km+ long routes; (2) as the vehicle travels fast, dynamically loading large dense map chunks becomes a bottleneck; (3) native FP32 map point cloud precision does not accommodate long routes and subsequently degrades localization accuracy. For the experiments, we bypassed these limitations by slow-playing the dataset and pre-shifting the map origin to eliminate the floating-point precision issues.

All five methods pass the urban road dataset as shown in Table I, showing that both Apollo and Autoware generalize well to our truck platform. The performance of ours-ndt and ours-intensity demonstrates that the surface roads contain abundant geometric and texture information, and accurate localization can be achieved by utilizing either information. In Table II, we show the results for the typical highway dataset. It can be seen that methods that utilize 3D geometric information perform well, namely ours, ours-ndt, and autoware; whereas methods that rely on texture, i.e., Apollo and ours-intensity, exhibit large longitudinal errors in some cases. The distribution visualizations (see Fig. 4) suggest intensity-based LiDAR localizations have large longitudinal uncertainty, showing that the texture information in highway environments is less reliable than geometric information in these cases, especially to provide longitudinal localization constraints.

TABLE I: Accuracy evaluation on urban road dataset

Method	lat/m RMSE	lat/m Max	lon/m RMSE	lon/m Max	yaw/ ^o RMSE	yaw/ ^o Max
Ours	0.019	0.071	0.032	0.137	0.050	0.282
Ours-ndt	0.019	0.083	0.029	0.137	0.051	0.280
Ours-int	0.019	0.073	0.028	0.169	0.050	0.285
Apollo	0.034	0.154	0.071	0.254	0.079	0.289
Autoware	0.016	0.076	0.020	0.087	0.064	0.284

TABLE II: Accuracy evaluation on typical highway dataset

Method	Clip Fail	Frame Fail	lat/m RMSE	lat/m Max	lon/m RMSE	lon/m Max	yaw/ ^o RMSE	yaw/ ^o Max
Ours	0	0%	0.022	0.179	0.031	0.135	0.038	0.259
Ours-ndt	0	0%	0.025	0.169	0.027	0.144	0.037	0.262
Ours-int	4/15	0.14%	0.022	0.184	0.067	0.653	0.046	0.276
Apollo	3/15	1.70%	0.067	0.297	0.108	0.948	0.118	1.548
Autoware	0	0%	0.022	0.179	0.031	0.246	0.050	0.273

For road construction, we find repavements to have the greatest impact on localization by mostly affecting intensity LiDAR localizer. Experiment shows geometric-reliant methods remain robust against repavements. Different types of repavements have different effects on intensity-reliant methods. For Botts’ Dots, intensity localizers show high uncertainty and subject to large error due to the sparsity and LiDAR blooming. Repavements with shifted lane markings can mislead localization. Some road work would guide the

vehicle to deviate from the original road, such as road widening projects, which would cause insufficient information and render intensity localization ineffective, often leading to loss of localization (see Fig 4-b,c,d).

Rapid terrain elevation shifts are usually caused by roadside mowing. Although geometric LiDAR localizers are often robust to this scenario, we find them occasionally uncertain and error-prone, and could lead to loss of localization if not handled. Since road texture is unaffected, intensity localizer can help stabilize LiDAR localization, as shown in Fig. 4-e. In practice, we also mitigate the problem by maintaining fresh maps and by segmenting out affected terrain map.

Side-by-side large trucks cause LiDAR occlusion and reduces information for localization. When occluded on one side, geometric LiDAR localization remains unaffected. However, when occluded on both sides, i.e., ego is “sandwiched” by two other trucks, we find the geometric and texture information to be complementary: the geometric localizer becomes uncertain, but the intensity localizer provides lateral constraint, as shown in Fig. 4-f. Lateral accuracy and certainty is safety critical when ego is driving between two other trucks. Another type of occlusion involves both mapping and onboard: the mapping dataset was occluded on one side, while onboard localization is occluded on the other side; this scenario exhibits similar characteristics to the “sandwich” occlusion cases. The results for our method and Apollo / Autoware in the above three types of challenging environments are summarized in Table III.

TABLE III: Pass/Fail comparisons on the challenging dataset

Method	Repavement (9)	Terrain Shift (7)	Occlusion (7)
Ours	0	0	0
Apollo	4	2	5
Autoware	1	2	1

C. Evaluation of the Control EKF Module

From (8), it is evident that the lateral feedback loop is highly sensitive to yaw-rate estimation noise. A second-order low-pass filter is typically used to reduce noise, with a cost of introducing lag in estimation. Field tests show that when yaw rate delay exceeds 100 ms at 120 km/h, the closed-loop system has little stability margin and can oscillate with small disturbances. We evaluate 3 yaw-rate signals: the direct output from ESKF, the Control EKF, and a second-order low-pass filter. In the ESKF, the INS runs at 100 Hz and unavoidably propagates IMU noise, producing a noisy yaw-rate output. The Control EKF, in contrast, embeds vehicle dynamics into its prediction step, which suppresses noise and reduces latency. Fig. 5 illustrates the difference: the ESKF output is dominated by high-frequency noise, while the Control EKF yields a smoother estimate and reduces effective delay by ~ 100 ms than the low-pass filter. This improvement directly benefits closed-loop tracking.

VI. CONCLUSIONS

In this work, we highlight the unique challenges of localization on highways compared to urban roads. Through

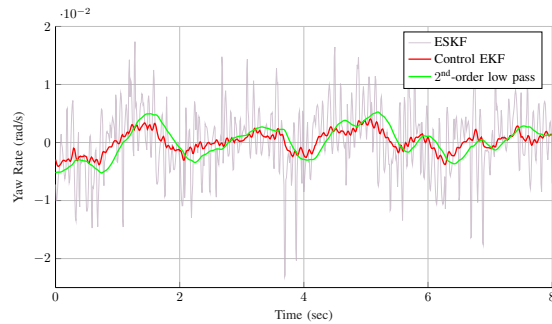


Fig. 5: Yaw rate comparison on highway. The raw ESKF output is too noisy for downstream control module, and using a low pass filter introduces lag. The proposed control EKF achieves smoother yaw rate estimation without lag.

extensive comparisons with baseline methods in challenging highway scenarios, we demonstrate that leveraging geometric structural information is essential for reliable highway localization, while incorporating road texture cues offers complementary robustness under particularly difficult conditions. Moreover, integrating control commands into the localization framework enhances trajectory smoothness without introducing additional system lag. Finally, our public dataset addresses a critical gap in existing localization benchmarks, and we believe it will enable more comprehensive evaluation and foster future research on robust localization.

REFERENCES

- [1] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE international conference on robotics and automation (ICRA)*, 2018.
- [2] Autoware Foundation, "Autoware (ros 2 open-source autonomous driving stack)," <https://github.com/autowarefoundation/autoware>, 2025, version used: main; accessed 2025-09-01.
- [3] R. W. Wolcott and R. M. Eustice, "Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017.
- [4] L. Pan, K. Ji, and J. Zhao, "Tightly-coupled multi-sensor fusion for localization with lidar feature maps," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5215–5221.
- [5] S. Li, S. Wang, Y. Zhou, Z. Shen, and X. Li, "Tightly coupled integration of gnss, ins, and lidar for vehicle navigation in urban environments," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 721–24 735, 2022.
- [6] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2003, pp. 2743–2748.
- [7] Z. Li, G. Li, C. Wang, Q. Xu, and R. Xiong, "L3-Net: Towards Learning based LiDAR Localization for Autonomous Driving," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [8] J. Michel, B. Steder, S. Kohlbrecher, and W. Burgard, "Global Localization with a LiDAR Intensity Map," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [9] Y. He, S. Liang, X. Rui, C. Cai, and G. Wan, "Egovm: Achieving precise ego-localization using lightweight vectorized maps," *arXiv preprint arXiv:2307.08991*, 2023.
- [10] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
- [11] M. Meyer, M. Humenberger, P. Gargallo, A. Smolic, and M. Pollefeys, "Image-Based Localization using Hourglass Networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [12] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-Based Localization Using LSTMs for Structured Feature Correlation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [13] A. Kendall and R. Cipolla, "Geometric Loss Functions for Camera Pose Regression with Deep Learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [14] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 329–343, 2011.
- [15] F. Ma, J. Shi, Y. Yang, J. Li, and K. Dai, "Ack-msckf: Tightly-coupled ackermann multi-state constraint kalman filter for autonomous vehicle localization," *Sensors*, vol. 19, no. 21, p. 4816, 2019.
- [16] H. Min, X. Wu, C. Cheng, and X. Zhao, "Kinematic and dynamic vehicle model-assisted global positioning method for autonomous vehicles with low-cost gps/camera/in-vehicle sensors," *Sensors*, vol. 19, no. 24, p. 5430, 2019.
- [17] I. Vizzo, P. Stotko, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp—simple, accurate, and robust registration if done the right way," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4387–4393.
- [18] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time elastic lidar odometry with loop closure," *arXiv:2109.12979*, 2022.
- [19] T. Shan, B. Englot, *et al.*, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IROS*, 2020.
- [20] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems (RSS)*, 2014.
- [21] 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.
- [22] J. Laconte, A. Kasmi, R. Aufrère, M. Vaidis, and R. Chapuis, "A survey of localization methods for autonomous vehicles in highway scenarios," *Sensors*, vol. 22, no. 1, p. 247, 2021.
- [23] M. Harr, K.-D. Mueller, A.-M. Hellmund, and N. Wagner, "Robust localization on highways using low-cost gnss, front/rear mono camera and digital maps," in *AmE 2018-Automotive meets Electronics; 9th GMM-Symposium*. VDE, 2018, pp. 1–7.
- [24] M. J. Choi, J. K. Suhr, K. Choi, and H. G. Jung, "Low-cost precise vehicle localization using lane endpoints and road signs for highway situations," *IEEE Access*, vol. 7, pp. 149 846–149 856, 2019.
- [25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012, pp. 3354–3361.
- [26] J. Martinez, S. Doubov, J. Fan, S. Wang, G. Mátyus, R. Urtasun, *et al.*, "Pit30m: A benchmark for global localization in the age of self-driving cars," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4477–4484.
- [27] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
- [28] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-net: Towards learning based lidar localization for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6389–6398.
- [29] T. D. Barfoot, *State Estimation for Robotics*. Cambridge: Cambridge University Press, 2017.
- [30] C. Chen and M. Tomizuka, "Lateral control of commercial heavy vehicles," *Vehicle System Dynamics*, vol. 33, no. 6, pp. 391–420, 2000.
- [31] —, "Vehicle lateral control on automated highways: a backstepping approach," *ASME International Mechanical Engineering Congress and Exposition*, vol. 18244, pp. 693–700, 1997.
- [32] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [33] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.
- [34] S. Agarwal, K. Mierle, and Others, "Ceres solver," <http://ceres-solver.org>, 2022.
- [35] D. Akita, K. Koide, and S. Oishi, "Small-gicp: An efficient and accurate point cloud registration algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1151–1157.