


Article

Ego-Lane Index Estimation Based on Lane-Level Map and LiDAR Road Boundary Detection

Baoguo Yu ¹, Hongjuan Zhang ^{2,3,*}, Wenzhuo Li ², Chuang Qian ⁴, Bijun Li ^{2,3}  and Chaozhong Wu ⁴

¹ The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China; yubg@sina.cn

² State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China; alvinlee@whu.edu.cn (W.L.); lee@whu.edu.cn (B.L.)

³ Engineering Research Center for Spatio-Temporal Data Smart Acquisition and Application, Ministry of Education of China, Wuhan University, Wuhan 430072, China

⁴ Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China; qian_c@whut.edu.cn (C.Q.); wucz@whut.edu.cn (C.W.)

* Correspondence: hongjuanzhang@whu.edu.cn

Abstract: Correct ego-lane index estimation is essential for lane change and decision making for intelligent vehicles, especially in global navigation satellite system (GNSS)-challenged environments. To achieve this, we propose an ego-lane index estimation approach in an urban scenario based on particle filter (PF). The particles are initialized and propagated by dead reckoning with inertial measurement unit (IMU) and odometry. A lane-level map is used to navigate the particles taking advantage of topologic and geometric information of lanes. GNSS single-point positioning (SPP) can provide position estimation with meter-level accuracy in urban environments, which can limit drift introduced by dead reckoning for updating the weight of each particle. Light detection and ranging (LiDAR) is a common sensor in an intelligent vehicle. A LiDAR-based road boundary detection method provides distance measurements from the vehicle to the left/right road boundaries, which provides a measurement for importance weighting. However, the high precision of the LiDAR measurements may put a tight constraint on the distribution of particles, which can lead to particle degeneration with sparse particle sets. To deal with this problem, we propose a novel step that shifts particles laterally based on LiDAR measurements instead of importance weighting in the traditional PF scheme. We tested our methods on an urban expressway at a low traffic volume period to ensure road boundaries can be detected by LiDAR measurements at most time steps. Experimental results prove that our improved PF scheme can correctly estimate ego-lane index at all time steps, while the traditional PF scheme produces wrong estimations at some time steps.

Keywords: ego-lane index estimation; lane-level map; particle filter; road boundary detection; LiDAR; GNSS



Citation: Yu, B.; Zhang, H.; Li, W.; Qian, C.; Li, B.; Wu, C. Ego-Lane Index Estimation Based on Lane-Level Map and LiDAR Road Boundary Detection. *Sensors* **2021**, *21*, 7118. <https://doi.org/10.3390/s21217118>

Academic Editor: Chris Rizos

Received: 17 July 2021

Accepted: 15 September 2021

Published: 27 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Ego-lane index estimation refers to the determination of the lane index currently occupied by the vehicle, which is essential for intelligent vehicles, as it can recommend lane changes if needed. Visual multi-lane-marking detection is fully investigated in recent years. The driving lane and its adjacent lanes can be detected visually to infer which lane the vehicle locates. Zhao et al. [1] proposed a Catmull–Rom spline-based multi-lane detection and tracking system that could run in real time and detect both straight and curved shapes of lanes. Kang et al. [2] investigated multi-lane detection in highway scenarios based on geometric lane estimation by assuming that all lanes were parallel to each other and have the same width. Hur et al. [3] tried to dispose of this assumption and detect multiple lanes in urban driving environments using conditional random fields. They successfully detected non-parallel lanes found at intersections, in splitting lanes, and in merging lanes.

However, since visual features vary with the illumination, weather condition and distance of the region, all above feature-based algorithm is restricted to the illuminant variation. To improve the robustness for extracting the multi-lane marking, some researchers proposed multi-lane detection algorithms based on neural networks. Chao et al. [4] detected multi-lanes based on a deep convolutional neural network. Huval et al. [5] used a convolutional neural network (CNN) architecture for highway scenarios. Neven et al. [6] built a network called LaneNet to distinguish each lane-border pixel. Garnett et al. [7] introduced a 3D LaneNet network architecture to predict the 3D layout of lanes from a single image. Although these neural-network-based methods performed well for multi-lane detection, they are limited in development by massive manually labeled datasets and extensive training on high-performance GPUs [8].

The digital map becomes more and more important for intelligent vehicles, which usually organizes index-aware lanes in tree data structures and supports spatial search of lanes. Map-enhanced sensor perception becomes a promising way for ego-lane index identification. A direct way to estimate the ego-lane index is localizing the vehicle precisely and then calculating the ego-lane index based on a lane-level map. To achieve accurate and consistent positioning performance, a common approach is to use the GNSS real-time kinematic (RTK) technique, but RTK often fails and results in serious errors owing to poor satellite geometry and disruption of radio signal reception because of skyscrapers in urban areas [9–11]. The map-matching method based on sensor data and a prior map is an alternative for precise localization. LiDAR sensor can achieve a high-precision position by matching LiDAR scans to a dense-point map [9,12–14]. However, the dense-point map requires expensive maintenance, as it is sensitive to seasonal changes and needs to be updated frequently. Some researchers used camera sensors to extract lane-related features and matched these features to a lane-level map to determine the vehicle pose and ego-lane index [15,16]. Lee et al. [17] integrated the ego-lane index information into the road-markings-based map-matching function using a DeepRoad network for ego-lane index identification. Choi et al. [18] identified in-lane localization using visual lane endpoints detection, the relation between a camera and a road, and the estimated vehicle's orientation from a map. They identified ego-lane by generating a hypothesis about an ego-vehicle position per lane by the in-lane localization and verifying each hypothesis by matching the lane endpoints and an additional landmark such as a road sign with the map. These vision-and-map-based methods are more cost effective but need pixel-level benchmarks on camera images and time-consuming labeling processes to assess their performance [19].

A map-based probabilistic framework for ego-lane index estimation is another option. Rabe et al. [20] generated hypotheses of trajectory with map data and lane changes to describe several driving possibilities, aligned the hypotheses to the trajectory generated from vehicle odometry and yaw rate in a weighted least-squares sense, and finally determined the likelihood of the ego-vehicle being on a certain lane. They also proposed a PF framework to this ego-lane index estimation problem [21]. Visual lane detection was used for in-lane localization and radar objects projected onto a map to update the weights of particles. These two methods were compared by real driving conditions with a mix of one- to four-lane segments and found that the optimization-based approach performed better after intersections, whereas it often suffered from ambiguities on straighter stretches. The PF method was further extended for downtown scenarios with many lanes [22]. Specifically, a combination of weight update and sampling steps was proposed for the lane-marking distance observations to avoid diverging lane probability estimates. Ballardini et al. [23] also proposed a probabilistic framework for highway-like scenarios using a hidden Markov model (HMM) with a transient failure model, which considered inaccurate or missing road marking detections. They used GNSS measure and the number of lanes of the road, retrieved from a map service OpenStreetMap prior to enhancement of the visual ego-lane index estimation [24]. Similarly, Kasmi et al. [25] also used the lane information from OpenStreetMap to assist ego-lane determination and introduced a modular Bayesian network (BN) to infer the ego-lane from multiple inaccurate information sources. Svensson et al. [26]

proposed a Bayesian filter to infer probability for the correct lane assignment by fusing lane-marking and vehicle detection data and map information.

All above works mainly relied on four sources of information: vision-based lane/road marking detection and radar-based adjacent vehicle detection, a map including lane geometry information, GNSS/IMU navigation data, and a filter method. Road boundary is an important map element and sensor-based road boundary detection can give us an estimate for the distances to the road boundaries, which is able to help infer ego-lane index. To take advantage of this information, in this paper, we proposed a PF framework using a LiDAR-based road boundary detection, a lane-level map, and GNSS/IMU/odometry navigation data. A lane-level map representation *Lanelets* was used in our work, where lanelets represent the drivable environment under both geometrical and topological aspects; more details about lanelets are presented in [27]. Navigation data and the distances to the road boundaries calculated from the LiDAR-based road boundary detection were used to update the particles. As the LiDAR-based road boundary detection can be determined accurately, it may put a tight constraint on the distribution of particles, which can result in sparse particle sets and performance deterioration. An important contribution of this paper is that we used the LiDAR measurement to shift particles laterally instead of updating the particle weight for each particle, which successfully avoids particle degradation.

2. Materials and Methods

In this work, LiDAR, GNSS, IMU, wheel odometer measurements, and a lane-level map were fused by PF. Figure 1 shows the flowchart of our methods. Inertial sensor measurements were obtained at high sampling rates, which can produce high-frequency position and orientation information by the mechanization of the inertial navigation system (INS). Velocity measurement from odometry and INS mechanization were loosely coupled to achieve the real-time pose of the vehicle, which is called dead reckoning. GNSS SPP can provide real-time position estimation, but it only has meter-level positioning accuracy in urban areas because of signal blockages and reflections. Particles were initialized around the initial pose and predicted by dead reckoning at each time step. Based on the predicted position and orientation of each particle, taking advantage of road geometry and topology properties of the lane-level map, ego-lane index, and distance to left/right road boundaries could be calculated for each particle. Particle weight was updated by comparing its position with GNSS SPP measurement. The distance measurement from LiDAR-based road boundary detection was used to shift particle position laterally and estimate particle ego-lane index again. To avoid degenerate particle sets, a residual resampling was applied whenever the effective sample size became too low. Finally, the lanelet id with the highest sum of particle weights was chosen as the final ego-lane index.

Various coordinate systems were used in this work, including the following:

1. INS Navigation System (n -frame)

The n -frame is a local geodetic frame with platform mass center as the coordinate origin. The X axis pointing towards the east, the Y axis pointing to the north, and the Z axis completing a righthanded coordinate system. It is also called an east–north–up (ENU) system, forming a right-handed coordinate system.

2. World Coordinate System (w -frame)

The w -frame is used to express the GNSS positioning results. The initial GNSS position is taken as the origin. The X - Y plane is the local horizontal plane, with the X axis pointing east and the Y axis pointing north. The Z axis is perpendicular to the X - Y plane, forming a right-handed coordinate system. The w -frame is parallel to the n -frame.

3. LiDAR Coordinate System (l -frame)

The l -frame moves with the vehicle, with the LiDAR measurement center as the coordinate origin, the X axis pointing to the right along the LiDAR horizontal axis, the Y axis pointing forward along the LiDAR longitudinal axis, and the Z axis being perpendicular to the X - Y plane, forming a right-handed coordinate system.

4. Map Coordinate System (m -frame)

The discrete map points use the w -frame.

Transformations among these coordinate systems are assumed to be rigid-body transformations.

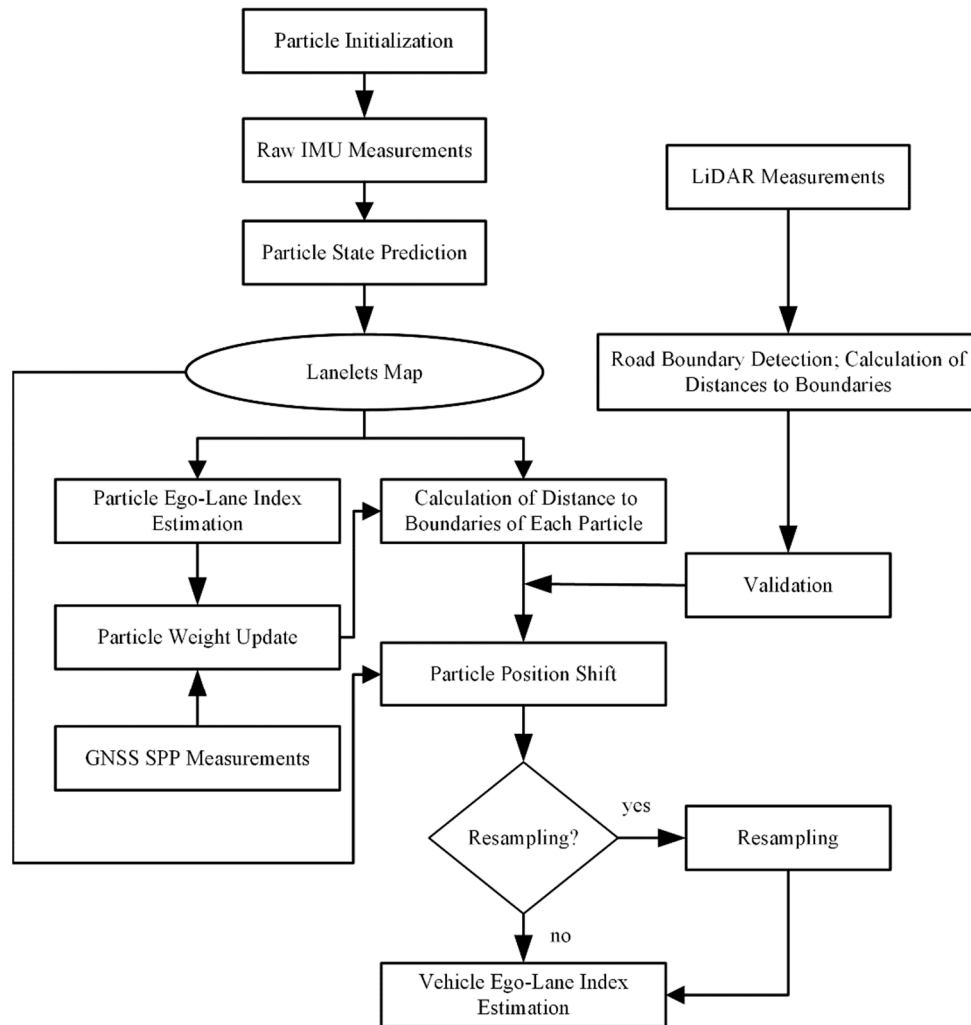


Figure 1. Flowchart of our method.

2.1. Digital Lane-Level Map

For lane-level navigation, we need a digital map containing geometric and topologic information of all available lanes to find the currently used lane. A lanelet map was used here, which was proposed by FZI Research Centre for Information Technology, Germany, in 2014 [27]. The lanelet map is organized by connected lanelets. A lanelet is defined by two polylines composed of a list of points as left and right boundaries, as shown in Figure 2. Except for the set of lanes and their interconnection, the lanelet map also contains the corresponding road traffic regulations, such as traffic signs, rules at intersections, and traffic lights. The vehicle must obey all the regulatory elements, which are linked to the lanelets when running on public roads. To make the spatial search in a lanelet map more efficient, R-tree is used to enable the spatial search for objects inside an arbitrary bounding box in $O(\log n)$.

In this work, we focused on measuring the distance of a pose to the bounds of a lanelet and getting the lanelet id of the pose, so that the regulatory elements were not considered. Neighboring and succeeding lanelets shared the same boundaries, and this

information could determine their topologic relationship and the number of total parallel lanes in a certain road segment. To construct a lanelet map, we collected accurate positions in the w -frame of discrete points of polylines of all lanelets in our experimental area. The data collection vehicle mounted high-precision GNSS, IMU, LiDAR, and camera sensors; therefore, our map data had very high accuracy. Finally, the map data were structured by the liblanelet, as presented in [28].

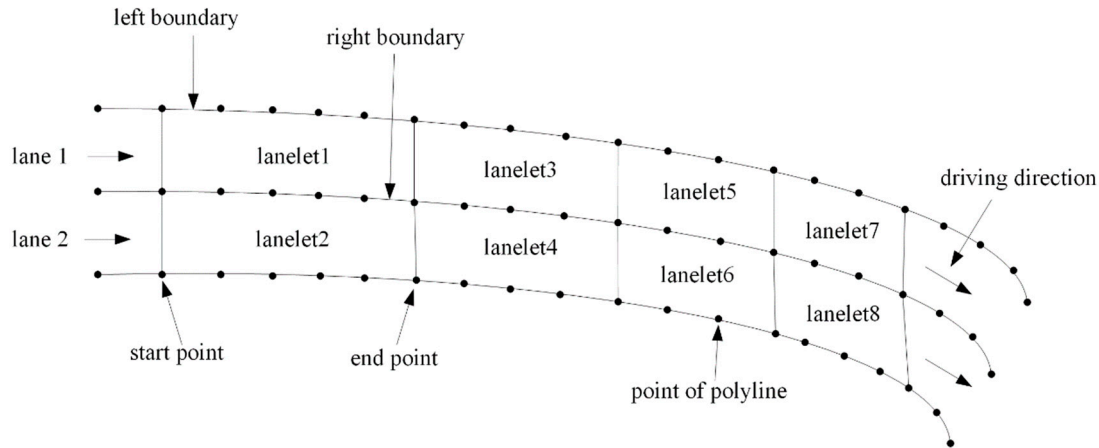


Figure 2. Structure of lanelets.

2.2. Particle Initialization and Prediction

The vehicle starts with a known initial position (b^0, l^0, h^0) in the w -frame, initial velocity (v_x^0, v_y^0, v_z^0) in the n -frame, and initial attitude angles $(\gamma^0, \theta^0, \varphi^0)$. Particle state X^t at the t th time step includes $(b^t, l^t, h^t, v_x^t, v_y^t, v_z^t, \gamma^t, \theta^t, \varphi^t, ll^t, d_{left}^t, d_{right}^t)$, where ll^t is the lanelet id where the vehicle locates in, which is calculated by function $f_1(b^t, l^t, h^t, \varphi^t)$ by searching for the lanelet map, d_{left}^t and d_{right}^t are the distances to the left and right road boundaries, which are calculated by function $f_2(b^t, l^t, h^t, ll^t)$ by fitting the polylines of boundaries as curves to get the distances from a point to curves. To initialize N particles, the j th particle X_j^0 is drawn from the initial probability density function (PDF) $p(X^0)$, which is assumed to be a Gaussian distribution. Afterwards, at each time step X_j^t is predicted by integrating INS and velocity from odometry.

INS mechanization could estimate the position, velocity, and attitude of a system from the raw data of the IMU if the initial pose is known. However, the estimation may contain large errors and drift rapidly because of the drift errors of the accelerometer and gyroscope. Performance of INS depends on IMU's grade. Usually, a low-cost IMU has larger drift errors and the navigation solution deviates from ground truth in a short time period. More details about INS mechanization can be found in [29]. Real-time velocity measurement from odometry can compensate for the drift errors of INS to a certain extent. An error propagation model is used to fuse INS and velocity data to obtain a better navigation solution. The error propagation model via a first-order Taylor expansion is defined as follows:

$$\begin{aligned} \delta x_j &= [\delta r_j^n, \delta v_j^n, \varepsilon_j^n, \delta b_{j,a}, \delta b_{j,g}] \\ \delta x_j^{t-} &= \phi_j^{t-1} \delta x_j^{t-1} + u_j^{t-1} \end{aligned} \quad (1)$$

where the error state δx includes position error δr^n , velocity error δv^n , attitude error ε^n in the n -frame, accelerometer drift δb_a , and gyroscope drift δb_g , ϕ_t is state transition matrix and u_t is the driven response of the input white noise at time t . Velocity measurement is used to correct the error state δx via the extended Kalman filter (EKF) [30] and feed it back

to the IMU mechanization for estimating the final navigation state. The EKF observation functions are

$$\begin{aligned} z_j^t &= H\delta x_j^t + v_j^t \\ z_j^t &= v_{j,IMU}^n - v_{odo}^n \end{aligned} \quad (2)$$

where z is the observation vector, H relates the error state vector and the measurements, v is a model error. Based on the prediction model in Equation (1), the error state vector corresponds to an error covariance P , which is predicted as

$$P_j^{t-} = \phi_j^{t-1} P_j^{t-1} \phi_j^{t-1T} + \Delta_j^{t-1} \quad (3)$$

EKF updates the error state vector as

$$\begin{aligned} \delta x_j^t &= \delta x_j^{t-} + K(z_j^t - H\delta x_j^{t-}) \\ P_j^t &= (I - KH)P_j^{t-} \end{aligned} \quad (4)$$

where K is Kalman gain. Finally, X_j^t can be estimated by

$$\begin{aligned} [b_j^t, l_j^t, h_j^t] &= C_n^w [r_j^{n,t-} - \delta r_j^{n,t}] \\ [v_{j,x}^t, v_{j,y}^t, v_{j,z}^t] &= C_n^w [v_j^{n,t-} - \delta v_j^{n,t}] \\ [\gamma_j^t, \theta_j^t, \varphi_j^t] &= [\gamma_j^{t-}, \theta_j^{t-}, \varphi_j^{t-}] - \varepsilon_j^{n,t} \\ ll_j^t &= f_1(b_j^t, l_j^t, h_j^t, \varphi_j^t) \\ [d_{j,left}^t, d_{j,right}^t] &= f_2(b_j^t, l_j^t, h_j^t, ll_j^t) \end{aligned} \quad (5)$$

where C_n^w is the rotation matrix.

In the scheme of PF, a weight is assigned to each particle X_j^t to approximate the posterior PDF

$$w_j^t = w_j^{t-1} \frac{p(z^t | X_j^t) p(X_j^t | X_j^{t-1})}{q(X_j^t | X_j^{t-1}, z^t)} \quad (6)$$

where $p(z^t | X_j^t)$ is the likelihood of observation z^t , $p(X_j^t | X_j^{t-1})$ is transition PDF, and $q(X_j^t | X_j^{t-1}, z^t)$ is proposal function. The calculation of weights of particles will be described in Section 2.4.

2.3. LiDAR-Based Road Boundary Detection

In this work, we used the LiDAR-based road boundary detection method proposed in [31]. This method provides a speed and accuracy tradeoff solution for road boundary detection in structured environments. Ground points are firstly segmented from raw LiDAR points by a RANSAC plane fitting method to separate off-ground and on-ground points since road boundaries are a part of the ground. Feature points are then extracted from the ground points. Three spatial features are used to extract the feature points: height difference between a point and neighbors, smoothness of the area around a point, and horizontal distance between two adjacent points. To classify the feature points as road boundary points, a road-segmentation-line-based method is used. After feature points are extracted and classified, there are still many false points, including vehicles, pedestrians, railway tracks, adjacent roads, etc. An iterative Gaussian process regression (GPR) algorithm is employed to model road boundary and filter out false points. Finally, the left and right boundaries are fitted as linear lines in the form of $ax + by + c = 0$ in the l -frame. To calculate the distance from the LiDAR to the road boundary, the closest point from the fitting line to the LiDAR is acquired.

This method is designed for point cloud from “Velodyne HDL-64E” LiDAR so that it contains some specific parameters that suit for “HDL-64E”, for example, the number of scan layers, the vertical azimuth of scanning layer, and the horizontal angular resolution. To apply this method to our “Velodyne HDL-32E” LiDAR, we changed the source codes to improve its availability. The source code can be found in [32].

As road boundary detection provides distance measurements for PF in our work, detection results should be validated. Three indicators were used: the coefficient a should be less than 0.005, the number of effective points after the iterative GPR algorithm should be larger than 20, and the difference between distances calculated at two adjacent time steps should be around v/f , where v is lateral velocity and f is frequency. If road boundary detection was invalid, we assumed that there were no distance measurements at the particle update step.

2.4. Particle Update and Ego-Lane Index Estimation

First, GNSS SPP with a meter-level positioning accuracy was used to update particle weight. The distance d_j between the j th particle and the SPP position is calculated and compared with a threshold d_{\max} , which was determined by the positioning precision of SPP and set to 10 m in this work, resulting in a GNSS weight w_j^t

$$w_{j,\text{position}}^t = \begin{cases} 0, & \text{if } d_j > d_{\max} \\ w_j^{t-1}, & \text{if } d_j \leq d_{\max} \end{cases} \quad (7)$$

In a general PF scheme, the distances calculated from LiDAR road boundary detection can be used for the importance weighting with a Gaussian likelihood

$$w_{j,\text{LiDAR}}^t = \exp\left(-\frac{(d_{\text{left,LiDAR}}^t - d_{\text{left,LiDAR}}^t)^2 + (d_{\text{right,LiDAR}}^t - d_{\text{right,LiDAR}}^t)^2}{2\delta_{\text{distance}}^2}\right) \quad (8)$$

where $\delta_{\text{distance}}^2$ is the error variance of distance calculation from LiDAR road boundary detection, which is set to 0.1 in this work. Finally, for each particle, its overall weight w_j is calculated as follows:

$$w_j^t = w_{j,\text{LiDAR}}^t w_{j,\text{position}}^t \quad (9)$$

As the distance to the road boundaries can be determined very accurately by LiDAR, the likelihood in Equation (8) can be tight, compared to the particle distribution after the prediction step. High weights will be assigned to only a few particles, and many particles will be deleted in the next resampling step. Generally, this is the desired effect, as the filter may converge quickly. However, the likelihood in Equation (8) is tight only in the lateral position but arbitrarily wide in the longitudinal position. In this case, the few surviving particles may not represent the actual distribution. Inspired by [22], we used a different approach in this work.

Distance measurements were used to update the particle position instead of particle weights. Since the distances from the vehicle to road boundaries were known, we shifted each particle laterally such that its lateral position matched the distance measurements. New distances became

$$\begin{aligned} d_{j,\text{left}}^t &= d_{\text{left,LiDAR}}^t + \zeta_{\text{left}}, \zeta_{\text{left}} \sim N(0, \delta_{\text{distance}}^2) \\ d_{j,\text{right}}^t &= d_{\text{right,LiDAR}}^t + \zeta_{\text{right}}, \zeta_{\text{right}} \sim N(0, \delta_{\text{distance}}^2) \\ (b_j^t, l_j^t, h_j^t) &= f_3(d_{j,\text{left}}^t, d_{j,\text{right}}^t) \end{aligned} \quad (10)$$

where function f_3 can calculate position from distance measurements via geometrical calculation using the map. New lanelet id is calculated by new position (b_j^t, l_j^t, h_j^t) with function $f_1(b^t, l^t, h^t, \varphi^t)$. As lane widths are stipulated by regulations, and they can be known in advance, if the lane number of a road is known, the road width is fixed. Therefore,

either $d_{j,left}^t$ or $d_{j,right}^t$ is used in Equation (10) in practice, and the other one is calculated by road width. A residual resampling, which eliminates particles with low weights while copies particles with high weights, is applied whenever the effective sample size becomes too low. Finally, the lanelet id with the highest sum of particle weights is chosen as the final lane index.

3. Experimental Results

Field experiments were carried out to test our PF framework of ego-lane index estimation. The test vehicle we used is a Chery Tiggo car equipped with GNSS operating at 1 Hz, IMU at 600 Hz, and LiDAR at 10 Hz (shown in Figure 3a). “Trimble BD982” was used for GNSS positioning, which supports BDS/GPS/Galileo/GLONASS and multiple GNSS frequencies. “Velodyne HDL-32E” LiDAR used in this work has a horizontal angle range $[0^\circ, 360^\circ]$ and a vertical angle range $[-30^\circ, 10^\circ]$ with 32 scans. “Honeywell HG4930” is a very high-performance micro-electromechanical system (MEMS) IMU, and its angular and velocity random walk are $0.04^\circ/\sqrt{h}$ and $0.3 \mu g/\sqrt{Hz}$, respectively. The odometry was “SICK Incremental encoders DFS60”, which was mounted on the left rear wheel of the vehicle. The encoders can measure 2048 pulse per revolution so that they can produce high-precision velocity measurements.

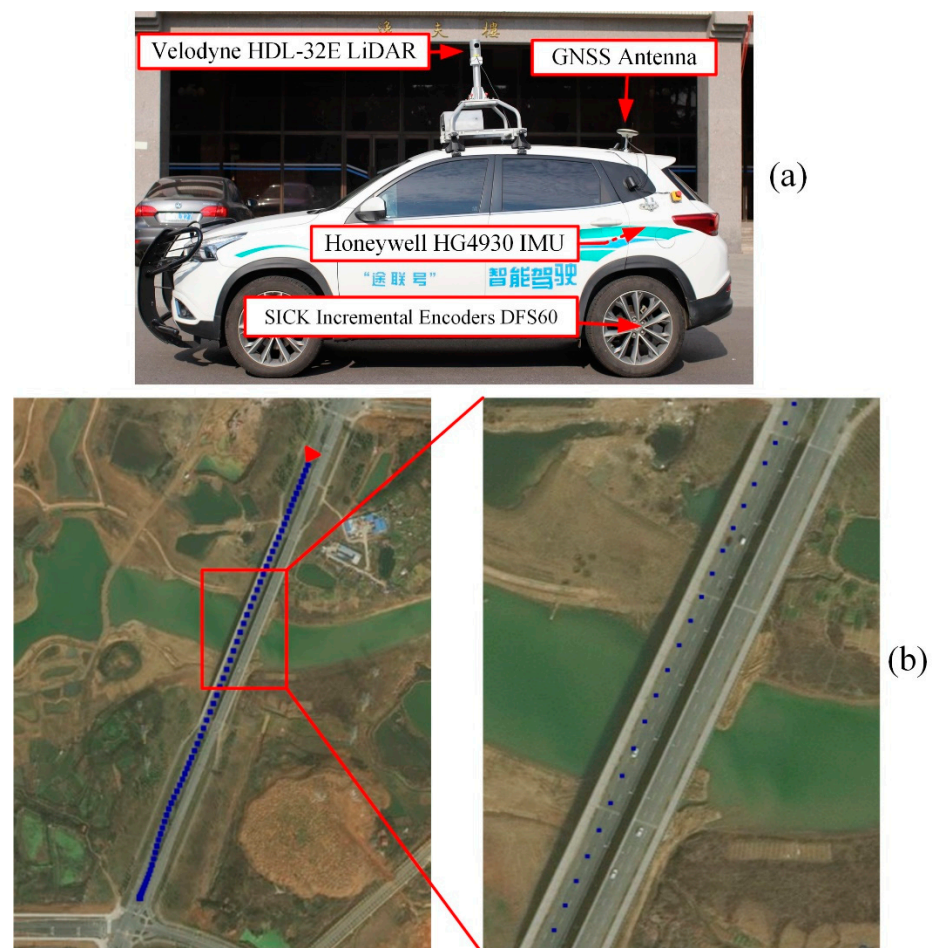


Figure 3. The vehicle and sensors used in our work (a) and an overhead satellite image of the test region (b). GNSS RTK trajectory of the test route is denoted by blue dots. The red arrow shows the starting point and the driving direction. To be more specific, the area in the red frame is zoomed in.

Our experimental field was on the Huashan road in Wuhan, China with an open-sky environment for GNSS. Therefore, both GNSS RTK and SPP can achieve high-precision

positioning estimation. In order to evaluate the performance of our method in dense urban environments, where GNSS positioning is poor because buildings block, reflect, and diffract the signals, we simulated a high-occlusion environment (satellite signals with elevation angle in $[0, 40^\circ]$ are blocked, and random noises are added to pseudo-range measurements). GNSS RTK results from the open-sky environment are regarded as ground truth positions of vehicles with centimeter-level accuracy, which is denoted with blue dots in Figure 3b.

SPP results from the simulated high-occlusion environment provide real-time GNSS measurements for our method. Figure 4 shows the sky plot of visible satellites of GPS + BDS + GLONASS above the platform's horizon at the first epoch. In the simulated high-occlusion environment, the satellites with elevation angles lower than 40° are eliminated, and the positioning precision becomes a few meters. Figure 5 shows the positioning errors and yaw errors in the n -frame of GNSS SPP results, and dead reckoning results in the high-occlusion environment by comparison with the ground truth of GNSS RTK in the open-sky environment. We can observe that horizontal errors of SPP can be as large as 5 m and exhibit lateral positioning deviations of up to a few lanes. The vertical accuracy is about 2 times worse than the horizontal accuracy because the GNSS receiver can only track satellites in two quadrants of its vertical plane which does not provide the same strength of geometric calculations [33]. The positioning precision of dead reckoning is a little worse than the GNSS SPP measurement. Yaw angle is essential for the calculation of lanelet id as shown in Equation (5). Therefore, yaw angle errors from dead reckoning are also shown in the figure and the errors increased within time to up to -20° .

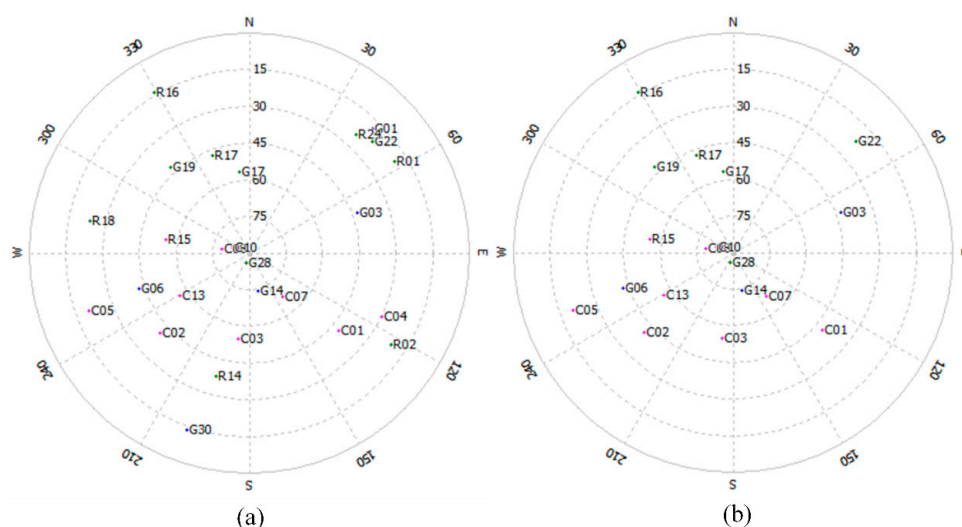


Figure 4. Sky plot of visible satellites of GPS + BDS in the open-sky environment (a) and in the simulated high-occlusion environment (b) at the first epoch.

Figure 6 shows a part of our digital lane-level map, and the lines in the map represent the polylines of the left and right boundaries of lanes. Accurate discrete points in the polylines are acquired by high-precision navigation and IMU equipment. The points are then organized with the lanelet library. There are about 20 lanelets in our test region. The test road includes three or four parallel lanes, as shown in the right images in Figure 6. Our trajectory changes lanes frequently in purpose. Figure 7 shows the lanelet ids at each time step calculated by the true trajectory of GNSS RTK in the open-sky environment and by the biased trajectory from the dead reckoning. It can be seen that ego-lane index estimation is wrong at many time steps because of biased positions and yaw angles, and the vehicle is assumed to be off-road at some time steps, with lanelet id being 0, which is dangerous for intelligent vehicles, as it may result in wrong decisions.

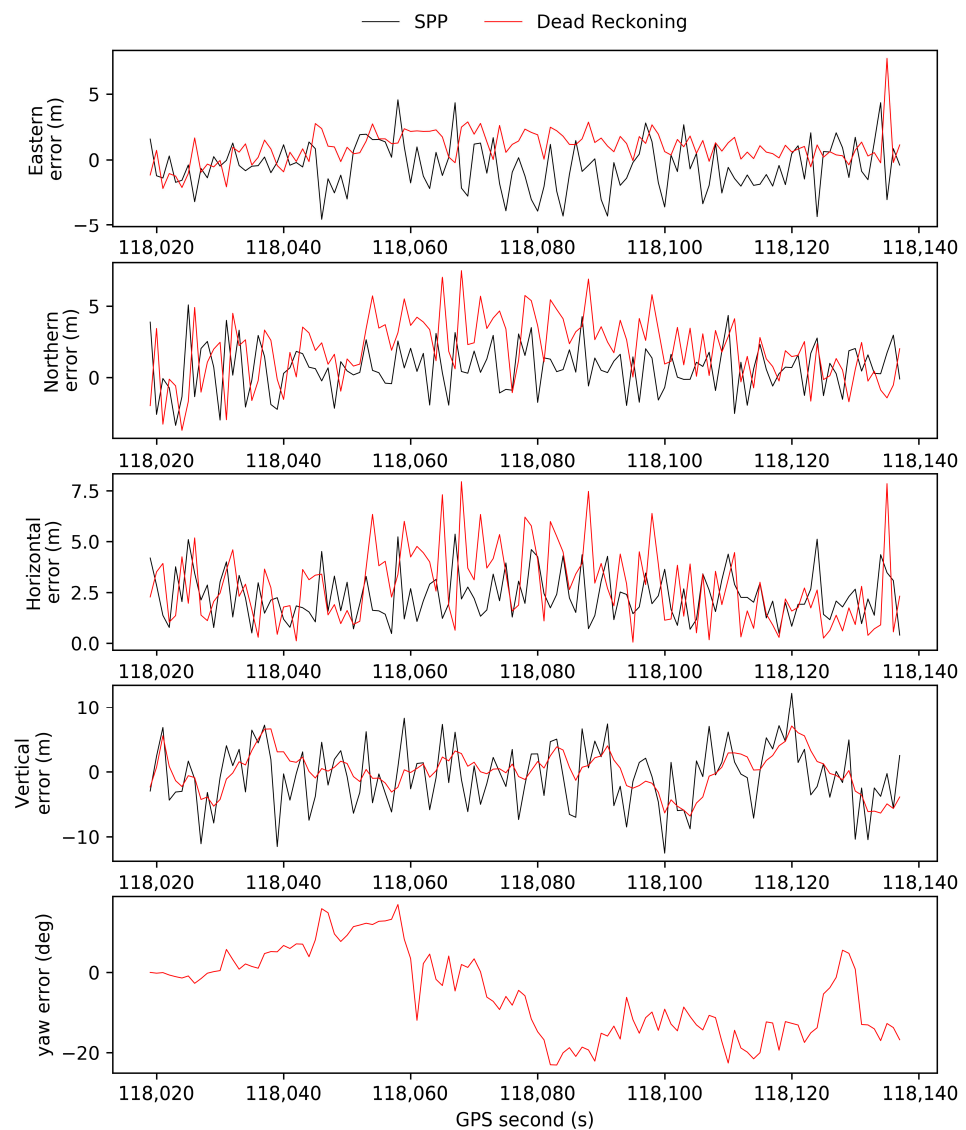


Figure 5. Positioning errors in the eastern, northern, horizontal, and vertical directions of GNSS SPP in the simulated high-occlusion environment and the INS/odometry dead reckoning.

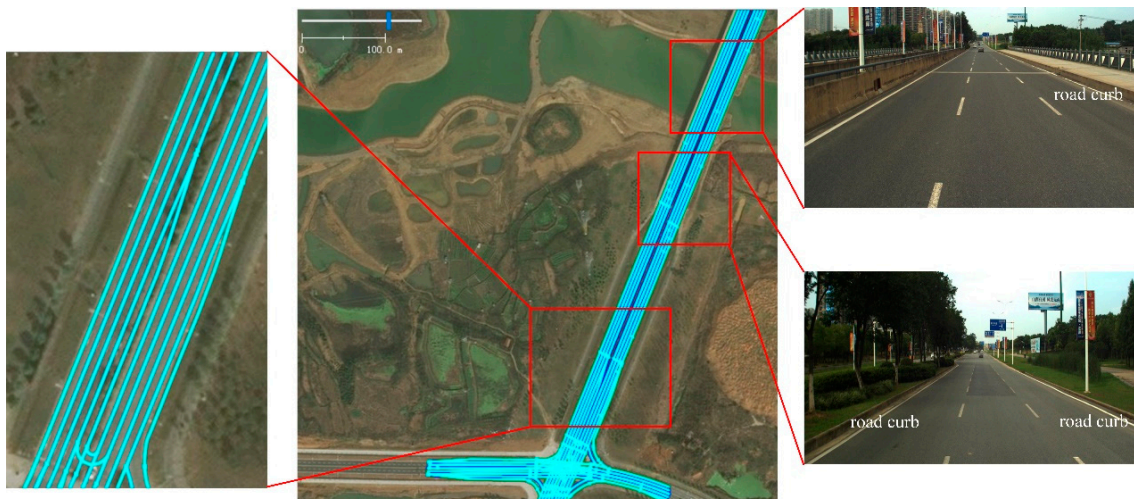


Figure 6. A part of the digital lane-level map used in our work; the area in the red frame is zoomed in.

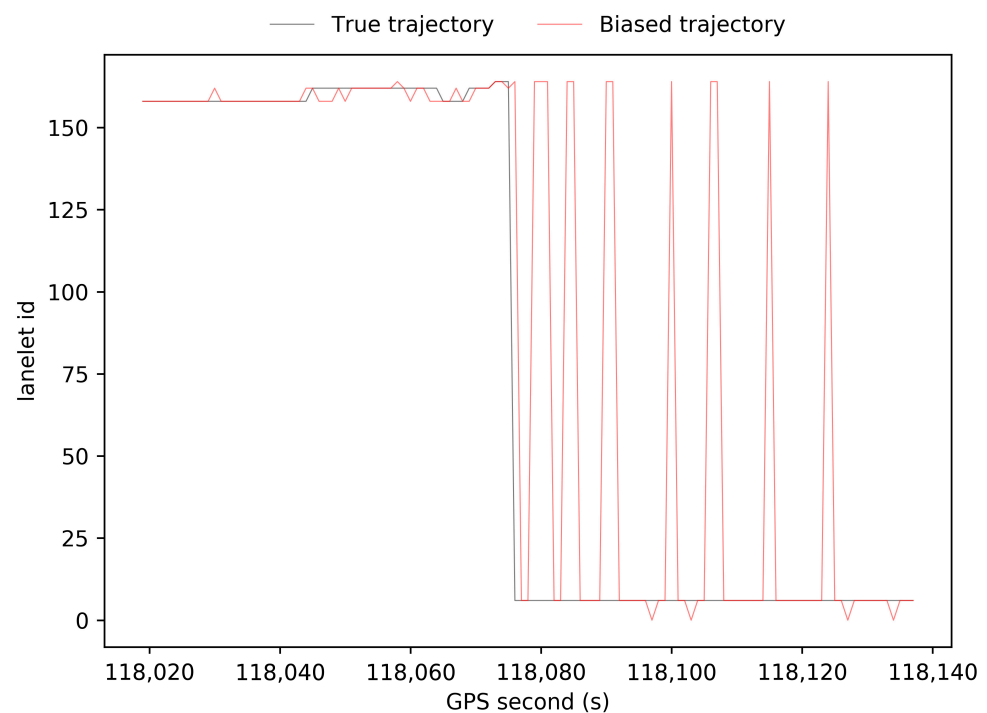


Figure 7. Evolution of lanelet ids of the true trajectory and the biased trajectory from dead reckoning.

From the right images in Figure 6, we can also infer that the road is highly structured with road curbs. There is a bridge in the trajectory which contains a road curb on one side. In other places of the trajectory, the curbs are distributed on both sides. Therefore, the LiDAR-based road boundary detection method performs well in our work. In most places of the trajectory, both left and right road boundaries are detected correctly, as shown in Figure 8a. However, the moving vehicles on the road disturbed the detection results as road curbs cannot be correctly extracted from the LiDAR points as shown in Figure 8b–f. In Figure 8b, only one boundary is detected, which can be used in Equation (10) as the LiDAR measurement for the left boundary. In Figure 8c,d, the left and right road boundaries are detected, but they are not used for Equation (10), as they do not satisfy the three indicators described in Section 2.3—the coefficients of the fitting lines of left road boundaries are larger than 0.005, and the number of effective points for the right road boundaries is less than 20. For some cases, road boundaries are still effectively detected despite moving obstacles, as shown in Figure 8e,f, in which either left or right road boundary detection satisfies the indicators. To avoid the bad effects of the moving vehicles on the LiDAR-based road boundary detection, we chose a low traffic-volume period.

Figure 9 shows the lanelet ids at each time step calculated by the ground truth of GNSS RTK in the open-sky environment, by the results of our improved PF scheme, and by the results of the traditional PF scheme in the simulated high-occlusion environment. Compared to the results of the biased trajectory from dead reckoning in Figure 7, both PF schemes have positive effects on the ego-lane index estimation, taking advantage of distance measurements from LiDAR detection. Our improved PF scheme finds the correct lane index at all time steps, but there are still some wrong estimations in the results of the traditional PF scheme. Figure 10a shows the distribution of particles at the 50th time step after update by Equation (10) in our improved PF scheme (yellow squares) and the particles at the 50th time step after update by Equation (8) in the traditional PF scheme (black squares). First, 100 particles are randomly initialized around the true location over a four-lane road segment following a uniform distribution over the whole segment. Then, lateral distances to road boundaries are observed and importance weight updating according to the likelihood shown in Equation (8) is applied repeatedly. The particle set converges to the black squares shown in Figure 10a after 50 steps of prediction, importance

weight updating, and resampling. One can see that there are only four particle sets are left and particle degradation occurs. The diversity of particles is worse and badly represents the posterior PDF of the position of the vehicle based on sensor measurements, which leads to the wrong estimations of ego-lane.

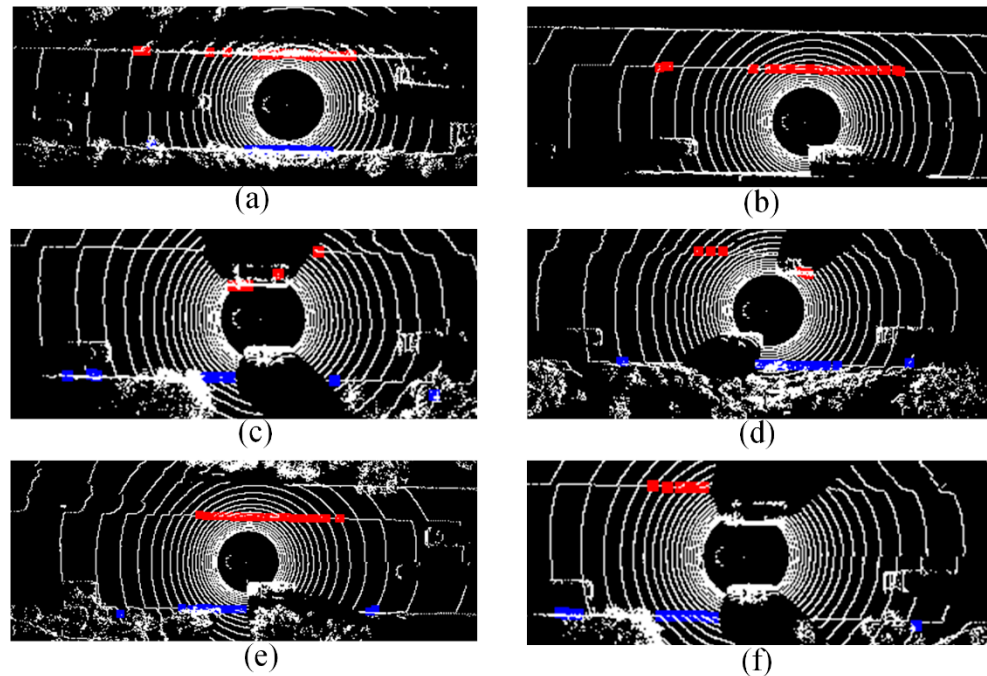


Figure 8. Some results of LiDAR-based road boundary detection and (a–f) show different scenarios; red dots mean left road boundary, and blue dots represent right road boundary.

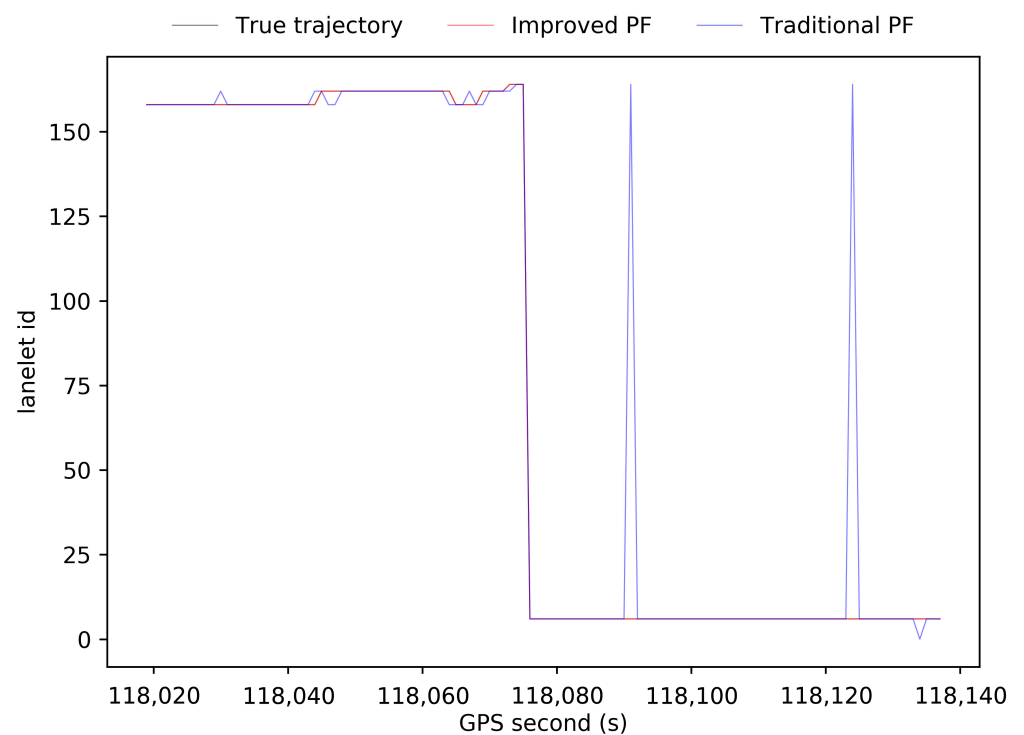


Figure 9. Evolution of lanelet ids of the true trajectory, the results from our improved PF scheme, and the results from the traditional PF scheme.

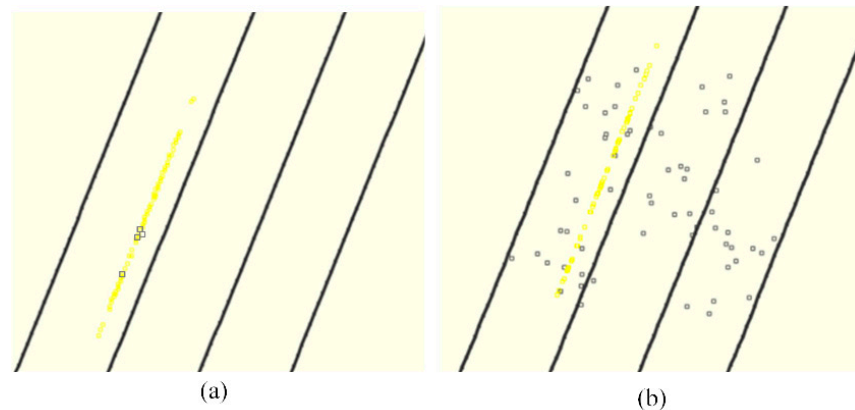


Figure 10. Distribution of particles on the lane-level map: (a) shows the particles at the 50th time step after update in our improved PF scheme (yellow squares) and in the traditional PF scheme (black squares); (b) shows the particles at the 3rd time step before update (black squares) and after update (yellow squares) in our improved PF scheme.

Our improved PF scheme relieves this problem by shifting each particle laterally such that its lateral position meets the distance calculation from LiDAR detection instead of importance weight updating. Using this approach, the particle distribution evolves to the one shown with yellow squares in Figure 10a after 50 filter steps. Figure 10b shows particles after prediction with black squares and after lateral shift with yellow squares at the third time step. We can deduce that the longitudinal distribution of particles represents the posterior probability much better than with the traditional importance weighting: the 100 particles (yellow squares in Figure 10a) spread rather uniformly along the road, while before the update (black squares in Figure 10a), they converge to only a few particle sets, leaving large areas without any particles. This meets the knowledge from LiDAR road boundary detection which puts strong constraint in the lateral direction but no constraint in the longitudinal direction. Therefore, our approach keeps the diversity of particles and leads to a more stable and reproducible behavior of the PF without increasing the number of particles.

4. Conclusions and Discussion

In this work, a novel PF scheme for ego-lane index estimation was presented with sensors that are readily available in an intelligent vehicle: GNSS, IMU, odometry, and LiDAR sensors, combined with a lane-level map which is organized with a lanelet library. The lane index is presented as lanelet id. In total, 100 particles were initialized and propagated via dead reckoning by IMU and odometry. In urban scenarios, GNSS signals may be blocked by high buildings or trees leading to meter-level positioning accuracy, which is insufficient for the intelligent vehicle. However, it can limit the drift of dead reckoning. A LiDAR-based road boundary detection proposed in [31] was applied for calculating distances from the vehicle to left and right road boundaries. The high-precision distance measurement puts a tight constraint on particles in the lateral direction while putting arbitrarily a wide constraint in the longitudinal direction. The traditional importance weight updating process in PF can hardly deal with this problem, leading to particle degradation. A novel update step was proposed by shifting particles laterally to place corresponding to the distance measurements so that particle distribution can correctly represent the posterior PDF of the position of the vehicle. Together with the support of topologic and geometric information of lanes from the map, a reliable ego-lane index estimation could be performed. Field experiments prove that our method can correctly estimate the ego-lane index during the trip.

Our system greatly relies on accurate road boundary detection which gives important inference of the lateral position where a vehicle locates on the map. However, the road

boundary detection method used in this paper is badly affected by moving obstacles, as LiDAR points hit the obstacles rather than road curbs so that positions of curbs become unknown. In future works, we will focus on solving road boundary detection under traffic jams, for example, using deep learning to predict the position of the road curb when it is blocked by moving obstacles. To make our approach more practicable, we will try low-cost MEMS and test our approach with more scenarios, such as downtown scenarios with complex intersections.

Author Contributions: Conceptualization, H.Z.; data curation, W.L. and C.Q.; funding acquisition, B.Y.; methodology, H.Z., W.L., C.Q. and B.L.; supervision, B.L. and C.W.; writing—original draft preparation, H.Z.; writing—review and editing, B.Y., W.L., C.Q., B.L. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Project No. U1764262, Project No. 41801377) and the Key-Area Research and Development Program of Guangdong Province (Project No. 902180320175).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Zhao, K.; Meuter, M.; Nunn, C.; Müller, D.; Müller-Schneiders, S.; Pauli, J. A novel multi-lane detection and tracking system. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 1084–1089.
- Kang, S.-N.; Lee, S.; Hur, J.; Seo, S.-W. Multi-lane detection based on accurate geometric lane estimation in highway scenarios. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 221–226.
- Hur, J.; Kang, S.-N.; Seo, S.-W. Multi-lane detection in urban driving environments using conditional random fields. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 1297–1302.
- Chao, F.; Yu-Pei, S.; Ya-Jie, J. Multi-lane detection based on deep convolutional neural network. *IEEE Access* **2019**, *7*, 150833–150841. [[CrossRef](#)]
- Huval, B.; Wang, T.; Tandon, S.; Kiske, J.; Song, W.; Pazhayampallil, J.; Andriluka, M.; Rajpurkar, P.; Migimatsu, T.; Cheng-Yue, R. An empirical evaluation of deep learning on highway driving. *arXiv* **2015**, arXiv:1504.01716.
- Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards end-to-end lane detection: An instance segmentation approach. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 286–291.
- Garnett, N.; Cohen, R.; Pe'er, T.; Lahav, R.; Levi, D. 3d-lanenet: End-to-end 3d multiple lane detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2921–2930.
- Zhu, D.; Song, R.; Chen, H.; Klette, R.; Xu, Y. Moment-based multi-lane detection and tracking. *Signal Process. Image Commun.* **2021**, *95*, 116230. [[CrossRef](#)]
- Qian, C.; Zhang, H.; Li, W.; Tang, J.; Liu, H.; Li, B. Cooperative GNSS-RTK Ambiguity Resolution with GNSS, INS, and LiDAR Data for Connected Vehicles. *Remote Sens.* **2020**, *12*, 949. [[CrossRef](#)]
- Ruchti, P.; Steder, B.; Ruhnke, M.; Burgard, W. Localization on openstreetmap data using a 3d laser scanner. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5260–5265.
- Raaijmakers, M.; Bouzouraa, M.E. In-vehicle roundabout perception supported by a priori map data. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 15–18 September 2015; pp. 437–443.
- Wolcott, R.W.; Eustice, R.M. Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. *Int. J. Robot. Res.* **2017**, *36*, 292–319. [[CrossRef](#)]
- Liu, H.; Ye, Q.; Wang, H.; Chen, L.; Yang, J. A Precise and Robust Segmentation-Based Lidar Localization System for Automated Urban Driving. *Remote Sens.* **2019**, *11*, 1348. [[CrossRef](#)]
- Tang, J.; Chen, Y.; Niu, X.; Wang, L.; Chen, L.; Liu, J.; Shi, C.; Hyppä, J. LiDAR scan matching aided inertial navigation system in GNSS-denied environments. *Sensors* **2015**, *15*, 16710–16728. [[CrossRef](#)] [[PubMed](#)]
- Du, X.; Tan, K.K. Comprehensive and practical vision system for self-driving vehicle lane-level localization. *IEEE Trans. Image Process.* **2016**, *25*, 2075–2088. [[CrossRef](#)] [[PubMed](#)]
- Cui, D.; Xue, J.; Zheng, N. Real-time global localization of robotic cars in lane level via lane marking detection and shape registration. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1039–1050. [[CrossRef](#)]

17. Lee, S.; Choi, J.; Seo, S.W. Ego-lane index-aware vehicular localisation using the DeepRoad Network for urban environments. *IET Intell. Transp. Syst.* **2021**, *15*, 371–386. [CrossRef]
18. Choi, K.; Suhr, J.K.; Jung, H.G. In-lane localization and ego-lane identification method based on highway lane endpoints. *J. Adv. Transp.* **2020**, *2020*, 8684912. [CrossRef]
19. Nguyen, T.T.; Spehr, J.; Xiong, J.; Baum, M.; Zug, S.; Kruse, R. A survey of performance measures to evaluate ego-lane estimation and a novel sensor-independent measure along with its applications. In Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea, 16–18 November 2017; pp. 239–246.
20. Rabe, J.; Meinke, M.; Necker, M.; Stiller, C. Lane-level map-matching based on optimization. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1155–1160.
21. Rabe, J.; Necker, M.; Stiller, C. Ego-lane estimation for lane-level navigation in urban scenarios. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 896–901.
22. Rabe, J.; Hübner, M.; Necker, M.; Stiller, C. Ego-lane estimation for downtown lane-level navigation. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1152–1157.
23. Ballardini, A.L.; Cattaneo, D.; Izquierdo, R.; Parra, I.; Sotelo, M.; Sorrenti, D.G. Ego-lane estimation by modeling lanes and sensor failures. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–7.
24. Ballardini, A.L.; Cattaneo, D.; Izquierdo, R.; Parra Alonso, I.; Piazzoni, A.; Ángel Sotelo, M.; Sorrenti, D.G. Vehicle Ego-Lane Estimation with Sensor Failure Modeling. *arXiv* **2020**, arXiv:2002.01913.
25. Kasmi, A.; Denis, D.; Aufrère, R.; Chapuis, R. Probabilistic framework for ego-lane determination. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1746–1752.
26. Svensson, D.; Sörstedt, J. Ego lane estimation using vehicle observations and map information. In Proceedings of the 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 19–22 June 2016; pp. 909–914.
27. Bender, P.; Ziegler, J.; Stiller, C. Lanelets: Efficient map representation for autonomous driving. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 420–425.
28. Lanelet2. Available online: <https://github.com/fzi-forschungszentrum-informatik/Lanelet2> (accessed on 11 September 2021).
29. Shin, E.-H. Estimation Techniques for Low-Cost Inertial Navigation. Ph.D. Thesis, The University of Calgary, Calgary, AB, Canada, 2005.
30. Ribeiro, M.I. Kalman and extended kalman filters: Concept, derivation and properties. *Inst. Syst. Robot.* **2004**, *43*, 46.
31. Wang, G.; Wu, J.; He, R.; Tian, B. Speed and accuracy tradeoff for LiDAR data based road boundary detection. *IEEE/CAA J. Autom. Sin.* **2020**, *8*, 1210–1220. [CrossRef]
32. LidarRoadBoundaryDetection. Available online: <https://github.com/wangguojun2018/LidarRoadBoundaryDetection> (accessed on 11 September 2021).
33. Li, X.; Zhang, X.; Ren, X.; Fritsche, M.; Wickert, J.; Schuh, H. Precise positioning with current multi-constellation global navigation satellite systems: GPS, GLONASS, Galileo and BeiDou. *Sci. Rep.* **2015**, *5*, 8328. [CrossRef] [PubMed]