Intelligent Vehicle Self-Localization Based on Double-Layer Features and Multilayer LIDAR

Intelligent Vehicle Self-Localization Based on Double-Layer Features and Multilayer LIDAR

Zhuping Wang, Jian Fang, Xinke Dai, Hao Zhang* and Ljubo Vlacic

***Abstract*—High-precision self-localization is one of the basic functions of intelligent vehicles. Moreover, location information is necessary prior information for tasks such as behavioural decision-making and path planning. Recently, the technology of map-based localization has been widely concerned due to its accuracy. A prior map for localization stores one or more environmental features extracted by on-board sensors. These features represent fundamental information about the ground environment such as curbs and lane markings, or the vertical environment such as building outlines. In order to enhance the robustness and positioning accuracy, this paper uses two layers of features to express the environment: one is a bottom layer composed of ground and curb features, and the other is an upper layer composed of a 2D point cloud with vertical features. Firstly, a novel collision-based detection method is proposed to extract the curb features, and a vertical-projection-based detection algorithm is used to detect vertical feature points from multilayer LIDAR. Then, the fusion of a multi-frame feature point cloud is used to represent the rich and complete information on the environment of the intelligent vehicle. Finally, the Monte Carlo localization algorithm is used to obtain an optimal estimate of the vehicle position. The experimental results indicate that the proposed localization algorithm can realize localization accuracy.**

***Index Terms*—intelligent vehicle, self-localization, map-based, curb detection, vertical feature, multilayer LIDAR**

I. Introduction

Intelligent vehicle is one of the most popular research topics in the field of mobile robot. By achieving L5 level automated driving, intelligent vehicles improve road safety and transportation efficiency under sound infrastructure conditions. Intelligent vehicles possess certain basic systems, such as autonomous navigation, obstacle detection, behavioural decision-making, and localization. The precise position of the vehicle is important because it determines where the vehicle is located in the environment.

In general, high-precision positioning results can be obtained through global navigation satellite system (GNSS) devices. In [1], Knaup et al. summarize how the GNSS devices and inertial measurement units (IMUs) have been used in robotic positioning systems over the past few decades. These methods can achieve high accuracy in open areas. However, obstacles such as trees, buildings, and tunnels may affect the reception of satellite signals; consequently, the performance may deteriorate in urban areas, as discussed in [2].

Another approach to achieve high-precision positioning is to use map information. This method matches sensor data with the aprior map and selects the best match as the optimal estimate of the vehicle position. Generally, environmental features such as road edges, lane markings, and building outlines are pre-stored in apriori map and used to represent environmental information. With improved map accuracy, researchers have proposed a number of map-based vehicle location algorithms. Based on different sensors , the methods of map-based positioning can be divided into vision-based positioning and LIDAR-based positioning. The vision-based approach is sensitive to weather, light, and shadows. In [3], the authors highlight that the detection results would be affected if the camera is too close or the sensing range is too far, which is not a problem for LIDAR sensors. In [4], the paper proposes a monocular forward-view vehicle-mounted camera, but it requires point-wise LIDAR measurements and manually exhaustive labels as its ground truth. Road-SLAM [5] mentions that the influence of surrounding object shadows can be significant in some cases. The comparison in [6] is mainly based on the new camera system and low resolution Velodyne scanning, but we use a better multilayer LIDAR. A vision-based positioning methods is provided in both [7] and [8] suggesting that the additional detectors will improve the results, especially in areas with lots of parked vehicles along the road. In contrast, the laser sensor offers higher precision , a wider field of vision and is insensitive to lighting conditions. Thus, this approach has attracted the attention of an increasing number of researchers.

Urban features that are significantly distinct are frequently utilized for vehicle self-localization and map representation. In [9], building outlines are used as environmental features and associated with road/building maps for vehicle localization. When the GPS signals are blocked, the road map can help to bound the vehicle pose around the road; and the building map can help the vehicle adjust from coarse to fine tuning in several areas. In the vehicle localization method proposed in [10], the researchers use region descriptors extracted from occupancy grid maps as the features for localization. These region descriptors are obtained by the maximally stable extremal regions algorithm. For orchard environment, the authors in [11] propose a method for tree-crop identification and agricultural robot localization using LIDAR. In [12], the authors use an accumulated multilayer 2D vector map of a building as the location information, and obtain the location of the vehicle

This work was supported by the National Natural Science Foundation of China (U1764261 and 61733013), a Shanghai International Science and Technology Cooperation Project (18510711100), a Shanghai Shuguang Project (18SG18) and the Fundamental Research Funds for the Central Universities.

Z. Wang, J. Fang, Xinke Dai and H. Zhang are with the Department of Control Science and Engineering, Tongji University, Shanghai, 200092, P. R. China (e-mail: elewzp@tongji.edu.cn, 1631547@tongji.edu.cn, 1830774@tongji.edu.cn, zhang_hao@tongji.edu.cn).

L. Vlacic is with the School of Engineering, Griffith University, Gold Coast, QLD 4222, Australia (e-mail: L. Vlacic@griffith.edu.au).

through map matching.

In all of the works discussed above, the researchers use the characteristics of vertical space as positioning information, and the subsequent literatures mainly focus on improving positioning performance based on road information. In [13], Guan et al. propose an algorithm to automatically extract road information from mobile laser scanning point cloud data such as road markings, road surfaces, and road curbs. A lane-map-based localization algorithm using LIDAR sensors is introduced by Kim et al. in [14]. The lane lines are firstly extracted, and then a positioning algorithm based on the lane line map is proposed. In [15], in order to improve the positioning accuracy, the authors use Around View Monitor (AVM) cameras and laser sensors to extract lane line data. Most of the articles mentioned above only fit the curve contours of the lanes, and do not care about the lengths, breakpoints and vertical features. As a result, they provide little assistance in positioning in the longitudinal direction only with lateral position information.

As the road boundary, the curb is one of the most common features in the urban environment. Therefore, curb detection is an important part of environmental perception. A significant number of studies have been conducted on curb detection methods using LIDAR sensors. A scanning 2D LIDAR measurement system is used for fast detection and tracking of road curbs in [16]. The authors in [17] present a curb detection algorithm that uses the scan line of Velodyne LIDAR directly as the processing unit. In [18] and [19], a 3D-LIDAR sensor is used to segment the road and detect its curbs. In [20], curb detection and intersection features are used for localization. A Monte Carlo Localization (MCL) method based on curb features and a prior occupancy grid map is proposed to obtain precise localization information on urban roads. However, the use of curb features alone is not ideal. Therefore, many researchers have combined other features or methods for localization. In [21], the detected road curbs are matched to a prior map, and the data of GPS and IMU are combined to obtain the position of the vehicle. Some authors achieve higher positioning accuracy by combining two or more features. In [22], Choi proposes to use a hybrid grid map and a feature map to obtain location information using LIDAR. In [23], the multilayer LIDAR sensor is used to detect features of road markings and curbs, and then MCL is used for localization.

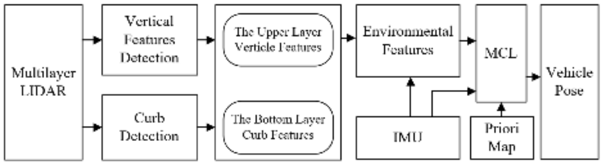

Multilayer LIDAR
Vertical Features Detection
Curb Detection
The Upper Layer Verticle Features
The Bottom Layer Curb Features
Environmental Features
IMU
MCL
Priori Map
Vehicle Pose

Fig. 1. The framework of the proposed method.

In summary, the previous methods have considered either fundamental ground features such as curbs and lane markings, or vertical features such as building outlines. To enhanced the robustness and positioning accuracy, this paper uses two layers of features stored by grid maps to represent the environment: one is a bottom layer composed of ground curb features, and the other is an upper layer composed of a 2D point cloud with vertical features. The framework of the proposed positioning system is shown in Fig. 1. Firstly, the multilayer LIDAR is used to detect the curb and vertical features, and then an IMU is used to fuse multi-frame data to obtain more complete environmental features. Finally, an aprior map is combined with MCL to obtain the estimate of the vehicle position.

The main contributions of the method are three-fold:

1. A new high-precision localization method is proposed based on the combination of curb features and 2D vertical features. To make better use of point cloud data, we propose the double sampling of point cloud from the fused multi-frame data, which can not only reduce noise but also extract high-precision curb contours and vertical features.

2. A curb detection method based on a collision model is proposed. The method can obtain complete curb surface data from each layer of the multilayer LIDAR data.

3. A novel method of extracting vertical features is put forward to retain 3D features and reduce computation by projection. Due to the use of multilayer LIDAR, the 2D point cloud with vertical features contains more environmental information than a conventional 2D map does.

The remainder of this paper is organized as follows. Section II introduces the methods of feature extraction. Section III describes the construction process of a prior map and and the MCL method based on road features. Section IV presents the experimental results and analysis. Finally, Section V summarizes the contributions of this paper.

II. Feature Detection

A. Curb Detection

Curbs are chosen as features to construct a prior map because they are among the most salient features of the road environment, and are insensitive to factors such as weather or light. In our study, the LIDAR sensor is mounted on the top of the intelligent vehicle. The raw point cloud data is obtained in 3D polar coordinates (ρ, θ, γ), where ρ denotes the Euclidean distance of the sensor from a point, and γ and θ are the vertical and horizontal angles, respectively, relative to the reference frame of the sensor itself. These coordinates can be converted into Cartesian coordinates (x, y, z) using the following equations:

$$\begin{cases} x = \rho\, cos(\gamma)\, sin(\theta), \\ y = \rho\, cos(\gamma)\, cos(\theta), \\ z = \rho\, sin(\gamma), \end{cases} \tag{1}$$

$$[x, y, z]^T = [x, y, z + h_s]^T, \tag{2}$$

where x, y, and z represent the points in the sensor coordinate system. h_s is the height of the sensor from the ground. From equations (1) and (2), the laser scanning point set P_l that is to be processed is selected as follows:

$$P_l = \{p_{i,j} | p_{i,j}(x_{i,j}, y_{i,j}, z_{i,j}), i = 1, ..., N, j = 1, ..., N_i\}, \tag{3}$$

where $p_{i,j}$ is the j-th point on the i-th line, N is the number of laser line used and N_i is the number of points in the i-th line point cloud.

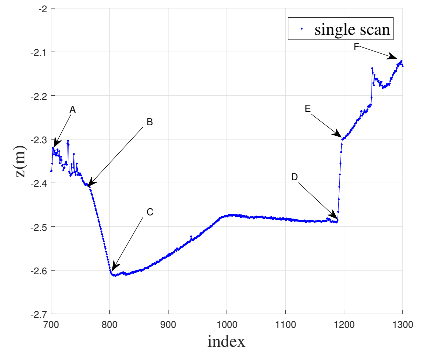


Fig. 2. Raw z-coordinate data.

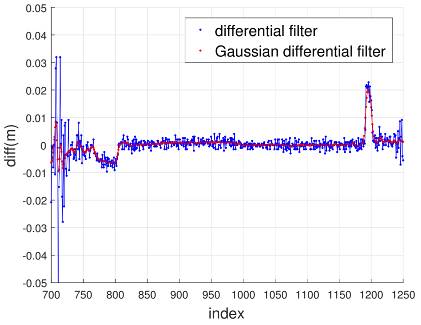


Fig. 3. Filtered data.

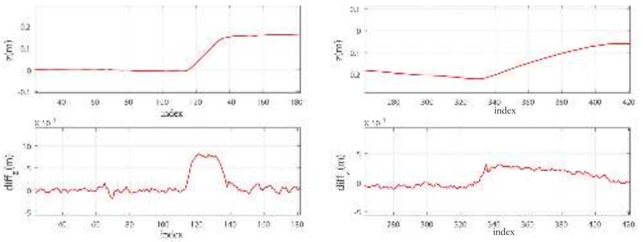
The curb heights are typically uniform in urban areas. When the curb is too small to detect, it is just neglected. When the curb is more than 20 cm, it is treated as an obstacle, like a building, a tree, or a utility pole. Further explanation will be provided in the following parts. Most commonly, the curb 10 to 20 cm higher than the road [25]. However, the height can change dramatically in the vertical direction at the junction of the road surface and the curbs. More details can be seen in Fig. 2. The curb is detected from these spatial features. The proposed curb detection method can be summarized as the following two steps:

1) Data Preprocessing: To reduce noise, a Gaussian filter is applied to the point cloud. Since the height changes the most significantly, only the z coordinate is processed using the filter. The laser scanning point set $z = \{z_i, i = 1, 2, ..., N\}$ for each line is considered separately. The Gaussian filter is expressed as follows:

$$z_f(i) = \sum_{j=0}^{2N_r} z(i - N_r + j)u(j), \tag{4}$$

where $z(i)$ denotes the raw z coordinate of a point, $z_f(i)$ represents the filter data, $u(j)$ denotes the Gaussian weight, and N_r is the blur radius.

In Fig. 2, line CD represents the road surface; lines BC and DE represent the curbs. BC and DE can change dramatically in the vertical direction. Lines AB and EF stand for the unidentified points that are outside the curb, such as sidewalks. In Fig. 3, we can see that the differential-filtered data contains considerable noise, whereas the Gaussian-differential-filtered data is smooth.



(a) (b)

Fig. 4. Road surface data. (a) Differential-filtered data from the right side of the road. (b) Differential-filtered data from the left side of the road.

2) Curb Detecting: Because each laser emission angle is different, the data resolution varies. As the vehicle is travelling on the right side of the road, it is closer to the right curb than to the left. Thus, the number of sampling points at the right side is less than on the left. It can also be observed in Fig. 2 that the points in line BC is denser than those in line ED. Considering about the same height, points of the right edge of the road varies more rapidly, while the height of the left edge changes more slowly. These behaviours can be seen in Fig. 4, where the horizontal axis also implies the number of points. The filtered maximum response values for the left and right curbs are different. Therefore, it is difficult to detect complete edge features using the same parameters on both sides.

However, because the curb height is constant and within a certain range, these characteristics can be used to obtain better detection results. Under the most common circumstances, vehicles travel on the structured roads. Assuming that the road surface is flat, the response of the filter will always remain at a small value, as shown in Fig. 3. Because the filtering responses for the road surface and the curb differ by an order of magnitude, the road surface can be easily detected by using a fixed threshold. Therefore, comprehensive consideration of the road surface boundaries and curb heights can make it easy to solve this problem. This paper presents a curb detection method based on road surface characteristics and curb height, and proposes an algorithm based on collision detection. The process of curb detection is divided into four steps, described in detail as follows.

a) Bottom boundary points. As shown in Fig. 3, the filtering responses for the road surface and the curb differ by an order of magnitude. Δz is the height value after point cloud filtering. The first point where Δz is greater than the threshold is the junction of the road surface and the curb. By analysing the collected data, we can easily distinguish the data corresponding to the road surface from that corresponding to the curb based on a fixed threshold δ_p, instead of using a complex method to calculate δ_z.

$$\Delta z > \delta_p. \tag{5}$$

b) Plane intersection points. To detect the top boundary points, we propose a collision detection model, as shown in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIV.2020.3003699, IEEE Transactions on Intelligent Vehicles

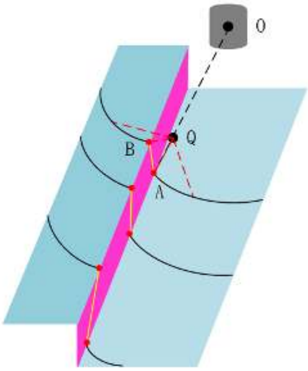


Fig. 5. Collision detection model.

Fig. 5. The system is modelled as follows. Suppose that a sphere with radius r is launched from the centre point O of the laser sensor to point A at the bottom edge of the road, the first point of contact with the sphere is point B at the top boundary of the road edge.

The sphere moves along the ray OA, calculating the distance from the point cloud to the centre of the sphere at each step. If the distance is less than the radius of the sphere, a collision is considered to occur. Note that the computational burden of this approach is very high. Therefore, an approximate calculation method is proposed.

In Fig. 5, point B that needs to be calculated, is at the top edge of the road at a certain height above the road surface. Ray OA generates an intersection point Q with the horizontal plane whose height is h_c. h_c is the upper limit value of the curb height, with which we can guarantee the following analysis. Starting at point A, the point cloud approaches increasingly closer to point Q, and obtains the minimum value at point B. Then, it stats moving away from point Q. In addition, with the different laser beam angles, the planar distance of the point cloud AB segment along the roadway differs, resulting in different effective lengths of the roadway. Therefore, accurate acquisition of the top edge points can expand the coverage range of the roadside data. From the similar triangle, the coordinates of point Q can be found as follows:

$$\begin{cases} x_q = x_a \dfrac{(h_s - z_a - h_c)}{h_s - z_a}, \\ y_q = y_a \dfrac{(h_s - z_a - h_c)}{h_s - z_a}, \\ z_q = z_a + h_c, \end{cases} \tag{6}$$

where (x_a, y_a, z_a) denotes point A, (x_q, y_q, z_q) denotes point Q, and h_s indicates the height of the curb.

c) Top boundary points. Now that the plane intersection point Q has been obtained, the search radius r needs to be set. Since the length of l_{qa} must be greater than that ofl_{qb}, and l_{qb} is the shortest distance, the radius r is set to l_{qa}. The nearest point among all points within the radius r from point Q is the top boundary point B. The point B is calculated via the following function:

$$B(x_b, y_b, z_b) = min\ dist(p, Q), p \in \mathbf{M}, \tag{7}$$

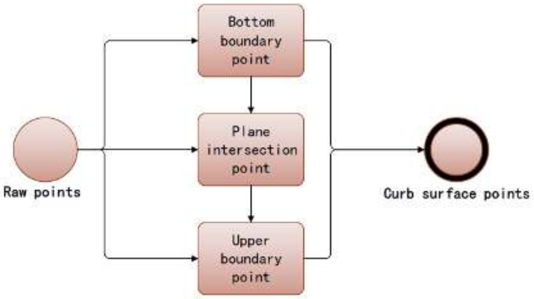


Fig. 6. Curb detection schematic diagram.

where the $dist(p, Q)$ function is used to calculate the distance between candidate point p and point Q. The point set $\mathbf{M}$ represents the points within the radius r.

d) Curb surface points. Once the top and bottom boundary points of the road edge surface are obtained, all points between these points need to be placed into the point set of the road edge surface.

This method can be used to obtain all curb surface points. It uses only two parameters for detection: the road surface threshold δ_p and the curb height h_c. Moreover, the same parameters are used for laser beams with different emission angles. The schematic diagram of the algorithm is presented in Fig. 6.

B. Detection of the 2D Point Cloud with Vertical Features

The road edge features can be obtained via the curb detection algorithm introduced above, so as to obtain the high-precision lateral positioning results. To improve the longitudinal positioning performance, we need to take more environmental features into consideration. Thus, the selection of the data for positioning is important.

One of the challenges of the LIDAR positioning methods based on map matching is the size of the apriori map. Large-scale outdoor 3D point cloud maps are so big that storing them in vehicles poses a problem. Compared to that of a 3D point cloud map, the amount of data contained in a 2D map is within acceptable limits. However, due to the small amount of data collected by the 2D laser and the high vulnerability of the vehicle, the uncertainty of the conventional 2D map is high [19].

Environmental features such as buildings, trees, and utility poles are typically used for localization. Thus, the correct detection of these features plays an important role in high-precision positioning. To improve the robustness of localization, we need to extract as many different environmental features as possible. However, extracting each feature separately requires a great deal of computation. In addition, some other features in the environment are not easy to detect. Therefore, the raw 2D point cloud is directly used for localization.

To obtain stable vertical features, the points in a certain vertical space are selected as the positioning features. The point clouds located in the same vertical plane are projected together and displayed as contours of the vertical plane, as

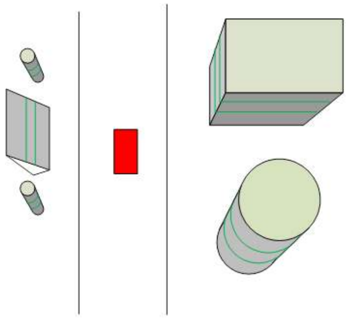


Fig. 7. Projection-based detection method.

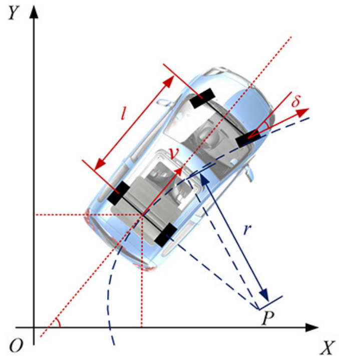


Fig. 8. Vehicle motion model.

shown in Fig. 7. The green lines represent two different horizontal planes. If the projections of the two layers coincide, the plane—the surface of the object—between the two layers is perpendicular to the ground, and can describe the contours of the obstacles. In the experiments, the 3D point cloud data is converted into the 2D map and stored as grid map. We first project each point on the upper layer of the grid and mark them. Then we do the same to the bottom layer. Grids marked twice mean that they overlap and that there are vertical obstacles. The stable vertical features can describe the contours of the obstacles and display the environmental features more directly. However, for non-vertical point clouds, such as slopes, contour information cannot be obtained because the projection points of the two layers do not coincide.

The 2D point cloud is constructed as follows:

$$\mathbf{P}_v = \{p_i \in \mathbf{P}_{v1} \cap \mathbf{P}_{v2}, i = 1, 2, ..., N\}, \tag{8}$$

where $\mathbf{P}_v$ denotes the set of points in the 2D point cloud, and $\mathbf{P}_{v1}$ and $\mathbf{P}_{v2}$ denote the sets of points projected from the two selected planes.

C. Multi-Frame Fusion

To obtain the most complete representation of environmental information, the multi-frame 2D point clouds are transformed into the current vehicle coordinate system. During the movement of the vehicle, the single-frame point cloud data obtained at each position is too sparse to completely extract the roadside and vertical features. Therefore, multi-frame point cloud data collected near the current location need to be fused. In the process of multi-frame fusion, 10 frames are used to fuse vertical points, and 15 frames are used for curb detection. Under the vehicle kinematics assumption shown in Fig. 8, IMU is used to measure the vehicle motion to uniformly transform the multi-frame point cloud data into the current vehicle coordinate system. $u_t = [\upsilon, \omega]^T$ represents the motion at time t, where υ represents the translational velocity and ω denotes the rotational velocity. If both velocities remain constant over a time interval, the vehicle will move in a circle the radius r:

$$r = |\frac{\upsilon}{\omega}|. \tag{9}$$

The position and heading deviations between the last and current frames on the x-y plane are represented by $T = [\Delta x \ \Delta y]$ and $\Delta\theta$, respectively. They are calculated as:

$$\begin{cases} \Delta\theta = \omega\Delta \ t, \\ \Delta x = r(1 - cos(\Delta\theta)), \\ \Delta y = r\sin(\Delta\theta). \end{cases} \tag{10}$$

The 2D point cloud coordinates at time $t-1$ are denoted by Q_{t-1}. The new 2D point cloud coordinates Q_t at time t are calculated as:

$$Q_t = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \left(Q_{t-1} - T\right). \tag{11}$$

With regard to the curb features, the noise in the detected data increases with the fusion of multi-frame data. Thus, a method to remove the noise is needed. Through the analysis of these noise data, we know that the noise is mainly distributed on the roadside, whereas the road interior noise is less. With an increasing amount of multi-frame data, the contours of the road edges become more abundant, making it easier to eliminate noise.

The beam model proposed in [24] is generally used to obtain environmental contours in the field of robotics. However, the beam model can not remove the noise effectively. If the angular resolution is large, the noise can be effectively removed, but the accuracy of the curb contour will be reduced, and vice verse. Therefore, this paper proposes method of curb data extraction and denoising based on double sampling, which can reduce the noise while ensuring the accuracy of the countours. Firstly, a low-resolution beam model is used for rough sampling to remove the noise. Then, a high-resolution beam model is used for fine sampling of the original data to ensure the high contour resolution. The double sampling guarantees both the contour accuracy and denoising precision. The extracted roadside data in the planar coordinates are first converted into polar coordinates: $C' = \{\theta_i, \rho_i\}, i = 1, \ldots, N_c$.

A low angular resolution $\delta_1 = 2\pi/n_s$ is used to remove the noise, where n_s represents the number of curb sampling points.

$$\begin{cases} k = \theta_i/\delta_1, \\ D[k] = \min\{\rho_i, D[k]\}, \end{cases} \tag{12}$$

where D denotes an array of polar coordinate radii represented by the curb data. The elements of the array are initialized to the sensor's maximum perception radius r_m, and k represents the index of the data in the array.

In the process of the second resampling, the noise-removed data is resampled, and a high angular resolution $\delta_2 = 2\pi/n_b'$ is used to improve the accuracy of the contours along the road, where n_b' represents the number of curb sampling points and is greater than n_s.

$$\begin{cases} k = \theta_i/\delta_1, \\ k' = \theta_i/\delta_2, \\ D'\left[k'\right] = \min\left\{\rho_i, D[k], D'\left[k'\right]\right\}, \end{cases} \tag{13}$$

where D' denotes a high-resolution array of polar coordinate radii represented by the curb data. Here, k is the index of a point in the low-resolution curb array, and k' denotes the index of the same point in the high-resolution curb array.

III. Localization

The proposed positioning method is based on a prior map; thus, the apriori map should first be generated from the extracted features. Then, we use the apriori map and the MCL method to obtain the optimal estimate of the vehicle pose. The following sections address the mapping process and the MCL method applied.

A. Occupancy Grid Mapping

The occupancy grid map is built by using feature point information. The position of the vehicle is obtained by a real-time kinematic (RTK) GPS device. $X_t = (x_t, y_t, \theta_t)$ is obtained by converting GNSS coordinates into Cartesian coordinates. The coordinates of the curb data and the 2D point cloud of vertical features are transformed from the sensor coordinate system into the global coordinate system as follows:

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + \begin{bmatrix} \cos(\theta_t) & -\sin(\theta_t) \\ \sin(\theta_t) & \cos(\theta_t) \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix}, \tag{14}$$

where $[x_s \ y_s]^T$ denotes the coordinates in the sensor coordinate system, and $[x_g \ y_g]^T$ denotes the coordinates in the global coordinate system. For each cell, the state c is calculated as:

$$c = \begin{cases} 1, & \text{if } n > N_t, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where n denotes the number of points falling into the cell, and N_t denotes a threshold. Only when n is greater than this threshold is the state of the cell considered to be occupied. This method can reduce the noise of the apriori map.

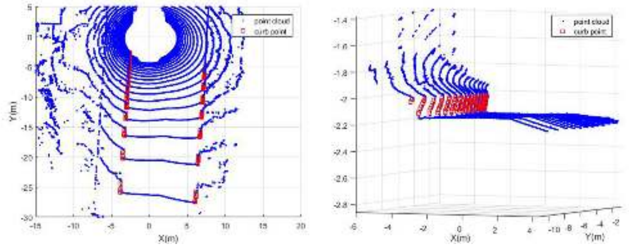
B. Monte Carlo Localization

The MCL method is a Bayesian filtering pose estimation method implemented based on the Monte Carlo method. Researchers have conducted in-depth researches on this method [24]. The internal state of the vehicle is modelled as a discrete-time Markov process $X_t = (x_t, y_t, \theta_t) \in \mathbb{R}^3$, $t \in \mathbb{N}$.

The position of the vehicle is represented by a set of particles. The particles are updated as the sensor measurements are updated, and the number of particles in the auxiliary particle set is determined in real time in accordance with the variance of the positioning results. The belief $bel(X_t)$ represents the probability distribution of the particles, each of which has a corresponding weight w_t^i:

$$bel(X_t) \approx \{X_t^i, w_t^i\}_{i=1}^{N_p}, \tag{16}$$

where N_p represents the number of particles, and t indicates the time point.



(a)

(b)

Fig. 9. Curb detection results. (a) 2D point cloud. (b) 3D point cloud

With the update of the sensor measurement data, the set of particles S_t is updated with a set of auxiliary particles $\widetilde{S}_t$. The iterative update of the particle set consists of the following four steps.

Step 1: In the motion update step, the poses X_t^i of the particles in $\widetilde{S}_t$ are predicted. The pose of each particle is calculated using the probability distribution $p(X_t|X_{t-1}, u_{t-1})$, where X_t represents the predicted particle distribution, and $u_{t-1} = [v \ w]^T$ denotes the motion of the vehicle. In other words, the current pose is predicted using the pose and vehicle motion information from the previous moment.

Step 2: In the measurement update step, the weight w_t^i corresponding to each particle is calculated based on the sensor measurement data and the apriori map. The observation model [24] is modelled as a likelihood field model as follows:

$$w_t^i = p(Z_t|X_t^i, m) = \varepsilon_\sigma(dist), \tag{17}$$

$$dist = min\{\|X_t^i - O\|_2 \mid O \in m\}, \tag{18}$$

where $\varepsilon_\sigma \sim \mathcal{N}(0, \sigma^2)$, X_t^i and m correspond to the i-th particle's pose and the occupancy grid map, respectively, and Z_t denotes the measurement value of the sensor. That is, the weights of the predicted particles in $\widetilde{S}_t$ are updated based on the measurement values.

Step 3: The vehicle position $\hat{X}_t$ is considered to be represented by the particle with the highest weight.

Step 4: Using the method of Sampling Importance Resampling [24], N_p particles are selected from $\widetilde{S}_t$. The probability that a particle is selected is proportional to the weight of the particle. In addition, to improve the diversity of the particles, samples are taken from the normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$, where μ represents the mean of the particles, and σ^2 denotes the variance of the particles. Then, the selected particles replace the current S_t set, and w_t^i is set to $1/N_p$.

IV. Experimental Results and Analysis

A. System Overview

Our intelligent vehicle is equipped with an Oxford Inertial+2 and NovAtel GPS receiver running at 100 Hz. This

RTK GPS device provides the ground truth for evaluating the error of the localization method. This method is implemented in C++. The ADLINK Industrial Personal Computer used as the controller has a 2.3 GHz Intel Core i7-3610QE CPU and 16 GB of RAM. The experimental tests were conducted on our campus.

A Velodyne HDL-32E LIDAR sensor is used to extract environmental features. This LIDAR sensor emits 32 beams of light to directly collect 3D point cloud information in the environment. The LIDAR sensor, small in size and light in weight, provides a 41.3° vertical view and a 360° horizontal view. The sensor has a measurement range of up to 70 meters and a resolution of 2 cm. Due to the sparse data collected by the peripheral lasers, 22 internal lasers are used.

B. Feature Extraction

Feature extraction is a key step in the process of localization. Each frame of data is processed in real-time to achieve a higher positioning response.

1) Curb Detection: The single-frame curb detection results are shown in Fig. 9. The red squares indicate curb points, and the normal point clouds are shown in blue. The parameter δ_p for detecting the road surface is set to 0.015, and the curb height h_c is set to 0.2m. As shown in Fig. 9(b), all the edge points on the curb surfaces have been detected. Thus, the task of detecting the complete curb surface with only two parameters is successfully realized.

To validate the multi-frame curb detection method, four scenarios are considered: a simple straight road, a complex straight, an intersection, and a curve. The results are shown in Fig. 10, where the red dots are the curb points detected in multiple frames. Note that the simple straight and the curve, shown in Fig. 10(a) and (d) respectively, are considered to be simple road shapes, whereas the other two road shapes shown in Fig. 10(b) and (c) are complex. The sampling resolution for noise rejection is set to 2°, and the resolution of the extracted trailing edge contours is 0.2°. Through double sampling,the noise caused by false detection points along the road can be eliminated without reducing the resolution of the curb contours.

2) Detection of the 2D Point Cloud of Vertical Features: An example of a multi-frame 2D point cloud is shown in Fig. 11, where the red dots represent the fusion of the multi-frame 2D point cloud and the IMU data and the blue points represent the map data. By detecting the data from the two planes, more stable vertical feature points are obtained. As shown in Fig. 11, the outlines of the trees and buildings are clearly visible. This environmental information can be obtained without complicated calculations, which is beneficial to the real-time performance of the positioning process. In addition, by fusing multi-frame data, more abundant environmental features can be obtained.

C. Vehicle Localization

The RTK GPS device is utilized to collect a series of poses along the vehicle trajectory, which is 1.34 km. Additionally, the data recorded by the LIDAR sensor is associated with the pose data. The proposed detection algorithm is used to extract the curb points and vertical feature points. Then, the extracted feature points are used to generate the apriori map. The map is constructed with sequences of records that are different from those used for localization. Two factors should be considered in determining the resolution of the apriori map. On the one hand, the apriori map should be able to accurately represent the characteristics of the environment, and on the other hand, the size of the map should be within the acceptable range (not too large). Considering these factors, the resolution of the grid is set to 0.20 m.

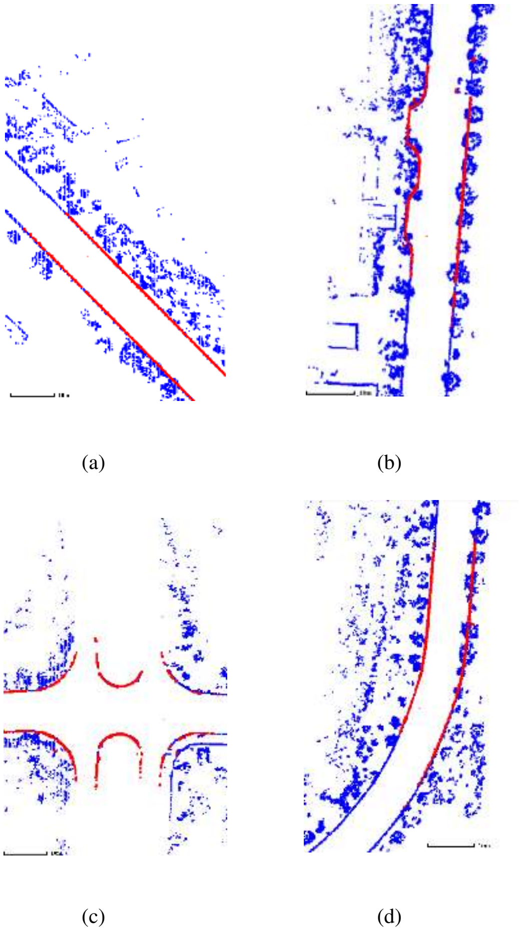


Fig. 10. Fusion of multi-frame curb data. (a) Simple straight. (b) Complex straight. (c) Intersection. (d) Curve.

The minimum number of particles is 200, and the maximum number is 1000. The number of sampling particles is determined in real time in accordance with the variance of the positioning result. The initial distribution of particles is generated from the low-precision positioning data acquired by the GNSS device.

Fig. 12 depicts the localization results of the test loop. The size of the circle implies the error, which can qualitatively represent the magnitude of the error, rather than a specific numerical value. The purple dots indicate the road boundary. The blue dots denote the 2D point clouds of vertical features. The red circles represent the lateral error, and the green

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIV.2020.3003699, IEEE Transactions on Intelligent Vehicles

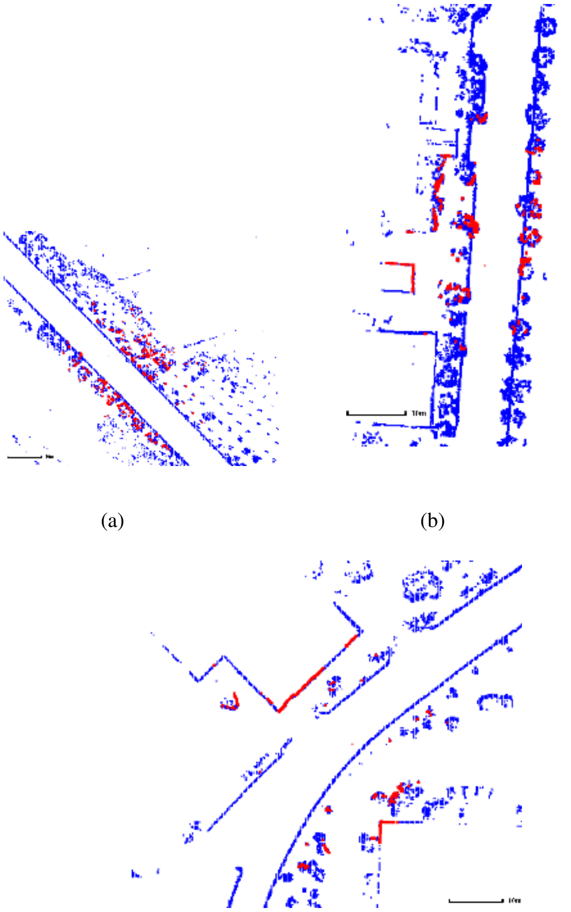
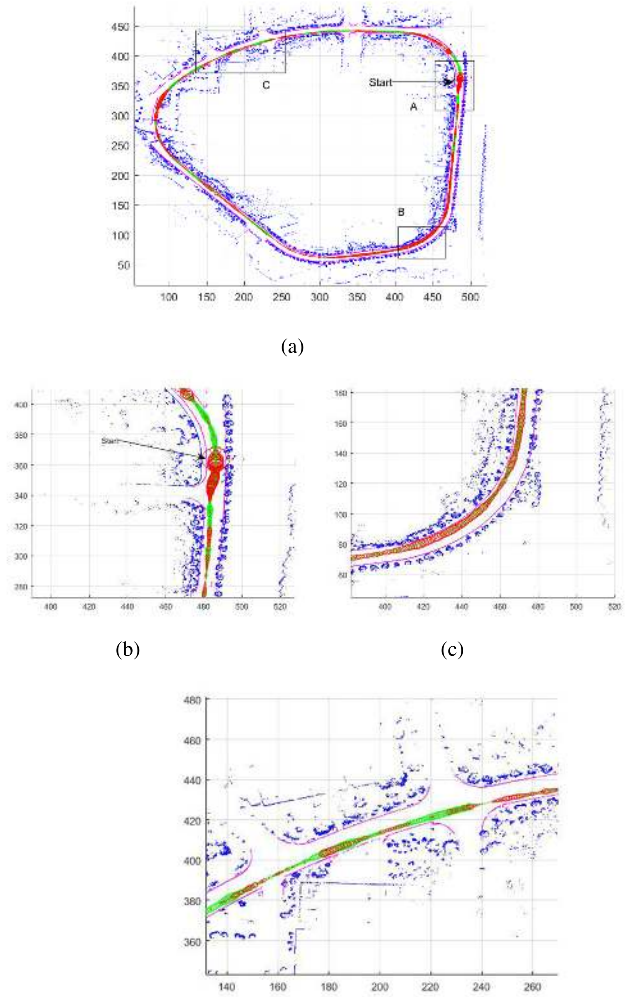

(a) (b) (c)

Fig. 11. Fusion of multi-frame 2D point cloud data. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.



(a)

(b) (c)

(d)

Fig. 12. Localization error. (m) (a) Trajectory with error. (b) Position A. (c) Position B. (d) Position C.

circles represent the longitudinal error. Fig. 12(b), (c), and (d) correspond to positions A, B, and C. As shown in Fig. 12(b), since the localization process is initialized from low-precision GNSS data at the starting position, the localization error near that position is large. When the vehicle moves a few metres, the particles converge, and the error becomes smaller. As seen from Fig. 12(c) and (d), the error of position B is greater than that of position C. This finding can be attributed to the presence of buildings at position C, which significantly improves the positioning accuracy. The results of positioning error can be qualitatively analysed from Fig. 12. Next, specific data will be used to quantitatively analyse these results. Fig. 13(a) illustrates the lateral, longitudinal and heading error relative to the ground truth (RTK GPS). Throughout the test, the lateral and longitudinal estimation error of particle convergence always remains at a small values, indicating that the positional reliability of the particles is very high.

Fig. 13(b) shows the cumulative distribution functions (CDFs) of the lateral and longitudinal errors. As shown in Fig. 13(b), the lateral and longitudinal errors are approximately 0.16 m and 0.22 m, respectively, with a confidence level of 95%. After the particles converge, the maximum lateral error is 0.20 m, and the maximum longitudinal error was 0.30 m. And the error distribution histograms are shown in Fig. 14.

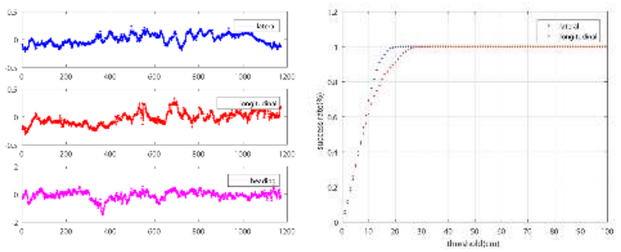
TABLE I: Comparison when disenabling vertical features or curbs

Type	Lateral (m)	Longitudinal (m)	Heading (°)
Ours	0.077 ± 0.090	0.091 ± 0.113	0.219 ± 0.301
disenabling curbs	0.093 ± 0.108	0.087 ± 0.108	0.232 ± 0.314
disenabling vertical features	0.086 ± 0.104	0.200 ± 0.266	0.335 ±0.409

TABLE II: Comparison when disenabling multi-frame fusion or double sampling.

Type	Lateral (m)	Longitudinal (m)	Heading (°)
Ours	0.077 ± 0.090	0.091 ± 0.113	0.219 ± 0.301
disenabling multi-frame fusion	0.092 ± 0.109	0.135 ± 0.116	0.300 ± 0.332
disenabling double sampling	0.104 ± 0.130	0.177 ± 0.200	0.323 ± 0.382

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIV.2020.3003699, IEEE Transactions on Intelligent Vehicles



(a)

(b)

Fig. 13. Error data. (a) Lateral, longitudinal and heading error plots. The horizontal axis represents the number of particles and the vertical axis represents the degrees. (b) CDFs of the lateral and longitudinal errors.

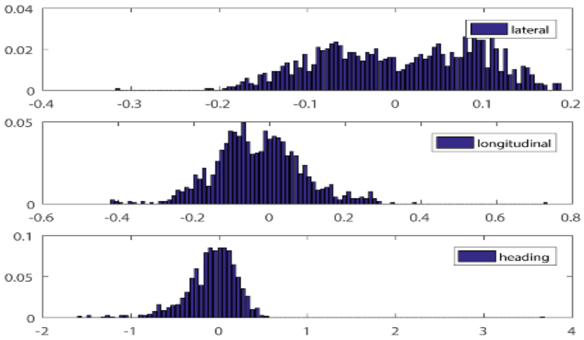
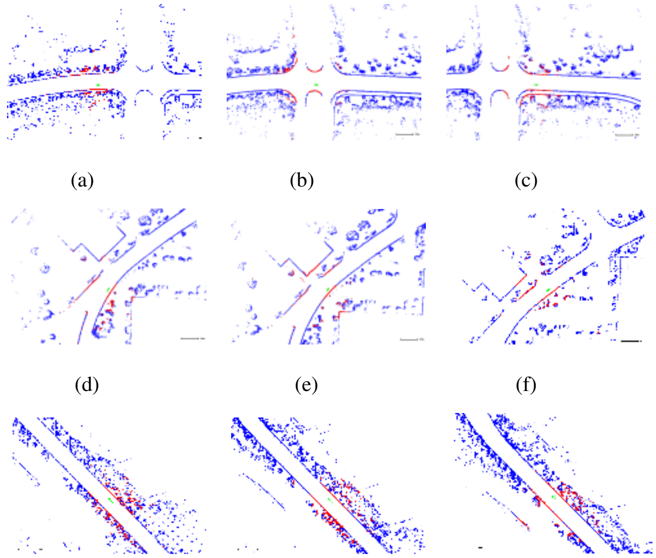


Fig. 14. Lateral, longitudinal and heading error histograms.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 15. Typical particle behaviours.

When comparing with other state-of-the-art methods, the data used in these papers are not open-sourced, making the comparison of our method using their data difficult. And the algorithm codes in these papers are not open-sourced, either. Furthermore, different sensor installation methods and the selection of data acuisition will affect the generation of the map, making their methods applied on our data unconvincing. Therefore, we pay more attention to improve the localization precision of our method. Table I shows the average lateral, longitudinal and heading errors, as well as the root mean square errors (RMSEs) resulting from different feature detection methods. We have compared three situations, including using only curb detection, using only vertical features and using both. It can be seen that the contribution of curb is mainly reflected in lateral positioning, and the vertical feature plays a major role in the longitudinal positioning and heading estimation. The comparison of different sampling methods is given in Table II. It can be seen that multi-frame fusion and double sampling improve the positioning results. We have compared three situations, including disenabling multi-frame fusion, disenabling double sampling and using both.

Fig. 15 shows the typical particle behaviours in three scenarios. The green dots represent particles; the red dots represent detected curb features and point clouds of vertical features; and the blue dots represent map data. For the scenarios involving intersections and buildings, the data collected by the sensors has abundant features. Thus, the distribution of the particles is relatively concentrated. In the case of a long straight road with only a few features, the particles appear scattered and banded. In addition, the average processing period of the method is about 40 ms, which is suitable for the real-time operation of autonomous vehicles.

V. Conclusions

This paper proposes an intelligent vehicle self-localization scheme based on double-layer features and multilayer LIDAR data. The bottom-layer features consist of curb features, which are the most prominent features of urban road. The collision-based curb detection method proposed in this paper needs only a few parameters to obtain the complete curb surface information. The upper-layer features are generated from the 2D point clouds of vertical information, which capture buildings, trees, and so on. These raw 2D point clouds can accurately and completely express the environmental information. The fusion of multi-frame data not only increases the number of feature points, but also reduces the noise in the road curb data through contour extraction algorithm. Then, the map is matched with curbs and vertical features detected with multiple frames, using the Monte Carlo localization algorithm to achieve accurate localization.

Compared with 3D point cloud map, the apriori map that stores the double-layer features is not only extremely small in size but also provides sufficient features to realize precise self-localization. The 2D point clouds with vertical features collected from multilayer LIDAR data contain more environmental features than the conventional 2D point clouds do. Moreover, the positioning accuracy is improved by introducing curb features. In future work, the camera and LIDAR sensor data will be combined to improve the feature detection accuracy and positioning accuracy.

References

[1] J. Knaup and K. Homeier, ”RoadGraph - Graph based environmental modelling and function independent situation analysis for driver assistance systems,” in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Sept. 2010, pp. 428-432.

[2] T. Kos, I. Markezic and J. Pokrajcic, ”Effects of multipath reception on GPS positioning performance,” in *Pro. ELMAR-2010*, Sept. 2010, pp. 399-402.

[3] H. Lategahn and C. Stiller, ”Vision-Only Localization,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246-1257, June 2014.

[4] S. Panev, F. Vicente, F. De la Torre and V. Prinet, ”Road Curb Detection and Localization With Monocular Forward-View Vehicle Camera,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1-17, Nov. 2018.

[5] J. Jeong, Y. Cho and A. Kim, ”Road-SLAM : Road marking based SLAM with lane-level accuracy,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2017, pp. 1736-1473.

[6] M. Schreiber, C. Knöppel and U. Franke, ”LaneLoc: Lane marking based localization using highly accurate maps,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, June 2013, pp. 449-454.

[7] F. Poggenhans, N. O. Salscheider and C. Stiller, ”Precise Localization in High-Definition Road Maps for Urban Regions,” in *Proc. IEEE/RSJ Int. Conf. Int. Robots Syst. (IROS)*, Oct. 2018, pp. 2167-2174.

[8] X. Du and K. K. Tan, ”Comprehensive and Practical Vision System for Self-Driving Vehicle Lane-Level Localization,” *IEEE Trans. Image Processing*, vol. 25, no. 5, pp. 2075-2088, May 2016.

[9] L. Wei, C. Cappelle and Y. Ruichek, ”Horizontal/vertical LRFs and GIS maps aided vehicle localization in urban environment,” in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Jan. 2013, pp. 809-814.

[10] J. Wiest, H. Deusch, D. Nuss, S. Reuter, M. Fritzsche and K. Dietmayer, ”Localization based on region descriptors in grid maps,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2014, pp. 793-799.

[11] Underwood J P , Jagbrant G , Nieto J I , et al. Lidar-Based Tree Recognition and Platform Localization in Orchards. *Journal of Field Robotics*, 2015, 32(8):1056-1074.

[12] E. Javanmardi, M. Javanmardi, Y. Gu and S. Kamijo, ”Autonomous vehicle self-localization based on multilayer 2D vector map and multichannel LiDAR,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, June 2017, pp. 437-442.

[13] H. Guan, J. Li, Y. Yu, M. Chapman and C. Wang, ”Automated Road Information Extraction From Mobile Laser Scanning Data,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 194-205, Feb. 2015.

[14] D. Kim, T. Chung and K. Yi, ”Lane map building and localization for automated driving using 2D laser rangefinder,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, June 2015, pp. 680-685.

[15] H. Lee, S. Kim, S. Park, Y. Jeong, H. Lee and K. Yi, ”AVM / LiDAR sensor based lane marking detection method for automated driving on complex urban roads,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2017, pp. 1434-1439.

[16] W. S. Wijesoma, K. R. S. Kodagoda and A. P. Balasuriya, ”Road-boundary detection and tracking using ladar sensing,” *IEEE Trans. Robot. Automat.,*, vol. 20, no. 3, pp. 456-464, June 2004.

[17] T. Chen, B. Dai, D. Liu, J. Song and Z. Liu, ”Velodyne-based curb detection up to 50 meters away,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, June 2015, pp. 241-248.

[18] S. Xu, R. Wang and H. Zheng, ”Road Curb Extraction From Mobile LiDAR Point Clouds,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 996-1009, Feb. 2017.

[19] Y. Zhang, J. Wang, X. Wang and J. M. Dolan, ”Road-Segmentation-Based Curb Detection Method for Self-Driving via a 3D-LiDAR Sensor,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3981-3991, Dec. 2018.

[20] B. Qin, Z. J. Chong, T. Bandyopadhyay, M. H. Ang, E. Frazzoli and D. Rus, ”Curb-intersection feature based Monte Carlo Localization on urban roads,” in *Proc. IEEE ICRA*, May 2012, pp. 2640-2646.

[21] L. Wang, Y. Zhang, J. Wang, ”Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 276C281, 2017.

[22] J. Choi, ”Hybrid map-based SLAM using a Velodyne laser scanner,” in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2014, pp. 3082-3087.

[23] A. Y. Hata and D. F. Wolf, ”Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420-429, Feb. 2016.

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[25] *Code for Design of Urban Road Engineering in China, CJJ37-2012,* Ministry of Housing and Urban-Rural Development of the People’s Republic of China, Beijing, China, Jan. 2012.

Zhuping Wang received the B.Eng. and M.Eng. degrees from the Department of Automatic Control, Northwestern Polytechnic University, China, in 1994 and 1997, respectively; and the Ph.D. degree from National University of Singapore in 2003. Currently, she is a Professor at the College of Electronics and Information Engineering, Tongji University, Shanghai, China. Her research interests include intelligent control of robotic systems, self-driving vehicles, and multi-agent systems.

Jian Fang received the B.Eng. degree in measuring and controlling technology and instrument from Nanjing Normal University, Nanjing, China, in 2016 and the M.Eng. degree in control science and engineering from Tongji University, Shanghai, China, in 2019. His research interests include environment perception and vehicle localization in area of autonomous vehicles.

Xinke Dai received the B.Eng. degree in naval architecture and marine engineering from Zhejiang University, Hangzhou, China, in 2018. She is currently studying for the M.Sc. degree in control theory and control engineering at Tongji University, Shanghai, China. Her current research interests include vision-based SLAM and monocular depth estimation.

Hao Zhang received the B.Sc. degree in automatic control from Wuhan University of Technology, Wuhan, China, in 2001 and the Ph.D. degree in control theory and control engineering from Huazhong University of Science and Technology, Wuhan, China, in 2007. Currently, she is a Professor with the School of Electronics and Information Engineering, Tongji University, Shanghai, China. Her research interests include network-based control systems, multi-agent systems, and complex networks.

Ljubo Vlacic received the Grad. Diploma degree in engineering and the M.Phil. and Ph.D. degrees in electrical engineering (control) from the University of Sarajevo, Sarajevo, Bosnia and Herzegovina, in 1973, 1976, and 1986, respectively. He is currently a Professor and the Director of the Intelligent Control Systems Laboratory, Institute of Integrated and Intelligent Systems, Griffith University, Brisbane, QLD, Australia. Prof. Vlacic was a recipient of the 2003 Queensland Professional Engineer of the Year from the Queensland Division of the Institution of Engineers Australia in recognition of his contributions to the advancement of control systems and their applications, the 2004 Sir Lionel Hooke Award from the Australian Council of the Institution of Engineering and Technology, the 2004 IEEE Achievement Medal (by the IEEE Knowledge Board), and a number of appreciation awards for notable services.