Research Article

Lane-level localization system using surround-view cameras adaptive to different driving conditions

Tianyi Li¹, Yuhan Qian², Arnaud de La Fortelle³, Ching-Yao Chan⁴ and Chunxiang Wang¹

Abstract
This article presents a lane-level localization system adaptive to different driving conditions, such as occlusions, complicated road structures, and lane-changing maneuvers. The system uses surround-view cameras, other low-cost sensors, and a lane-level road map which suits for mass deployment. A map-matching localizer is proposed to estimate the probabilistic lateral position. It consists of a sub-map extraction module, a perceptual model, and a matching model. A probabilistic lateral road feature is devised as a sub-map without limitations of road structures. The perceptual model is a deep learning network that processes raw images from surround-view cameras to extract a local probabilistic lateral road feature. Unlike conventional deep-learning-based methods, the perceptual model is trained by auto-generated labels from the lane-level map to reduce manual effort. The matching model computes the correlation between the sub-map and the local probabilistic lateral road feature to output the probabilistic lateral estimation. A particle-filter-based framework is developed to fuse the output of map-matching localizer with the measurements from wheel speed sensors and an inertial measurement unit. Experimental results demonstrate that the proposed system provides the localization results with submeter accuracy in different driving conditions.

Keywords
Lane-level localization, deep learning, surround-view cameras, autonomous vehicle, vision-based localization

Date received: 20 April 2019; accepted: 11 March 2020

Topic: Mobile Robots and Multi-Robot Systems
Topic Editor: Lino Marques
Associate Editor: Oscar Mozos

Introduction
Vehicle self-localization is one of the essential functions in advanced driver-assistance systems. The most accessible localization technique is the global navigation satellite system (GNSS). However, GNSS cannot always have sub-meter accuracy when the signal is blocked by surrounding buildings. Much effort has been made to increase the localization accuracy, including map-matching methods based on light detection and ranging (LiDAR) or cameras. LiDAR-based approaches¹⁻³ are able to achieve centimeter-level accuracy. Nevertheless, LiDAR is not feasible for mass deployment because of its high price currently. As cameras are relatively cheaper, and images

¹ Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China
² Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China
³ Centre for Robotics, MINES ParisTech, PSL Research University, Paris, France
⁴ California PATH (Partners for Advanced Transportation Technology), University of California, Berkeley, Richmond, CA, USA

Corresponding author:
Chunxiang Wang, Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China.
Email: wangcx@sjtu.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
contain abundant information, vision-based methods have gained a lot of attention.\textsuperscript{4}

Despite decades of research, most vision-based methods can hardly perform self-localization adaptive to different driving conditions without much manual effort. Image-feature-based methods\textsuperscript{5} have poor performance during heavy traffic condition. Semantic-feature-based methods (such as lane, ground, and curb detection) gain much attention recent years. Rule-based methods\textsuperscript{6,7} always have high accuracy for some specific driving conditions, but a perfect rule is difficult to design to suit for all kinds of real driving conditions, especially when many arrows exist, lanes are splitted, or vehicle is changing lanes. Deep-learning-based methods\textsuperscript{8,9} extract features in diverse driving conditions. However, manual labeling in a large data set is required for most deep-learning-based methods. It costs much and limits the application of deep-learning-based methods on a large scale. In this article, we develop a cost-effective localization system adaptive to different driving conditions (see Figure 1).

In the proposed system, a map-matching localizer (MML) is proposed to estimate the probabilistic lateral position based on raw images from surround-view cameras and the lane-level road map. MML contains a sub-map extraction module, a perceptual model, and a map-matching model. The proposed system fuses the information from MML and low-cost sensors including inertial measurement unit (IMU), wheel speed sensors (WSS), and a GNSS receiver to provide the lane-level localization. The key contributions of our article are the following:

- We devise a probabilistic lateral road feature (PLRF) as a sub-map which contains the lateral information of the lane-level map. PLRF is a general feature which requires no assumptions of road structures.
- We design a deep learning network as the perceptual model to process raw images from surround-view cameras. Unlike conventional deep-learning-based methods, the perceptual model is trained by auto-generated labels from the map information to reduce manual effort.
- We develop a particle-filter-based framework to fuse the results of the MML and the measurements from low-cost sensors. This framework provides robust lane-level localization results in different driving conditions.

**Related work**

Related work about map-based localization systems can be divided into three parts: the choice of maps, detection, and localization.

**Map**

Many different types of digital maps have been used for localization, including the roadway map,\textsuperscript{10} the 3D dense map,\textsuperscript{11} the feature map,\textsuperscript{12} and the lane-level road map.\textsuperscript{13} The roadway map, for example, OpenStreetMap, models roads as polygons without the information of width, so it is not sufficient for a localization method to achieve the lane-level accuracy. The 3D dense map contains many points that can construct the surface of surrounding objects with centimeter-level accuracy, but it requires too much storage when the area is large. The feature map is designed including features related to detection results. Users often have to self-generate them in advance, which is infeasible on a large scale. The lane-level road map contains the positions of lane lines, centerlines, and curbs, which does not require much storage and has lane-level information. Besides, many researchers focus on auto-generation of lane-level road maps.\textsuperscript{14,15} Thus, we design our localization system using the lane-level road map.

A sub-map should be extracted for efficiently matching with observations. Typical sub-maps of lane-level road maps fit lane lines as splines,\textsuperscript{16} clothoids,\textsuperscript{17} or cubic polynomials,\textsuperscript{18} leading to inevitable errors when the shapes of lanes are irregular. In this article, we propose a PLRF that represent the distribution of lateral features without limitations of road structures.
Detection

Vision-based map-matching methods exploit visual features from images, for example, SIFT, SURF, DIRD, ORB, CNN features, and so on. However, these features suffer from heavy traffic conditions, since surrounding vehicles also have these local features which are noises for map matching. To provide robust localization, some researchers detect semantic features and then match them with the lane-level road map. Brown and Brennan presented mapped lane features for lateral localization using a mono camera. The mono camera is frequently blocked by front vehicles. To avoid the influence of occlusions, some researchers utilize surround-view cameras: four fish-eye cameras mounted around the vehicle. Raw images from surround-view cameras have the strong distortion, and therefore, most methods use the composite bird’s-eye view synthesized from raw images. Dongwook Kim et al. presented a rule-based method that detects the lane lines and the stop lines in the bird’s-eye view image. Alexandru Gurghian et al. designed an end-to-end method to estimate lane positions in the bird’s-eye view image.

To generate bird’s-eye view image, it is supposed that the camera intrinsics and model are accurate and that the ground is flat. As it is hard to avoid the error of projection and undistortion, the features far from the cameras are always vague in the bird’s-eye view image. Therefore, most surround-view camera-based methods restrict the observation region of bird’s-eye view images to only detect 2 m or 3 m away from the ego vehicle (see Figure 2). However, it is important for lane-level localization system to get the observation of the curbs on multilane roads. For accurate features of the curb on the road, we design the perceptual model using raw images from surround-view cameras instead of bird’s-eye view image.

Localization

As noises and outliers are inevitable in the observed data, robust localization methods are developed for consistent position estimation. The Kalman filter is the most widely used localization method on linear systems. To work on nonlinear systems, the extended Kalman filter and the unscented Kalman filter were developed. Optimization-based methods have been demonstrated to have better performance than the Kalman filter and its variants on sophisticated nonlinear systems. Most localization methods suppose that all errors are Gaussian. Particle filters, however, generate samples without the assumptions of the state-space model and the state distribution.

Lane-level localization system

In this section, we present the lane-level localization system which outputs the robust positioning result using surround-view cameras, other low-cost sensors, and a lane-level road map (see Figure 3).

An MML is designed to provide the probabilistic lateral estimation as the observation. As the distribution of the MML outputs are non-Gaussian, we develop this system based on particle filters to fuse with the prediction from the IMU and WSS. The particle filter includes initialization, prediction, observation, weight update, resampling, and output estimation.

The steps of the proposed system are listed as follows:

- The system initializes with several particles. Each particle contains a state of the vehicle and the probability of being this state.
- Each particle predicts the motion estimation using the measurements of IMU and WSS.
The local PLRF is output from the perceptual model and matched with the sub-map to get the position observation.

The weight of the particles is updated by the distribution of observation.

The system decides if the resampling step should be performed, and outputs the posterior position estimation.

**Initialization**

An example of initialization is illustrated in Figure 4.

During initialization, we sample a number of particles close to the measurement from a low-cost GNSS receiver. Specifically, the state of the $i$th particle at time $k$, denoted as $x^{(i)}_k$, contains the Cartesian $XY$ coordinates $p_x^{(i)}_k, p_y^{(i)}_k$, and the heading angle $\theta_k^{(i)}$. Each particle has its weight, $\omega_k^{(i)}$, showing the possibility that the ego vehicle is at the particle’s position. The GNSS coordinates are firstly transformed to local coordinates $[p_x^{\text{gnss}}, p_y^{\text{gnss}}]^T$ using universal transverse mercator coordinates. According to the accuracy of the low-cost GNSS receiver, we sample $N$ particles within the valid region with the same weights. The initial direction of each particle is set to the direction of the road.

**Prediction**

At time $k, k > 0$, we use IMU, WSS, and the last state to predict the current state based on dead reckoning (DR). The local PLRF is output from the perceptual model and matched with the sub-map to get the position observation.
each particle gets a sub-map $m_k^{(i)}$ from the sub-map extraction module according to the prediction $x_k^{(i)}$. The perceptual model processes raw images from surround-view cameras to generate the feature $f_{\text{obs}, k}$. The sub-map and the feature are given to the matching model to calculate the probabilistic lateral estimation.

Sub-map extraction. Given the predicted position of a particle and a lane-level road map, the sub-map extraction module outputs a sub-map as prior knowledge. Unlike 3D dense maps, a lane-level road map only requires low-cost sensors, small storage, and little manual effort.

As most open-source digital maps only contain information at the road level, we generate a lane-level road map\(^\text{15}\) including the precise position of road centerlines, lane lines, and curbs. Unlike typical sub-maps, we devise a PLRF as our sub-map. The PLRF (see Figure 5) is a mixed representation of lane lines and the road surface in the lateral direction without limitations of driving conditions.

The sub-map is generated as follows.

Given a lane-level road map and the expected position of all the particles, we draw a line $l$ on the road through the expected position, perpendicular to the centerline of the road. As the generated map contains all the road centerlines, the centerline of this road is extracted by searching the centerline nearest to the expected position. We denote the intersection of the line $l$ and the road centerline as $O$. We extract a line segment $l_0$ from $l$ whose midpoint is $O$ with a length of $L_0$. Note that $L_0$ should be greater than the width of all the roads in the map to guarantee that the $l_0$ includes all the lateral information of the road. To simplify the computation, we discretize $l_0$ as a list of equidistant points $s = [s_0, ..., s_M]^T$. $P_A(s_j|h)$ denotes the probability that the point $s_j$ is on the lane line $h$. $h$ denotes the polyline stored in the lane-level map as the lane boundary. $P_B(s_j)$ denotes the probability that the specific point $s_j$ is on the road $r$. PLRF is a vector $f = [f_0, ..., f_M]^T$, whose entry $f_j$ is a value relative to the probability $P_A(s_j|h)$ and $P_B(s_j)$.

To compute PLRF, we estimate $P_A(s_j|h)$ and $P_B(s)$ according to the lane-level road map. There exist noises with 1–2 dm accuracy in our generated map, which is inevitable in most mapping methods. If the noise of the lane-level map is almost negligible, then we can set the probability $P_A(s_j|h)$ as a Dirac delta function. Considering this noise distribution from mapping, we set the probability $P_A(s_j|h)$ to follow the Gaussian distribution in the proposed system

$$P_A(s_j|h) \sim \mathcal{N}(\mu_h, \sigma_h)$$

where $\mu_h$ and $\sigma_h$ are the mean and standard deviation of the lane line $h$. The mean $\mu_h$ and the standard deviation $\sigma_h$ are determined by the generation approach of the lane-level map.

The event $B$ that a point is on the road is defined as follows:

- If $l$ intersects no curb, then all elements in $s$ are on the road. It occurs at the crossroads.
- If $l$ intersects once the curb of the road, all elements in $s$ at the right side of the left curb or all elements in $s$ at the left side of the right curb are on the road. It occurs at a three-way junction.
- If the line $l$ intersects twice the curbs of this road and the road is straight here, then all elements in $s$ between the two intersections are on the road. If the road is bending, we should treat the two curbs as the inner and outer curbs. The elements in $s$ between one inner curb and one outer curb are on the road. If the line $l$ may intersect twice the same curb, then the elements in $s$ between the inner curb are off the road, but between the outer curb are on the road.

\[ \text{Figure 5. The illustration of a PLRF and a local PLRF. A PLRF is a vector which contains the data correlated with the probability of the existence of lane lines and road surface on a lateral line segment whose midpoint is on the road centerline. A local PLRF is a vector shorter than the PLRF, and it represents the data on a lateral line segment whose midpoint is a possible ego position. PLRF: probabilistic lateral road feature.} \]
As we have defined the event $B$ above, $P_B(s)$ is determined as follows: If $s_j$ is on the road, then $P_B(s_j/C_0/C_1)$ is set to 1. Otherwise, $P_B(s_j/C_0/C_1)$ is set to zero. The element of the PLRF at the particle $i$ is defined as

$$f_j = \sum_h P_A(s_j|h) + 2P_B(s_j/C_0/C_1) - 1$$

(2)

It is noted from equation (2) that the position close to a lane line tends to have a larger value. If the position is on the road, it has a value not less than 1. Otherwise, the value is $-1$. Our PLRF mixes all the lateral characteristics in one vector to simplify further computations. The PLRF $f$ is defined as the sub-map $m_k$.

Perceptual model. The perceptual model extracts a local PLRF $f_{obs}$ from raw images. The local PLRF (see Figure 5) is regarded as the PLRF that is observed by surround-view cameras of the ego vehicle. The differences between local PLRFS and PLRFS are the midpoint and the length of the line segment. The possible ego position is set as the midpoint of the local line segment $l_{obs}$. The length of $l_{obs}$ is denoted as $L_1$.

The task of estimating the local PLRF from images is formulated as a regression problem. Designing a rule-based algorithm to extract local PLRF from raw images is a challenging task as raw images of fish-eye cameras have large distortions. Since deep neural networks have outperformed traditional machine learning approaches in challenging tasks like classification and regression, we design a deep neural network shown in Figure 6 to estimate the local PLRF $f_{obs}$.

At time $k$, four raw images from surround-view cameras are cropped and resized to the color ones with a dimension being 210 (height) $\times$ 280 (width). These images are fed into four shared-weight subnetworks. We take layers from off-the-shelf CNNs and load the pretrained weights for better performance. Specifically, we use the first five layers of AlexNet in this article. Each subnetwork followed by a fully connected layer and then these outputs are merged. To output a local PLRF, we apply three fully connected layers with 512 neurons each. The LeakyReLU activation function in this neural network is with the negative slope of 0.1.

Unlike conventional deep-learning-based methods, our model requires no manual labels. Our instrumented vehicle is equipped with real time kinematic-global positioning system (RTK-GPS) to provide the ground truth position, and we generate lane-level map in the test region. Given the ground truth position and a lane-level road map, the true local PLRF for training can be auto-generated. First, the sub-map (PLRF) through the ground truth $GT$ is extracted by equation (2). Then, we get the local PLRF by cropping the sub-map to change the midpoint to $GT$ and shorten the length of the vector. Therefore, the training data set for the perceptual model is generated without much manual effort. We take least square errors as the loss function. Note that

Figure 6. The perceptual model takes raw images from surround-view cameras as inputs and outputs a local PLRF using a deep neural network. The net contains four parallel subnetworks. Each subnetwork has the same structure as the convolutional layers of AlexNet to extract the features. Each subnetwork followed by a fully connected layer and then these outputs are concatenated together. The remaining three are fully connected. PLRF: probabilistic lateral road feature.
the accuracy of ground truth positions and the lane-level map are crucial to the performance of the perceptual model as they determine the accuracy of labels. If the lane-level road map is not accurate or noisy, the label may be influenced and then lead to a risky result. Therefore, the map should be checked before learning. If there is something wrong in the map, we should modify the map generation module or manually correct the map. It is supposed that the lane-level map is reliable and that it can be obtained from other sources such as the map company, so we do not introduce the method of map generation and maintenance in this article.

**Matching model.** The matching model (see Figure 7) estimates probabilistic lateral locations by computing the correlation between a local PLRF and a sub-map (PLRF). We adopt the idea of the correlation layer and define the relation between a local PLRF and a sub-map (PLRF). We introduce the method of map generation and maintenance other sources such as the map company, so we do not introduce the method of map generation and maintenance in this article.

![Figure 7. The matching model.](image)

To estimate the weight using the GMM output of the matching model. The weight is updated by

\[
\omega_{\text{upd},k}^{(i)} = \frac{\omega_{k}^{(i)} \sum_{h} g_{k,h}^{(i)} (\text{plat}_{k,}\text{h})}{\sum_{j} \omega_{j}^{(i)} \sum_{h} g_{k,h}^{(i)} (\text{plat}_{k,}\text{h})}
\]

**Resampling and output estimation**

Resampling is performed according to the effective number of particles\(^{33}\) which is computed by

\[
N_{\text{eff}} = \frac{1}{\sum_{i} \left(\omega_{\text{upd},i}^{(i)}\right)^2}
\]

If the \(N_{\text{eff}}\) is less than a given threshold \(N_{\text{thr}}\), then resampling is performed: draw \(N\) new particles with the same weight \(\omega_{k}^{(i)} = \frac{1}{N}\) and the probability distribution of them are the same as that of old particles. Otherwise, \(\mathbf{x}_{k}^{(i)} = \mathbf{x}_{k}^{(i)}\) and \(\omega_{k}^{(i)} = \omega_{\text{upd},k}^{(i)}\).

The output of the localization system is set to the weighted average of the particles.

**Experiments**

To evaluate our proposed system, we first generated the lane-level road map of the test region (see Figure 8) based on Ladybug3 spherical camera systems.\(^{15}\) In the whole area, we collected data several times in different days by our instrumented vehicle which is a Chery Tiggo car equipped with a Xsens MTi IMU, two WSS mounted on rear wheels, a low-cost GNSS receiver, and a RTK-GPS receiver. RTK-GPS was used to provide ground truth positions for evaluation. In the data-collection area, the accuracy of RTK-GPS is 2 cm. The Xsens MTi IMU is a nine-axis industrial-grade inertial sensor, and its yaw rate gyroscope is used for DR. We selected three regions to test the localization accuracy, while the data collected at the other regions were used for training the perceptual model of the MML.

Given the length of local PLRFs, we set the parameters of the last six layers of the perceptual model. The first convolutional layer after the concatenate layer contains 384 filters of \(3 \times 3\) with a stride of 1. The second convolutional layer contains 384 filters of \(3 \times 3\) with a stride of 2. The last convolutional layer is same as the second one. The first two fully connected layers are with the size of 384, followed by the last one of \(N_{\text{obs}} = 400\).

The perceptual model was built upon PyTorch, and we used the pretrained weights of AlexNet convolutional
layers to initialize the weights. We chose the Adam optimizer\textsuperscript{34} with a learning rate of 0.0001 and the weight decay of $10^{-5}$. The batch size is set to 64. After 300,000 iterations, we stopped the training process. For comparison, we also build a CNN-based model using the mono image. The nets and parameters are almost same, except for the number of input subnetworks. The parameters for experiments are presented in Table 1.

The test data were used to evaluate the performance of both the perceptual model and the lane-level localization system. Note that the data for testing and training were collected in different regions (see Figure 8) to evaluate the generalization ability of the perceptual model to extract the features of lane lines, ground, and curbs. The detailed description of test regions is presented in Table 2. The plants distribute periodically and grow gradually on both sides of most roads in the test regions, so it is difficult to locate the vehicle by observing the vegetation. To reduce restrictions on the surroundings, our method only observes the lane lines and curbs as features. There are curves, lane splitting, and multi-lane roads in the test regions. The vehicle was driving on the test roads several times to collect data for different conditions of illumination and traffic. Note that the results in this test cannot be applied directly in new areas with entirely different roads. If we also apply this method in a new area, additional training should be performed by adding new training data similar to this area. Nevertheless, the results in test regions are typical at this stage, as the data are collected in different conditions and can be used to evaluate the generalization ability on unseen roads. The processing system is a desktop with an Intel i7 CPU and an NVIDIA GTX TITAN X GPU. The proposed method is implemented using Python 2.7. The processing time of the proposed method is about 50 ms for one cycle.

If we take the position with the maximum probability as the output, the results of all the test data in three test region is shown in Figure 9, and the lateral positioning root mean square (RMS) errors are listed in Table 3. It is noted that the MML has better performance to use surround view images than the mono image. The success rate is listed in Table 4, which defines the inlier percentage of localization results.

### Table 1. Parameters for experiments.

| Unit | Value | Description                  |
|------|-------|------------------------------|
| $L_0$ | m     | 56                           |
| $L_1$ | m     | 20                           |
| $N_m$ | –     | 1120                         |
| $N_{obs}$ | – | 400                           |
| $N$ | –     | 100                          |
| $N_{thr}$ | – | 60                             |
| $\sigma_h$ | m | 0.15                         |
| $\sigma_x$ | m | 0.09                         |
| $\sigma_y$ | m | 0.09                         |
| $\sigma_\theta$ | rad | 0.003                       |
| $R_{gnss}$ | m | 10                           |

PLRF: probabilistic lateral road feature.

### Table 2. The description of test regions.

| Region | Length | # Lanes | Condition          |
|--------|--------|---------|--------------------|
| Region 1 | 0.6 km | 2       | Curves             |
| Region 2 | 0.6 km | 4       | Multi-lane roads   |
| Region 3 | 0.3 km | 2, 3, 4 | Lane-splitting roads |

---

**Figure 8.** The areas where we generated a lane-level road map. Samples for testing were collected on the roads within the red rectangles. The regions in red rectangles are shown as enlarged views besides satellite images. Samples for training were collected in other regions for the perceptual model.
In the first and second regions, the success rate using surround-view images is higher than that using the mono image. Although in the third region, the success rate using the surround-view images is lower than that using the mono image, its results have a smaller RMS error.

Figure 10 shows the accurate results in different driving conditions. In this figure, the sub-map at the position of ground truth is extracted so that the matching result evaluates the performance of the perceptual model. It is evident that the matching results show a large probability near ground truth, which demonstrates that the perceptual model provides effective local PLRFs adaptive to different driving conditions. Figure 10(a) shows that the car is changing the lane. The second case in Figure 10(b) is that the white arrows are on the ground. The car is driving into the sun before sunset in Figure 10(c). These cases are challenging to common rule-based lane-detection methods. Without extra effort, our deep-learning-based method overcomes this challenge, both the outputs using mono images and surround-view images are accurate.

Figure 11 shows the cases when the method using surround-view images outperforms that using the mono image. Figure 11(a) shows that the car is driving on the curve. Figure 11(b) shows a lane splitting case. In these two cases, the localization result using surround view images is still accurate, however, the result using the mono image does not have a large probability near the ground truth. That is because the mono image cannot capture the lateral features exactly at the left and right side of the vehicle. Using the mono image, the output of the perceptual model is confusing when the road condition is changing.

Figure 12 shows typical failure cases near the intersections. PLRF contains no more than one curb in these cases, so the features are mainly determined by the lane lines. As most lane lines on the test roads periodically distributed, the results will have multi-modal noise. If the localization fails near the intersection, the error may be a multiple of the lane width. Figure 13 shows the probabilistic lateral error from a surround-view image sequence in the first region. It is noted that both the maximum probabilistic position and the expectation are far from the ground truth at about 20 s when the car is at the T-shaped intersection (see Figure 12(b)).

To have robust positioning estimation, the proposed system fuses the results of MML with the prediction from IMU and WSS. The features on multi-lane roads are always periodic, so the noise can be under the multimodal distribution. Particle filters are adopted as the noises of MML is non-Gaussian. To compare with the proposed system, we also implement the Kalman filter. In the Kalman filter-based system, we use Gaussian distribution to approximate the observation probability. In the particle filter-based system, the particles update their weights by equation (4).

Figure 14 shows the result of localization systems from a sequence in the first test region. We can see that the system based on particle filters shows the better accuracy than that based on the Kalman filter, which demonstrate that the system based on particle filters is more robust to different driving conditions.

The results from numerous test cases verify that the localization method has a submeter accuracy even in different driving conditions. The accuracy is related to noises of the lane-level road map, as the map determines the PLRF and auto-generated labels. Most surround-view camera-based localization systems only provide the precise positioning estimation in the ego lane but cannot determine the

**Figure 9.** The lateral error distribution of the MML in the test regions. (a) The input is the mono image. (b) The input is surround-view images. MML: map-matching localizer.

**Table 3.** Lateral RMS error (m).

| Region  | Mono  | Surround view |
|---------|-------|---------------|
| 1       | 0.249 | 0.134         |
| 2       | 0.286 | 0.129         |
| 3       | 1.651 | 0.116         |

RMS: root mean square.

**Table 4.** Success rate (%).

| Region  | Mono  | Surround view |
|---------|-------|---------------|
| 1       | 96.86 | 98.33         |
| 2       | 92.74 | 94.07         |
| 3       | 98.79 | 92.81         |
Figure 10. Inputs and results in different driving conditions including the input images of the perceptual models, the local vector map, and the lateral localization errors from the MML. In the local vector map, the rectangle filled in green represents the ground truth of the ego vehicle, blue lines are curbs, and yellow or white lines represent lane lines. Although these conditions are difficult for most existing methods, the proposed method is still accurate. (a) The car is changing the lane on multi-lane roads. (b) Arrows on the ground of multi-lane roads. (c) The car is driving into the sun just before sunset. MML: map-matching localizer.

Figure 11. Inputs and results in different driving conditions. Although the driving condition is typical and the image quality is good, the method using mono image is not as accurate as that using surround view images. (a) The car is on the curve. (b) Three lanes after lane splitting.
lane on multi-lane roads, as they use the bird’s-eye view image from surround-view cameras. Some rule-based methods achieve centimeter-level accuracy in specific driving conditions. However, rule-based methods could meet challenges in diverse driving conditions. Conventional deep-learning-based methods may gain robust and accurate results, but their labels require much manual effort. Our proposed system is quite feasible with submeter accuracy in different driving conditions, which helps lane-level applications on a large scale.

Conclusions

In this work, we proposed a lane-level localization system adaptive to different driving conditions. The proposed system used surround-view cameras, other low-cost sensors and a lane-level road map. We proposed an MML that provided the probabilistic lateral estimation. In the MML, we devised the PLRF as the sub-map. The CNN-based perceptual model extracted a local PLRF from raw images captured by surround-view cameras. The matching model calculates the correlation between a sub-map and a local PLRF and outputs the probabilistic lateral estimation. The proposed localization system integrated the output of MML with the measurements from IMU, WSS, and a low-cost GNSS receiver based on particle filters. The proposed system is more feasible than common rule-based methods, as rule-based methods can hardly find out a perfect rule for diverse driving conditions. Besides, the proposed system requires little manual effort as the perceptual model was trained using auto-generated labels, unlike conventional...
deep-learning-based method. We evaluated the proposed system using data collected in different driving conditions. Experimental results demonstrated that the performance of MML is reliable in different driving conditions and that the localization method achieved submeter accuracy even in complicated situations. The localization error of the proposed system is inevitable due to the noises of the lane-level road map. Future work could improve the localization accuracy by fusing with more observations.

**Declaration of conflicting interests**
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Natural Science Foundation of China under Grant U1764264/61873165, in part by the International Chair on Automated Driving of Ground Vehicle, and in part by the Shanghai Automotive Industry Science and Technology Development Foundation under Grant 1733/1807.

**ORCID iD**
Tianyi Li https://orcid.org/0000-0003-3022-2627

**References**

1. Baldwin I and Newman P. Road vehicle localization with 2D push-broom LIDAR and 3D priors. In: 2012 IEEE International Conference on Robotics and Automation, Saint Paul, Minnesota, USA, 14 May 2012, pp. 2611–2617. IEEE.

2. Wolcott RW and Eustice RM. Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. *Int J Robot Res* 2017; 36(3): 292–319.

3. Basaca-Preciado LC, Sergiyenko OY, Rodriguez-Quinonez JC, et al. Optical 3D laser measurement system for navigation of autonomous mobile robot. *Opt Laser Eng* 2014; 54: 159–169.

4. Sergiyenko O, Tyrsa V, Flores-Fuentes W, et al. Machine vision sensors. *J Sensor* 2018; 2018. https://doi.org/10.1155/2018/3202761.

5. Lategahn H and Stiller C. Vision-only localization. *IEEE Trans Intell Transp Syst* 2014; 15(3): 1246–1257.

6. Brown AA and Brennan SN. Lateral vehicle state and environment estimation using temporally previewed mapped lane features. *IEEE Trans Intell Transp Syst* 2015; 16(3): 1601–1608.

7. Kim D, Kim B, Chung T, et al. Lane-level localization using an AVM camera for an automated driving vehicle in urban environments. *IEEE/ASME Trans Mech* 2017; 22(1): 280–290.

8. Gopalan R, Hong T, Shneier M, et al. A learning approach towards detection and tracking of lane markings. *IEEE Trans Intell Transp Syst* 2012; 13(3): 1088–1098.

9. Gurgian A, Koduri T, Bailur SV, et al. Deepplanes: end-to-end lane position estimation using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, 26 June–1 July 2016, pp. 38–45. IEEE.

10. Haklay M and Weber P. Openstreetmap: user-generated street maps. *IEEE Pervasive Comput* 2008; 7(4): 12–18.

11. McCormac J, Handa A, Davison A, et al. Semanticfusion: dense 3D semantic mapping with convolutional neural networks. In: 2017 IEEE International Conference on Robotics and automation (ICRA), IEEE, Singapore, 29 May 2017, pp. 4628–4635. IEEE.

12. Zhu J, Ai Y, Tian B, et al. Visual place recognition in long-term and large-scale environment based on CNN feature. In: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, Changshu, Suzhou, China, 26–30 June 2018 pp. 1679–1685. IEEE.

13. Joshi A and James MR. Generation of accurate lane-level maps from coarse prior maps and lidar. *IEEE Intell Transp Syst Mag* 2015; 7(1): 19–29.

14. Guo C, Kidono K, Meguro J, et al. A low-cost solution for automatic lane-level map generation using conventional in-car sensors. *IEEE Trans Intell Transp Syst* 2016; 17(8): 2355–2366.

15. Yang M, Gu X, Lu H, et al. Panorama-based multilane recognition for advanced navigation map generation. *Math Probl Eng* 2015: 1–14.

16. Aly M. Real time detection of lane markers in urban streets. In: 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, the Netherlands, 4–6 June 2008, pp. 7–12. IEEE.

17. Gackstatter C, Heinemann P, Thomas S, et al. Stable road lane model based on clothoids. *Advanced microsystems for automotive applications*. Berlin: Springer, 2010. pp. 133–143.

18. Loose H, Franke U, and Stiller C. Kalman particle filter for lane recognition on rural roads. In: 2009 IEEE Intelligent Vehicles Symposium, Xi’an, Shaanxi, China, 3 June 2009, pp. 60–65.

19. Lowe DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 2004; 60(2): 91–110.

20. Bay H, Tuytelaars T, and Van Gool L. Surf: speeded up robust features. In: *European conference on computer vision*. Springer, Berlin, Heidelberg, 7 May 2006, pp. 404–417. Springer.

21. Lategahn H, Beck J, and Stiller C. DIRD is an illumination robust descriptor. In: 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014, pp. 756–761. Springer.

22. Rublee E, Rabaud V, Konolige K, et al. Orb: an efficient alternative to sift or surf. In: *2011 International Conference on Computer Vision*, Barcelona, Spain, 6–13 November 2011, pp. 2564–2571. Springer.

23. Zhang B, Appia V, Pekkucuksen I, et al. A surround view camera solution for embedded systems. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 662–667. Springer.
24. Gruyer D, Belaroussi R, and Revilloud M. Accurate lateral positioning from map data and road marking detection. *Expert Syst Appl* 2016; 43: 1–8.
25. Julier SJ and Uhlmann JK. New extension of the Kalman filter to nonlinear systems. In: *Signal processing, sensor fusion, and target recognition VI*, volume 3068. International Society for Optics and Photonics, 28 July 1997, pp. 182–194. Society of Photo Optical.
26. Wan EA and Van Der Merwe R. The unscented Kalman filter for nonlinear estimation. In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Lake Louise, Alberta, 1–4 October 2000, pp. 153–158. Society of Photo Optical.
27. Strasdat H, Montiel JM, and Davison AJ. Visual slam: why filter? *Image Vision Comput* 2012; 30(2): 65–77.
28. Wu C, Huang TA, Muffert M, et al. Precise pose graph localization with sparse point and lane features. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Vancouver, BC, Canada, 24–28 September 2017, pp. 4077–4082. Society of Photo Optical.
29. Li T, Yang M, Zhou X, et al. A robust terrain-based road vehicle localization algorithm. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*, Piscataway, NJ, 19–June 2016, pp. 915–920.
30. Li F, Bonnifait P, Ibanez-Guzman J, et al. Lane-level map-matching with integrity on high-definition maps. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Redondo Beach, CA, USA, 11–14 June 2017, pp. 1176–1181. Society of Photo Optical.
31. Kao WW. Integration of GPS and dead-reckoning navigation systems. In: *Vehicle Navigation and Information Systems Conference, volume 2*, IEEE, 20 October 1991, pp. 635–643.
32. Dosovitskiy A, Fischer P, Ilg E, et al. Flownet: learning optical flow with convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, 26 April 2015, pp. 2758–2766.
33. Yu JX, Liu WJ, and Tang YL. Adaptive resampling in particle filter based on diversity measures. In: *2010 5th International Conference on Computer Science & Education*, Hefei, China, IEEE, 24–27 August 2010, pp. 1474–1478.
34. Kingma DP and Ba J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* 2014.