Stochastic Lane Shape Estimation Using Local Image Descriptors

Guoliang Liu, *Member, IEEE*, Florentin Wörgötter, and Irene Markelić

***Abstract*—In this paper, we present a novel measurement model for particle-filter-based lane shape estimation. Recently, the particle filter has been widely used to solve lane detection and tracking problems, due to its simplicity, robustness, and efficiency. The key part of the particle filter is the measurement model, which describes how well a generated hypothesis (a particle) fits current visual cues in the image. Previous methods often simply combine multiple visual cues in a likelihood function without considering the uncertainties of local visual cues and the accurate probability relationship between visual cues and the lane model. In contrast, this paper derives a new measurement model by utilizing multiple kernel density to precisely estimate this probability relationship. The uncertainties of local visual cues are considered and modeled by Gaussian kernels. Specifically, we use a linear-parabolic model to describe the shape of lane boundaries on a top-view image and a partitioned particle filter (PPF), integrating it with our novel measurement model to estimate lane shapes in consecutive frames. Finally, the robustness of the proposed algorithm with the new measurement model is demonstrated on the DRIVSCO data sets.**

***Index Terms*—Lane tracking, linear-parabolic model, local visual cues, multiple kernel density, partitioned particle filter (PPF).**

I. Introduction

Autonomous navigation on various roads requires the knowledge of lane information, which is also an open problem for driver-assistance systems. To extract lane boundary information, vision is a natural and powerful tool. However, high curvature, occlusions, varying illumination, and unmarked or partly marked lanes in the image are still challenging situations for this task [1], [2].

In general, locating lane boundaries in image sequences requires two steps: lane detection and lane tracking. Lane detection is to locate the lane boundaries in an image without strong prior knowledge regarding the lane position [3]. By contrast, the lane tracking is a procedure that tracks the lane edges from frame to frame by constraining the probable lane location in the current image using information about the lane location in previous images [4]. Lane detection can be regarded as an initialization step for lane tracking. To detect and track lane boundaries in images, we need some informative visual cues. Lane markers on the road are often painted in white or yellow, which contrasts with the dark concrete of the road. This white and yellow can be used as color cues for lane markers. The contrast between lane markers and concrete can be represented by gradient information, e.g., high contrast means high gradient magnitude. In addition, lane boundaries often can be represented by a mathematical model, e.g., straight lines and parabolic or circular curves. Here, we call this mathematical model of lane boundaries the *lane model* and parameters defined in this mathematical model as the *lane parameters*. Therefore, lane detection usually becomes a procedure that estimates lane parameters of the lane model from image cues, whereas lane tracking updates the parameters of this lane model for the remaining frames.

Depending on whether a prior knowledge of lane parameters is available in the lane detection step, previous methods can be grouped into two classes: bottom-up and top-down methods.

- Bottom-up methods try to obtain a lane model from image cues. To derive such a lane model, lane markers are usually extracted from images first. For instance, a threshold [5], a kernel mask [6], or a classifier [2], [7] can be used to find possible lane markers. This extraction step may generate a lot of edges that are not real lane markers; therefore, further processes are required, e.g., grouping [2], [8] and curve fitting [9]–[12].
- Top-down methods try to find optimized parameters of a known lane model that fits measured image cues best. Since the lane model is known, a set of hypotheses of lane parameters can be generated. To find out which hypotheses fits the measured image cues, usually, a likelihood function is used. This likelihood function can be defined as a mathematical function that describes the relationship between lane parameters and image cues. For evaluating these hypotheses, an optimization algorithm can be used, e.g., the metropolis algorithm [13], Tabu search [4], or importance resampling of particle filters [14]–[17].

Top-down methods in general are more flexible than bottom-up methods for combining multiple visual cues. Bottom-up methods usually need an edge extractor to find out possible lane markers, which often fails to locate the lane boundaries in images with strong distracting edges [4]. Moreover, the edge detector normally requires a threshold to detect potential lane edges. It is difficult to select such a threshold that eliminates the detection of noise edges without eliminating the detection of true lane edge pixels [18]. On the other hand, top-down methods can directly work on intensity and gradient information without thresholding. In addition, the likelihood function defined in

Manuscript received December 5, 2011; revised March 9, 2012 and June 8, 2012; accepted June 13, 2012. Date of publication July 10, 2012; date of current version February 25, 2013. This work was supported by the Bernstein Focus Neurotechnology Göttingen. The Associate Editor for this paper was P. Grisleri.

The authors are with Bernstein Center for Computational Neuroscience, Institute of Physics 3, University of Göttingen, 37077 Göttingen, Germany (e-mail: liu@physik3.gwdg.de; worgott@physik3.gwdg.de; irene@physik3.gwdg.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2205146

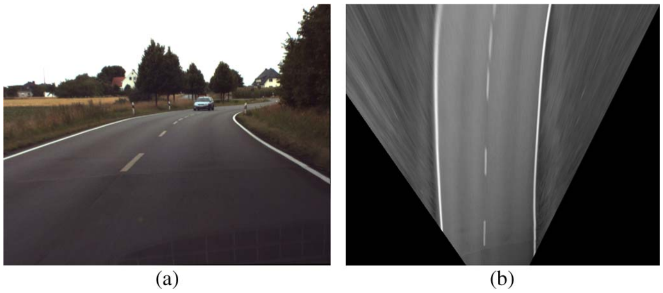
top-down methods can be any mathematical function that describes the relation between the lane model and image cues.

The likelihood function, which represents how well a hypothesis of lane parameters fits current image cues, is the key part of the most popular top-down methods. Previous methods try to derive this likelihood function by simply combining multiple visual cues in a multiplicative or additive manner assuming that visual cues are independent, without considering the uncertainties of local visual cues and the probability relation between the lane model and these visual cues. In contrast to others, we derive a new likelihood function by using multiple kernel density to estimate the probability density function of lane parameters. Here, the likelihood function is seen as a probability distribution function of lane parameters given image cues. Therefore, to derive the likelihood function is to estimate the probability density function of lane parameters.

In this paper, the probability distribution function of lane parameters is estimated by multiple kernel density, and the uncertainties of visual cues are modeled by Gaussian kernels. This idea was inspired by Dahyot [19], who proposed to use multiple kernel density to estimate straight lines, which is called statistical Hough transform (SHT). This SHT method is more robust than the standard Hough transform against image noise, as shown in the original paper. In addition, the SHT can directly work on the image density and gradient information without any thresholding. The main contribution of this paper is that we extend her idea to parabolic and linear-parabolic cases since most of lane boundaries have curved shapes.

After deriving the likelihood function, we use a partitioned particle filter (PPF) that is a novel top-down method to do lane detection and tracking. The original SHT algorithm is computationally expensive since no prior knowledge can be used; therefore, the SHT has to process all image data for each hypothesis of the lane parameters. In contrast to the SHT, the PPF can use prior knowledge of the lane model to generate a number of hypotheses, i.e., particles. Then, these hypotheses are verified by the derived likelihood function (also called measurement function) using nearby image data, which can greatly decrease the computational cost. Moreover, in contrast to Kalman-filter-based lane tracking methods [7], [20], the PPF maintains multiple hypotheses of lane parameters at any time, which is very useful to deal with challenging situations, e.g., occlusions, shadows, and lane changes. Furthermore, the PPF is better for handling high-dimensional states than the standard particle filter, as discussed in [14] and [21].

To demonstrate the proposed ideas, we choose a linear-parabolic model as the lane model in top-view images. The linear-parabolic lane model was introduced by Jung and Kelber [22], which is a trade-off between accuracy of the fit and robustness with respect to image artifacts. In contrast to Jung and Kelber who use the linear-parabolic model in the original image, we employ this model in the top-view image. The reasons are as follows. First, the parallel shape of the linear parts in the top-view image can be used to check the goodness of generated particles. Second, the parabolic model on the top-view image can be used to handle the curvature of lane boundaries. Third, this model can be well fitted into the PPF framework, which will be discussed in Section IV. The top-view images are produced by the well-known inverse perspective mapping (IPM) algorithm. IPM is an image transformation that removes the perspective effect, which is based on the flat ground hypothesis with known extrinsic and intrinsic parameters of the camera [23]. It remaps pixels from the original image to a top-view image that has a different coordinate system. This remapping procedure can be done by a fast lookup table with distortion compensation [24]. As the resolutions for near and far objects are different in the original image, an interpolation process is needed in the IPM algorithm. A resulting example image is shown in Fig. 1(b), which is a top-view image of Fig. 1(a) with cubic interpolation. We can see that the lanes in the IPM images look parallel and that the width between lanes becomes nearly constant. We note that the use of IPM is not mandatory but has proven to be useful because IPM creates a somewhat cleaned-up and less complex image where distractors (sky, objects, etc.,) are downplayed. Clearly, IPM can also produce problems (for example aliasing at large distances), but we still observed that IPM combined with a hierarchical model works well, mitigating such problems to a large degree.



(a) (b)

Fig. 1. (a) Image from the data set, where the lane markers are not parallel. (b) Transformed image after applying IPM shows a top view of the scene, and the lane markers appear nearly parallel.

This paper is organized as follows. Section II introduces the local image descriptors used in the algorithm. In Section III, the multi-kernel density estimation for line, parabolic, and linear parabolic are presented. Section IV describes the PPF for estimation of linear-parabolic parameters. Results and analyses are given in Section V, which introduces experimental results on different road situations. Finally, this paper is concluded in Section VI.

II. Local Image Descriptors

To detect lane markers from images, we need to define image descriptors (image cues) that describe the appearance features of lane markers. For every pixel in an image, basic descriptors can be defined, such as image intensity I_c, position (x, y), and gradient (I_x, I_y). Furthermore, advanced descriptors can be derived from these basic descriptors, e.g., the magnitude, alignment, and direction of the gradient: ΔI, ρ, and θ, respectively.

In practice, it is not necessary to use all descriptors since their information is redundant and noisy. In our case, we choose descriptors that are directly related to the lane model, i.e., the position (x, y) and gradient orientation θ of lane markers. Their uncertainties can be modeled by Gaussian kernels, which will be discussed in the following.

In addition, we also consider the magnitude of gradient ΔI and image intensity I_c since lane markers often have a specific color that is in contrast to the concrete background. However, the magnitude of gradient and image intensity are not directly related to the lane model; therefore, we need to find another way to put this information into the measurement model (likelihood function). Since the magnitude of gradient indicates the strength of edges, its inverse can be used as the variance of the gradient orientation θ, i.e., $\sigma_\theta^2 = (1/\Delta I)$. For instance, θ has a low variance if ΔI is high, whereas the weak edges have a high variance since ΔI is low. For the image intensity I_c, we simply combine its kernel with the probability distribution function of the lane parameters to form the final measurement model of the particle filter, which will be discussed in Section IV.

III. Lane Parameter Estimation by Multiple Kernel Density

Here, we use multiple kernel density to estimate the probability distribution of lane parameters given the observation space $Q_{xy\theta} = \{x_i, y_i, \theta_i, i = 1, \ldots, n\}$ of local image descriptors. Estimation of multiple kernel density is a nonparametric method to estimate the probability distribution function of random variables, given a number of samples of these variables. In our case, the random variables are lane parameters, whereas the samples are image pixels in the observation space $Q_{xy\theta}$. Here, we first introduce the previous work of estimating straight lines using multiple kernel density, which is also discussed in [19]. After that, we extend this idea to estimate parabolic and linear-parabolic shapes, which is the main contribution of this paper.

A. Line Model

For a straight line model, we have

$$\rho = x \cos\theta + y \sin\theta \tag{1}$$

where x and y are image coordinates, and ρ and θ are line parameters, which need to be estimated.

The probability distribution $p(\rho, \theta, x, y | Q_{xy\theta})$ can be written according to the Bayes rule as

$$p(\rho, \theta, x, y | Q_{xy\theta}) = p(\rho | x, y, \theta, Q_{xy\theta}) \cdot p(x, y, \theta | Q_{xy\theta}). \tag{2}$$

In (2), the first probability $p(\rho | x, y, \theta, Q_{xy\theta})$ is determined by (1), and the second probability $p(x, y, \theta | Q_{xy\theta})$ can be modeled by the multi-kernel density function. Thus, (2) becomes

$$p(\rho, \theta, x, y | Q_{xy\theta}) = \delta(\rho - x\cos\theta - y\sin\theta) \frac{1}{n} \sum_i K_x K_y K_\theta \tag{3}$$

where δ is the Dirac delta function, and $K_x = \mathcal{N}(x_i, \sigma_{x_i}^2)$, $K_y = \mathcal{N}(y_i, \sigma_{y_i}^2)$ and $K_\theta = \mathcal{N}(\theta_i, \sigma_{\theta_i}^2)$ are Gaussian kernels. $\sigma_{x_i}^2$, $\sigma_{y_i}^2$, and $\sigma_{\theta_i}^2$ are variances of x_i, y_i, and θ_i, respectively. In the experiment, we set $\sigma_{x_i}^2 = 1$, $\sigma_{y_i}^2 = 1$, and $\sigma_{\theta_i}^2 = (1/\Delta I)$.

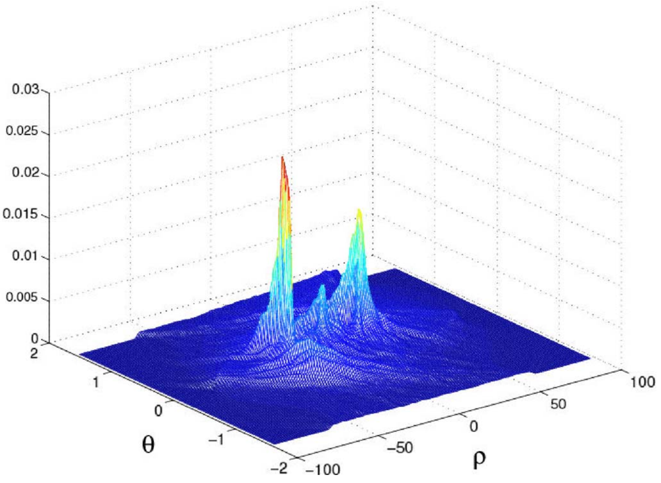


Fig. 2. Statistical distribution $p(\rho, \theta | Q_{xy\theta})$ given observation space $Q_{xy\theta}$ of Fig. 1(b).

The distribution $p(\rho, \theta | Q_{xy\theta})$ can be obtained by integrating (3) over (x, y)

$$p(\rho, \theta | Q_{xy\theta}) = \frac{1}{n} \sum_i K_\theta \cdot G_{li}(\rho, \theta) \tag{4}$$

where

$$G_{li}(\rho, \theta) = \frac{1}{\sqrt{2\pi \left(\sigma_{x_i}^2 \cos^2\theta + \sigma_{y_i}^2 \sin^2\theta\right)}} \cdot \exp\left(\frac{-(\rho - x_i\cos\theta - y_i\sin\theta)^2}{2\left(\sigma_{x_i}^2 \cos^2\theta + \sigma_{y_i}^2 \sin^2\theta\right)}\right). \tag{5}$$

A detailed description of (5) can be found in [19]. Given observation space $Q_{xy\theta}$ of an image, the statistical distribution of the lane parameters can be calculated using (4). One example result is shown in Fig. 2.

B. Parabolic Model

Similar to the line model estimation, the parabolic lane model also can be estimated using kernel densities. However, the parameter θ is not constant in the parabolic case. To model K_θ correctly, the gradient information is introduced to our parameter estimation.

For a parabolic lane model, we have

$$x = c + dy + ey^2 \tag{6}$$

where c, d, and e are the parabolic lane parameters. The orientation of the gradient θ is derived by the first-order derivative of (6)

$$\theta = a\tan(-2ey - d). \tag{7}$$

The probability distribution $p(c,d,e,x,y,\theta|Q_{xy\theta})$ can be written as

$$p(c,d,e,x,y,\theta|Q_{xy\theta})=p(c,d,e|x,y,\theta,Q_{xy\theta})\cdot p(x,y,\theta|Q_{xy\theta}) \tag{8}$$

where $p(c,d,e|x,y,\theta,Q_{xy\theta})$ is determined by (6) and (7), and the second probability term $p(x,y,\theta|Q_{xy\theta})$ can be modeled by multiple Gaussian kernel density

$$p(c,d,e,x,y,\theta|Q_{xy\theta}) = \delta_1 \cdot \delta_2 \cdot \frac{1}{n}\sum_i K_x K_y K_\theta \tag{9}$$

where $\delta_1=\delta(c+dy+ey^2-x)$ and $\delta_2=\delta(\theta-a\tan(-2ey-d)$ are Dirac functions, and K_x, K_y, and K_θ are Gaussian kernels, which are the same as in (3). The distribution $p(c,d,e|Q_{xy\theta})$ can be obtained by integrating (9) over (x,y,θ) as follows:

$$p(c,d,e|Q_{xy\theta}) = \frac{1}{n}\sum_i G_{pi}(c,d,e) \tag{10}$$

where

$$G_{pi}(c,d,e) = \int\int\int_{-\infty}^{\infty} \delta_1\delta_2 K_x K_y K_\theta \, dx dy d\theta. \tag{11}$$

Equation (11) is also known as the Radon transform. Because x and θ are represented by y using δ_1 and δ_2 functions, the threefold integration over (x,y,θ) in (11) is simplified to a single integration over y

$$G_{pi}(c,d,e) = \int_{-\infty}^{\infty} K'_x K_y K'_\theta \, dy \tag{12}$$

where

$$K'_x = \frac{1}{\sqrt{2\pi\sigma^2_{x_i}}} \exp\left(-\frac{(c+dy+ey^2-x_i)^2}{2\sigma^2_{x_i}}\right) \tag{13}$$

$$K'_\theta = \frac{1}{\sqrt{2\pi\sigma^2_{\theta_i}}} \exp\left(-\frac{(a\tan(-2ey-d)-\theta_i)^2}{2\sigma^2_{\theta_i}}\right). \tag{14}$$

Here, we cannot obtain an analytic expression of the Radon transform (12). Instead, we use the Gauss–Hermite numerical method to calculate G_{pi} [25].

C. Linear-Parabolic Model

The linear-parabolic model includes two parts: the linear part and the parabolic part. The linear part is used to model the lanes in the near vision field, whereas the parabolic part is used for the lanes in the far vision field [22], [26] as follows:

$$x = f(y) = \begin{cases} a+by, & \text{if } y > y_m \\ c+dy+ey^2, & \text{if } y \le y_m \end{cases} \tag{15}$$

where y_m is the border between the near and far vision fields, and the observation space $Q_{xy\theta}$ is divided into two subspaces: Q_{near} and Q_{far}. In the following, we assume that the separating line y_m is constant and at half of the image height, as shown in Fig. 3.

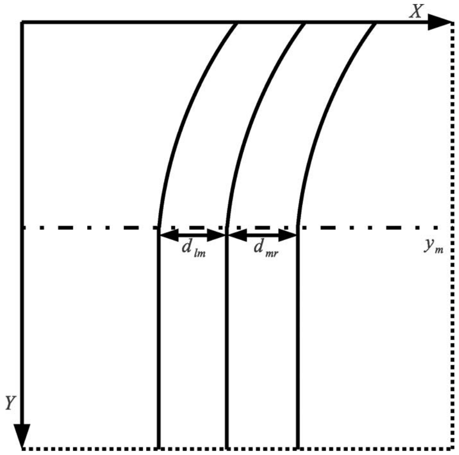


Fig. 3. Coordinate system of the image and the linear-parabolic multiple-lane model. In this case, the number of lanes is three. y_m is the separating line between the linear part (near vision field) and the parabolic part (far vision field). d_{lm} and d_{mr} are distances between the multiple lanes.

We impose continuity and differentiability conditions on function (15) at point y_m, such that $f(y_m^-)=f(y_m^+)$ and $f'(y_m^-)=f'(y_m^+)$. Combining with (1), we can further obtain

$$\begin{cases} a = \frac{\rho}{\cos(\theta)} \\ b = -\tan(\theta) \\ c = \frac{\rho}{\cos(\theta)} + \frac{y_m}{2}\left(-\tan(\theta)-d\right) \\ e = \frac{1}{2y_m}\left(-\tan(\theta)-d\right). \end{cases} \tag{16}$$

According to (16), the linear-parabolic model is totally determined by three parameters: (ρ,θ,d), given y_m. The coordinate system of the linear-parabolic model is shown in Fig. 3.

To estimate the linear-parabolic parameters (ρ,θ,d), we first estimate the linear part (ρ,θ) by (4) using observation Q_{near}, and then, we estimate the parabolic part d by (10) using observation Q_{far}. Because c and e are functions of (ρ,θ,d), we find

$$p(d|Q_{\text{far}}) = \frac{1}{n_{\text{far}}}\sum_i G_{pi}\left(c(\rho,\theta,d),d,e(\rho,\theta,d)\right). \tag{17}$$

This hierarchical process can be implemented by the PPF, as shown in Section IV.

IV. Partitioned Particle Filter for Lane Detection and Tracking

Here, we mainly illustrate the idea of how to use the probability distribution function of the lane parameters derived from multiple kernel density in the particle filter for lane detection and tracking. In [19], line parameters were estimated by combining the Hough transform with a multikernel probability model, i.e., SHT. However, the SHT is computationally expensive, and no prior information is used. By contrast, we use the PPF instead of the Hough transform to estimate the lane parameters. The measurement model of the PPF is derived by combining the probability distribution function of the lane parameters with the image intensity kernel. The PPF was designed for solving high-dimensional state problems [21]. First,

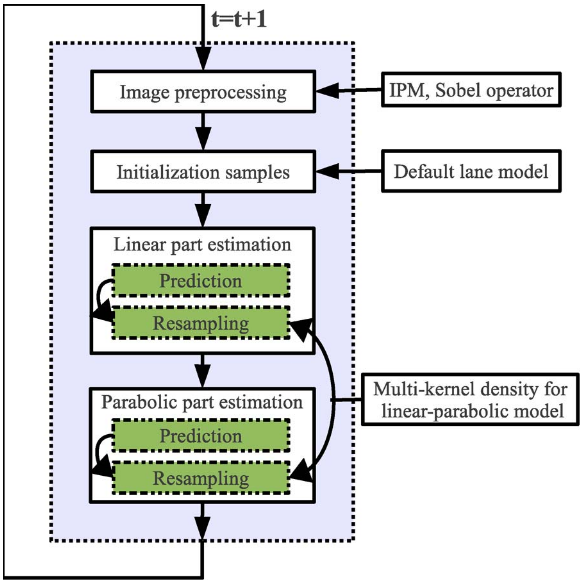


Fig. 4. Flowchart of our lane detection and tracking algorithm.

the high-dimensional state space in the PPF is partitioned into multiple subspaces. After that, each subspace is hierarchically estimated. This hierarchical process increases efficiency and robustness of the condensation algorithm [14].

For the linear-parabolic lane shape estimation, the state space of lane parameters in the PPF can be grouped into two subspaces: the linear part and the parabolic part. The PPF first estimates the linear part and then the parabolic part. Our whole algorithm is shown in Fig. 4. In the image preprocessing part, the perspective effect is removed by the IPM algorithm; then, the local image descriptors are calculated, e.g., gradient magnitude and orientation. After that, the initialization samples drawn from the default lane model are introduced to the particle array. These initialization samples can be used to recover the state from tracking failure [16]. Finally, the linear and parabolic parts are hierarchically estimated using the PPF with the observation model based on the multi-kernel density.

A. Multiple-Lane Model

Here, we use a simple multiple-lane model, as shown in Fig. 3. Additional conditions arising from the parallel lanes can be used to check the quality of the particles. For instance, the distances between each pair of nearby lanes are larger than 12 pixels, e.g., $d_{lm} > 12$ and $d_{mr} > 12$, and the difference between d_{lm} and d_{mr} is smaller than five pixels.

B. State Definition

In Section III-C, we introduce the probability estimation of linear-parabolic lane parameters from local image descriptors. The linear-parabolic lane model is defined by three parameters (ρ, θ, d). For the situation that the road has multiple lanes, the whole state probability distribution at time t is given by a set of N particles $X_t = \{x_t^j, j = 1, \ldots, N\}$, where $x_t^j = \{\rho_k^j, \theta_k^j, d_k^j, k = 1, \ldots, n_l\}$ is a single-particle state, and n_l is the number of lanes.

The dimension of the state x_t^j is $3 \times n_l$, which depends on the number of lanes n_l. In case of three lanes $n_l = 3$, state x_t^j has nine parameters. That means we need a large number of particles to correctly model this high-dimensional state distribution, which leads to high computational cost.

The solution to decrease the number of particles is to use the PPF, which separates the state x_t^j into two subgroups: the linear part $S_l = \{\rho_k^j, \theta_k^j, k = 1, \ldots, n_l\}$ and the parabolic part $S_p = \{d_k^j, k = 1, \ldots, n_l\}$. The subgroups are estimated by partitioned sampling in a hierarchical way. Because each subgroup requires less number of particles, the computational cost is lower.

C. Image Preprocessing

To remove the perspective effect of the image, an IPM algorithm is implemented on consecutive frames; then, the local gradient magnitude and orientation are calculated using the Sobel operator from the top-view image. To save computational cost, we threshold the gradient magnitude image to get an image mask. Only the pixels that have gradient values over this threshold are used as measurements, i.e., this threshold is set as ten in our experiments.

D. Initialization Samples

When lanes suddenly disappear, e.g., when occluded by a car, a robust algorithm has to maintain the best hypothesis or to find the lanes again when possible. The solution is to introduce a constant percentage of initialization samples of the PPF into the state distribution at every iteration. The initialization samples are drawn from the distribution of the default lane models. In our algorithm, we use $N' = 50$ initialization particles. This mechanism can recover the lane state from failures [4]. Here, we use the straight lane model as the default lane model, such that the value of d in random samples is defined as $d = b = -\tan(\theta)$.

E. Linear Part Estimation

The linear part estimation includes two steps: the first step is state prediction using a random-walk probability model. The second step is resampling using a specific observation model, i.e., the multikernel density model. Finally, only the particles that have high weight values are kept. This resampling process follows the idea of "survival of the fittest" [27].

1) Linear Part Prediction: Assuming the change of the lane boundary between two consecutive frames is small, a normal distribution can be used to model the state transition of the jth particle as

$$p\left(\hat{x}_t^{lj} | x_{t-1}^j) = \mathcal{N}(A^l x_{t-1}^j, \Sigma^l\right) \tag{18}$$

where $\mathcal{N}$ is the normal distribution. The matrix A^l is an identity matrix as we assume that the lane boundaries have smooth changes, and Σ^l is the covariance that handles the difference of lane boundaries between two consecutive frames. Because we only predict the linear part, Σ^l is defined as

$$\Sigma^l = \text{diag}\left\{V_k^l, k = 1, \cdots, n_l\right\} \tag{19}$$

where diag is the diagonal matrix function, and $V_k^l = \{\sigma_{\rho_k}^2, \sigma_{\theta_k}^2, 0\}$. $\sigma_{\rho_k}^2$ and $\sigma_{\theta_k}^2$ are the covariances of the Kth lane parameters ρ_k and θ_k. The predicted particle set is $\hat{X}_t^l = \{\hat{x}_t^{lj}, j = 1, \ldots, N_l\}$, where $N_l = N' + N$ is the number of particles used in the linear part estimation.

2) Linear Part Resampling: The observation space Q_k^l of the linear part describes the local features in the image. For the jth predicted particle, the measurement of the kth lane is z_{kt}^{lj}, which includes n_k^l pixels in Q_k^l that near the predicted lane. The observation model is derived from (4) with additional image intensity information

$$p\left(z_{kt}^{lj} | \hat{x}_t^{lj}\right) = \frac{1}{n_k^l} \sum_i K_{\text{ci}} \cdot K_{\theta_{kt}^j} \cdot G_{li}\left(\rho_{kt}^j, \theta_{kt}^j\right) \tag{20}$$

where $K_{\text{ci}} = \mathcal{N}(\mu_c, \sigma_c^2)$ is the Gaussian kernel model of the color intensity information, and μ_c and σ_c^2 are the mean and the covariance of this model. The weight for the jth particle is

$$w_t^{lj} = \eta \prod_{k=1}^{n_l} n_k^l p\left(z_{kt}^{lj} | \hat{x}_t^{lj}\right) \tag{21}$$

where η is a normalization factor, which ensures that the weights sum to 1. The number of candidate pixels n_k^l is considered as an important factor for weighting the particles because only the pixels that have higher gradient magnitude values are selected as candidate measurements. Finally, the new particle set $X_t^l = \{x_t^{lj}, j = 1, \ldots, N_p\}$ is obtained by resampling $\hat{X}_t^l$ based on the weights where N_p is the number of particles used to estimate the parabolic parameters. Those particles in $\hat{X}_t^l$ that have high weights will be kept, and the others that have lower weights will be removed from the particle set [28].

F. Parabolic Part Estimation

The algorithm for the parabolic part estimation is similar to the linear part, but the number of particles and the observation model used for estimation are different.

1) Prediction of Parabolic Part: The normal distribution is also used to model the state transition of the curved part as follows:

$$p\left(\hat{x}_t^{pj} | x_{t-1}^{lj}\right) = \mathcal{N}\left(A^p x_{t-1}^{lj}, \Sigma^p\right) \tag{22}$$

where $\mathcal{N}$ is a normal distribution, A^p is the identity matrix, and Σ^p is the covariance defined as

$$\Sigma^p = \text{diag}\left\{V_k^p, k = 1, \cdots, n_l\right\} \tag{23}$$

where $V_k^p = \{0, 0, \sigma_{d_k}^2\}$, and $\sigma_{d_k}^2$ is the covariance of the parabolic parameter d_k for the kth lane. As a result, the predicted particle set $\hat{X}_t^p = \{\hat{x}_t^{pj}, j = 1, \ldots, N_p\}$ is obtained.

2) Resampling of the Parabolic Part: For the jth particle, the measurement of the kth lane in the far-vision-field observation space Q_k^p is z_{kt}^{pj}, which corresponds to n_k^p pixels in Q_k^p that are near the predicted lane. The observation model can be derived from (17) as

$$p\left(z_{kt}^{pj} | \hat{x}_t^{pj}\right) = \frac{1}{n_k^p} \sum_i K_{\text{ci}} \cdot G_{pi}\left(\rho_{kt}^j, \theta_{kt}^j, d_{kt}^j\right). \tag{24}$$

The weight for the jth particle using the given observation model is

$$w_t^{pj} = \eta A \prod_{k=1}^{n_l} n_k^p p\left(z_{kt}^{pj} | \hat{x}_t^{pj}\right) \tag{25}$$

where η is the normalization factor. This process is called *weighted resampling* because it has an additional term $A = (1/w_t^{lj})$, such that it does not alter the distribution represented by the particle set [29]. However, this weighted resampling can introduce the impoverishment effect on the particle set [30]–[32]. To avoid this effect of the original PPF algorithm and to enforce the importance of the linear part, we set $A = (1/n_k^l)$. After resampling $\hat{X}_t^p$ by the weights, the new particle set $X_t = \{x_t^{pj}, j = 1, \ldots, N\}$ for the next iteration is obtained, and the weights are reset to be equal.

V. Results

To demonstrate the proposed algorithm, we selected two data sets from the DRIVSCO database [8], [33]: the Suburban Bridge data set and the Trailer data set. These data sets contain a variety of challenging situations like high-curvature roads, partly marked and occluded lanes, etc.

As aforementioned in Section I, the main contribution of this paper is the new measurement model derived by the multiple kernel density estimation. To evaluate the proposed model, two algorithms are required in the experiment: the PPF with the proposed multiple-kernel-density-based measurement model (PPF-Kernel) and the PPF with the measurement model that only uses image intensity and edge information (PPF-NoKernel), In other words, we compare the PPF-Kernel to the PPF-NoKernel to show the performance of the multiple-kernel-density-based measurement model.

For a quantitative analysis, we manually choose the lane markers in the IPM frames (every fifth frame) and then calculate the ground truth for the lane parameters using least squares. The root-mean-square error (RMSE) of the results from PPF-NoKernel and PPF-Kernel are shown in Fig. 6 for the Suburban Bridge data set and in Fig. 8 for the Trailer data set. In addition, we project the estimation results on the IPM images to allow visual inspection of the results, as shown in Fig. 5, for the Suburban Bridge data set and in Fig. 7 for the Trailer data set. In the following, we will give more details and analyses on these estimation results.

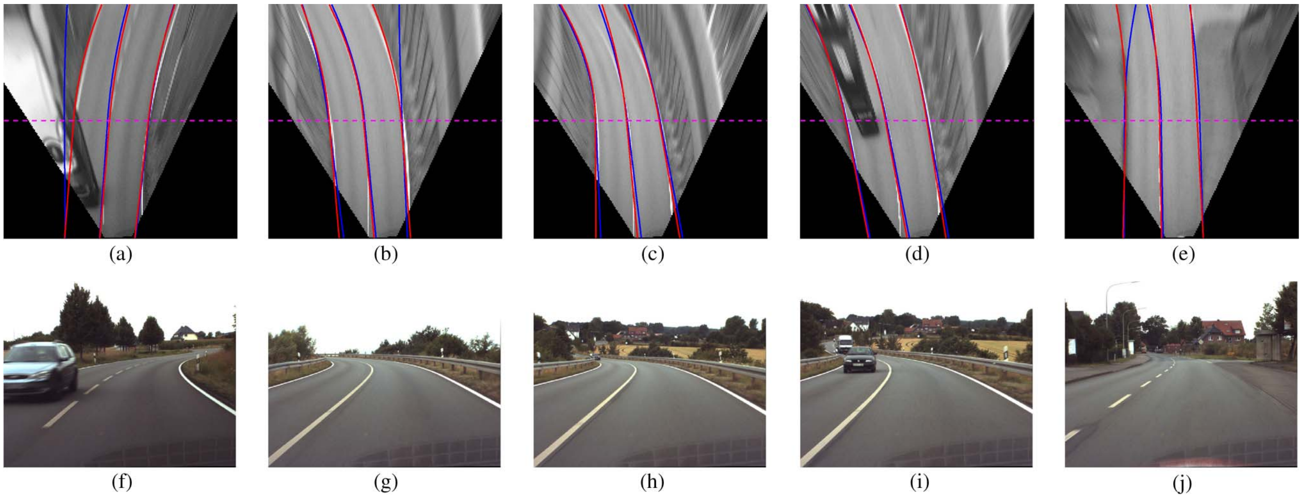
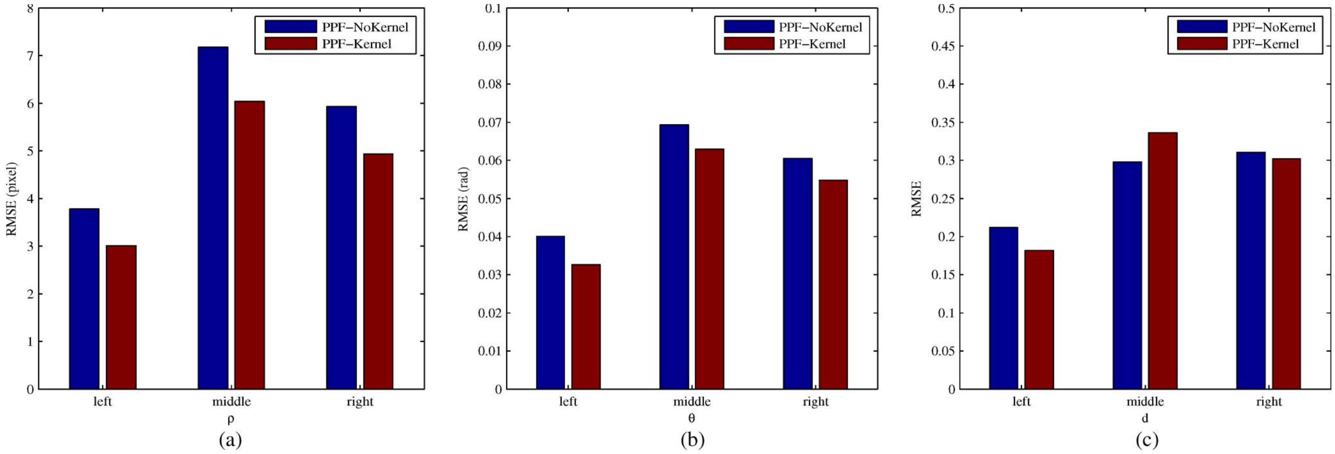


Fig. 5. Estimation results drawn from the IPM images of the Suburban Bridge data set. (Top) Results estimated by (red) PPF-Kernel and (blue) PPF-NoKernel. (Bottom) Original images from the camera. (a) IPM frame 38. (b) IPM frame 195. (c) IPM frame 470. (d) IPM frame 505. (e) IPM frame 748. (f) Original frame 38. (g) Original frame 195. (h) Original frame 470. (i) Original frame 505. (j) Original frame 748.



(a)

(b)

(c)

Fig. 6. RMSE of the estimated lane parameters (ρ, θ, d) by PPF-Kernel and PPF-NoKernel from the Suburban Bridge data set. Left, middle and right indicate the left, middle and right lanes, respectively.

A. PPF-Kernel Versus PPF-NoKernel

To show the performance of the proposed kernel-density-based measurement model, we compare the PPF-Kernel method to the PPF-NoKernel method. For PPF-NoKernel, we only employ the image intensity and edge information in the measurement models, which are defined as

$$p\left(z_{kt}^{lj}|\hat{x}_t^{lj}\right) = \frac{1}{n_k^l}\sum_i K_{\text{ci}} \tag{26}$$

$$p\left(z_{kt}^{pj}|\hat{x}_t^{pj}\right) = \frac{1}{n_k^p}\sum_i K_{\text{ci}} \tag{27}$$

where $p(z_{kt}^{lj}|\hat{x}_t^{lj})$ and $p(z_{kt}^{pj}|\hat{x}_t^{pj})$ are the measurement models for the linear part and the parabolic part, respectively; n_k^l is the number of particles for the linear part; and n_k^p is the number of particles for the parabolic part. $K_{\text{ci}} = \mathcal{N}(\mu_0, \sigma_0^2)$ is the Gaussian kernel model of the image intensity information. As lane markers are white with a high intensity value in this data set, we set $\mu_0 = 1$ and $\sigma_0^2 = 0.5$ for the normalized image, i.e., the original intensity value is divided by 255.

For the Suburban Bridge data set, the RMSE of the estimated lane parameters is shown in Fig. 6. It is clear that the PPF-Kernel has a smaller RMSE than the PPF-NoKernel for most lane parameters. We can also see the robustness of the PPF-Kernel against occlusions and noise (the bridge), whereas the PPF-NoKernel has larger errors for the estimation of the parabolic part, as shown in Fig. 5(a), (b), and (e). Furthermore, the PPF has a good performance to handle occlusions by the car, as shown in Fig. 5(a) and (d). One reason is the hierarchical processing of the PPF. Because we separate the lane into two parts, i.e., the linear part and the parabolic part, even if one part is occluded, the measurements from the other part can still support the right particles. The other reason is that we introduce random particles (drawn from the default lane model) into the particle set for each iteration, such that the PPF can recover from tracking failures.

For the Trailer data set, the curvature of the lanes is very small; thus, the difference of the estimation results between the PPF-Kernel and PPF-NoKernel are minor, which is in Fig. 7. However, the PPF-Kernel still has a smaller RMSE than the PPF-NoKernel for most lane parameters, as shown in Fig. 8.

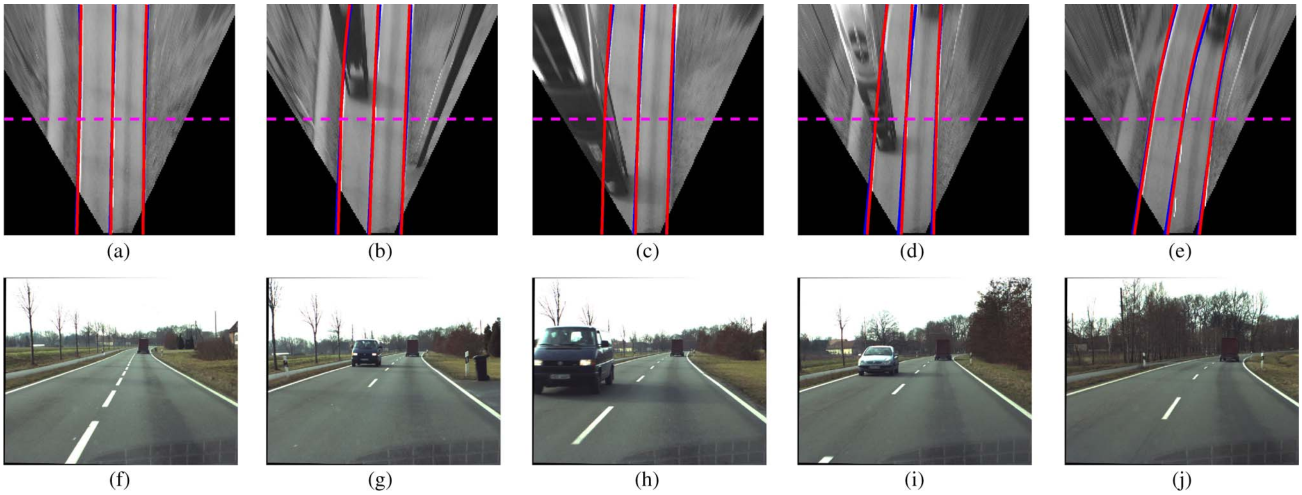


Fig. 7. Estimation results drawn from the IPM images of the Trailer data set. Top images show the results estimated by (red) PPF-Kernel and (blue) PPF-NoKernel. The bottom images are the original images from the camera. (a) IPM frame 20. (b) IPM frame 112. (c) IPM frame 123. (d) IPM frame 321. (e) IPM frame 446. (f) Original frame 20. (g) Original frame 112. (h) Original frame 123. (i) Original frame 321. (j) Original frame 446.

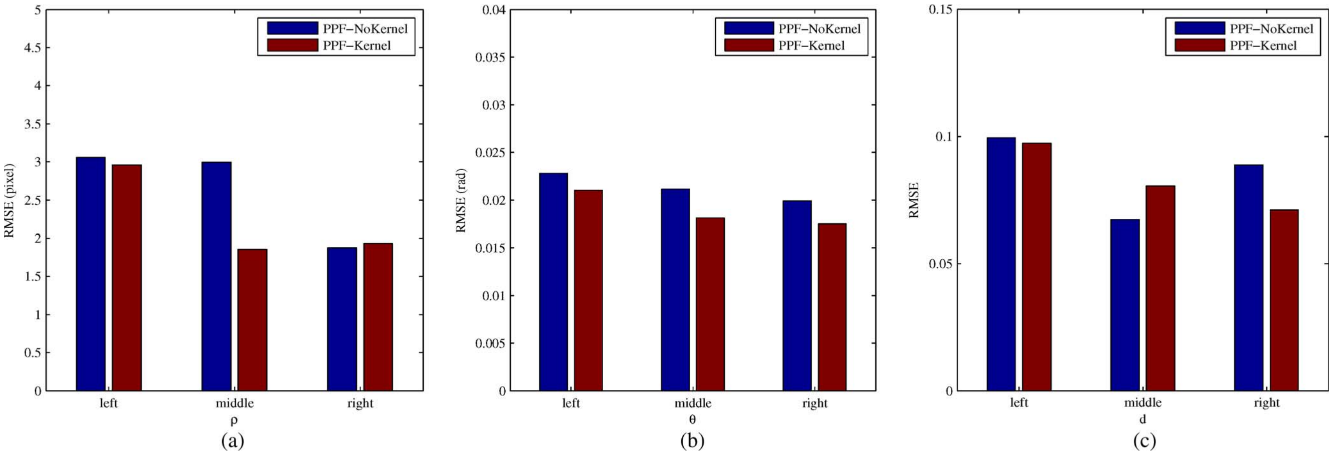


Fig. 8. RMSE of the estimated lane parameters (ρ, θ, d) by PPF-Kernel and PPF-NoKernel from the Trailer data set. Left, middle, and right indicate the left, middle and right lanes, respectively.

TABLE I
COMPARISON OF COMPUTATION TIME

Dataset	Method	Run time per frame (s)
Suburban Bridge	PPF-NoKernel PPF-Kernel	$0.11s$ $0.62s$
Trailer	PPF-NoKernel PPF-Kernel	$0.13s$ $0.66s$

B. Computational Cost

The current algorithm is programmed on a modern computer with an Intel Core 2 Quad central processing unit. The comparison of computational time is shown in Table I. The PPF-NoKernel is faster than the PPF-Kernel. The reason is that the Gauss–Hermitte numerical integral method in the PPF-Kernel is computationally expensive. However, there are several ways to speed up the proposed algorithm. For example, pixels in the observation space can be independently processed. Thus, this can be easily parallelized and computed by Graphics Processing Unit [34].

VI. CONCLUSION

In this paper, we have mainly focused on the improvement of the measurement model for lane shape estimation in the framework of particle filters. We considered the measurement model as a probability distribution function of the lane parameters. Therefore, we could employ multiple kernel density estimation to estimate this probability density. Furthermore, the final measurement model was derived by combining this estimated probability distribution function with the image intensity kernel. Finally, we used the PPF to estimate a linear-parabolic model on the DRIVSCO data sets as a demonstration of the proposed ideas. The experimental results show that the PPF with the proposed new measurement model (PPF-Kernel) is very robust concerning challenging scenes and achieves better estimation results than the PPF with an image intensity and an edge measurement model (PPF-NoKernel).

The main contribution of this paper is the use of multiple kernel density estimation in the measurement model. To get with this, we have introduced three measurement models (linear, parabolic, and linear-parabolic). Here, we only show how to use the proposed measurement model for the linear-parabolic shape estimation, but other measurement models could be also used in the same way. Furthermore, the proposed parabolic shape estimation can also be used for other applications, e.g., eyelid detection [25].

References

[1] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[2] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008.

[3] C. Kreucher, S. Lakshmanan, and K. Kluge, "A driver warning system based on the LOIS lane detection algorithm," in *Proc. IEEE IV*, 1998, pp. 17–22.

[4] Y. Zhou, R. Xu, X. Hu, and Q. Ye, "A robust lane detection and tracking method based on computer vision," *Meas. Sci. Technol.*, vol. 17, no. 4, pp. 736–745, Apr. 2006.

[5] M. Bertozzi and A. Broggi, "Real-time lane and obstacle detection on the gold system," in *Proc. IEEE Intell. Veh. Symp.*, Sep. 19–20, 1996, pp. 213–218.

[6] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," in *Proc. Intell. Veh. Symp.*, Oct. 1994, pp. 109–114.

[7] R. Aufrre, R. Chapuis, and F. Chausse, "A model-driven approach for real-time road recognition," *Mach. Vis. Appl.*, vol. 13, no. 2, pp. 95–107, 2001.

[8] I. Markelić, A. Kjaer-Nielsen, K. Pauwels, L. B. W. Jensen, N. Chumerin, A. Vidugiriene, M. Tamosiunaite, M. V. Hulle, N. Kruerger, A. Rotter, and F. Woergoetter, "The driving school system: Learning basic driving skills from a teacher in a real car," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1135–1146, Dec. 2011.

[9] D. J. Kang, J. W. Choi, and I. S. Kweon, "Finding and tracking road lanes using 'line-snakes,'" in *Proc. IEEE Intell. Veh. Symp.*, Sep. 1996, pp. 189–194.

[10] B. Yu and A. K. Jain, "Lane boundary detection using a multiresolution hough transform," in *Proc. Int. Conf. Image Process.*, Oct. 26–29, 1997, vol. 2, pp. 748–751.

[11] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 4–6, 2008, pp. 7–12.

[12] A. Borkar, M. Hayes, and M. T. Smith, "A novel lane detection system with efficient ground truth generation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 365–374, Mar. 2012.

[13] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection," in *Proc. Intell. Veh. Symp.*, Sep. 25–26, 1995, pp. 54–59.

[14] B. Southall and C. Taylor, "Stochastic road shape estimation," in *Proc. 8th IEEE ICCV*, 2001, vol. 1, pp. 205–212.

[15] U. Franke, H. Loose, and C. Knoppel, "Lane recognition on country roads," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2007, pp. 99–104.

[16] R. Danescu and S. Nedevschi, "Probabilistic lane tracking in difficult road scenarios using stereovision," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 272–282, Jun. 2009.

[17] R. Gopalan, T. Hong, M. Shneier, and R. Chellappa, "A learning approach towards detection and tracking of lane markings," *IEEE Trans. Intell. Transp. Syst.*, DOI: 10.1109/TITS.2012.2184756.

[18] C. Kreucher and S. Lakshmanan, "Lana: A lane extraction algorithm that uses frequency domain features," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 343–350, Apr. 1999.

[19] R. Dahyot, "Statistical hough transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1502–1509, Aug. 2009.

[20] E. Dickmanns and B. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, Feb. 1992.

[21] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Int. J. Comput. Vis.*, vol. 39, no. 1, pp. 57–71, Aug. 2000.

[22] C. R. Jung and C. R. Kelber, "A robust linear-parabolic model for lane following," in *Proc. 17th Brazilian Symp. Comput. Graph. Image Process.*, Oct. 17–20, 2004, pp. 72–79.

[23] M. Bertozzi and A. Broggi, "Gold: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62–81, Jan. 1998.

[24] T. Bergener and C. Bruckho, "Compensation of non-linear distortions in inverse-perspective mappings," Inst. Neuroinformatik, Ruhr-Univ. Bochum, Bochum, Germany, Apr. 1999, Tech. Rep.

[25] X. Liu, Q. Song, and P. Li, "A parabolic detection algorithm based on kernel density estimation," in *Proc. 5th Int. Conf. Intell. Comput.*, 2009, pp. 405–412.

[26] K. H. Lim, K. P. Seng, L.-M. Ang, and S. W. Chin, "Lane detection and Kalman-based linear-parabolic lane tracking," in *Proc. Int. Conf. IHMSC*, Aug. 26/27, 2009, vol. 2, pp. 351–354.

[27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA: MIT Press, Sep. 2005.

[28] G. Liu, F. Wörgötter, and I. Markelić, "Combining statistical hough transform and particle filter for robust lane detection and tracking," in *Proc. IEEE IV*, Jun. 2010, pp. 993–997.

[29] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2000, vol. 2, pp. 3–17.

[30] S. Duffner, J.-M. Odobez, and E. Ricci, "Dynamic partitioned sampling for tracking with discriminative features," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2009, pp. 1–11.

[31] K. Smith and D. Gatica-perez, "Order matters: A distributed sampling method for multi-object tracking," in *Proc. BMVC*, 2004, pp. 1–10.

[32] G. Liu, F. Wörgötter, and I. Markelić, "Lane shape estimation using a partitioned particle filter for autonomous driving," in *Proc. IEEE ICRA*, 2011, pp. 1627–1633.

[33] E. Baseski, L. Baunegaard With Jensen, N. Pugeault, F. Pilz, K. Pauwels, M. Van Hulle, F. Wörgötter, and N. Krüger, "Road interpretation for driver assistance based on an early cognitive vision system," in *Proc. VISSAP*, 2009, pp. 5–8.

[34] A. Abramov, E. E. Aksoy, J. Drr, F. Wörgötter, and B. Dellen, "3D semantic representation of actions from efficient stereo-image-sequence segmentation on GPUs," in *Proc. 5th Int. Symp. 3D Data Process., Vis. Transm.*, 2010, pp. 1–8.

Guoliang Liu (S'11–M'12) received the B.Sc. degree in physics from Shandong Normal University, Jinan, China, in 2005 and the M.Eng. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently working toward the Ph.D. degree in computer science with the University of Göttingen, Göttingen, Germany.

His research interests include computer vision, object tracking, robot navigation, nonlinear estimation, and sensor fusion.

Florentin Wörgötter received the Ph.D. degree from the University of Essen, Essen, Germany, in 1988.

From 1988 to 1990, he experimentally worked on visual cortex before he turned to computational issues with the California Institute of Technology, Pasadena. After 1990, he became a Researcher with the University of Bochum, Bochum, Germany, where he was concerned with experimental and computational neuroscience of the visual system. Between 2000 and 2005, he was a Professor of computational neuroscience with the Department of Psychology, University of Stirling, Stirling, U.K., where his interests strongly turned toward learning in neurons. Since July 2005, he has been leading the Department for Computational Neuroscience, Bernstein Center, University of Göttingen, Göttingen, Germany. His group has developed the RunBot, which is a fast and adaptive biped walking robot. His research interests include information processing in closed-loop perception–action systems, including aspects of sensory processing, motor control, and learning/plasticity. These approaches are tested in walking and driving robotic implementations.

Irene Markelić received the B.Sc. and M.Sc. degrees in computer science from the University of Koblenz–Landau, Koblenz, Germany, in 2002 and 2005, respectively, and the Ph.D. degree in computer science from the University of Göttingen, Göttingen, Germany, in 2010.

She is currently a Postdoctoral Researcher with the Institute of Physics 3, University of Göttingen. Her research interests include computer vision, developmental robotics, cognitive skill learning in humans and robots, and data mining.