

# Lane Shape Estimation Using a Partitioned Particle Filter for Autonomous Driving

Guoliang Liu, Florentin Wörgötter and Irene Markelić

**Abstract**—This paper presents a probabilistic algorithm for lane shape estimation in an urban environment which is important for example for driver assistance systems and autonomous driving. For the first time, we bring together the so-called Partitioned Particle filter, an improvement of the traditional Particle filter, and the linear-parabolic lane model which alleviates many shortcomings of traditional lane models. The former improves the traditional Particle filter by subdividing the whole state space of particles into several subspaces and estimating those subspaces in a hierarchical structure, such that the number of particles for each subspace is flexible and the robustness of the whole system is increased. Furthermore, we introduce a new statistical observation model, an important part of the Particle filter, where we use multi-kernel density to model the probability distribution of lane parameters. Our observation model considers not only color and position information as image cues, but also the image gradient. Our experimental results illustrate the robustness and efficiency of our algorithm even when confronted with challenging scenes.

## I. INTRODUCTION

Robust vision based street lane detection and tracking is an important factor for driver assistance systems (DAS) and autonomous driving, which can reduce the risk of car accidents. Despite numerous approaches developed in the past [1], many challenges remain in this area. One important aspect is the lane model used in the detection algorithms. Simple models, such as a straight line, do not allow accurate lane fits but are more robust against image artifacts. Complex models, on the other hand, such as parabolic, circular, and spline models are more flexible and, thus, allow better fits but are more sensitive to noise [2], [3].

In this paper, we use the linear-parabolic model introduced by Jung and Kelber [2] which is a trade-off between accuracy of the fit and robustness with respect to image artifacts. This model partitions the image into a near and a far vision field, as shown in Fig. 3, and assumes that the lanes are linear in the near and parabolic in the far field. Jung and Kelber's method for estimating the linear-parabolic model comprises two steps. First, an initial detection was performed with the well-known Hough transform (HT), and second, tracking of the detected lanes was realized by minimizing a weighted square error. However, a fixed-width tracking mask was needed to find possible lane edges in every iteration, and the used least squares method is sensitive to image noise. An

improvement was suggested by Lim et al. [4] who used the Kalman filter (KF) instead of the least squares method for model parameter estimation which was shown to be more robust. However, the KF maintains one hypothesis only and hence it is difficult to recover the true state after a tracking failure occurred [5]. In contrast, the Particle filter (PF) is able to maintain multiple hypotheses using simple measurements and is thus better suited for recovery from a tracking failure [5], [6].

Here, we use a version of the PF, called Partitioned Particle filter (PPF) [7] which not only inherits the advantages of the standard PF, but also benefits from an introduced hierarchical processing scheme by subdividing the state space of particles into subspaces and processing each subspace individually. By this, the computational complexity of the algorithm is reduced and its robustness increased.

Southall and Tailor [6] used the PPF to implement a lane detection and tracking framework and further developed a failure recovery mechanism to improve the tracking. The latter was achieved by introducing a constant number of random particles, which were drawn from provided default lane models, into the particle set of the PPF during each iteration. This failure recovery mechanism was also used by Zhou [5] and Danescu [3] with different lane models. Zhou et al. [5] used a PF to estimate a deformable template model, which was based on the assumption that lane markers on the ground floor are described as parabolas. Their observation model included both position and gradient direction information. Danescu and Nedeveschi [3] presented a lane detection and tracking system that used the PF without tracking initialization phase, i.e., the random particles drawn from default lane models were used to find possible lanes in the first frame.

Our paper further improves a) the linear-parabolic model estimation by combining it with a PPF, and b) introduces a new improved observation model in the PPF. Furthermore, we also include a failure recovery mechanism by drawing a number of random particles from default lane models during each iteration. We achieve a) by grouping the state space of the linear-parabolic model into several subspaces, which are estimated by the PPF in two steps, first the linear part, then the parabolic part. The intentions behind this are as follows: first, it is easier to formulate constraints that describe the parallel relation between multiple lanes for the linear part, i.e., equal widths and similar slopes. Second, for the driver the near vision field is more important than the far vision field, such that the linear part should be assigned a higher priority. Third, each subspace can have a

This work has been supported by European Commission under project FP6-IST-FET (DRIVSCO).

Guoliang Liu, Florentin Wörgötter and Irene Markelić are with Bernstein Center for Computational Neuroscience, III. Physikalisches Institut - Biophysik, University of Göttingen, 37077 Göttingen, Germany  
{liu, worgott, irene}@physik3.gwdg.de

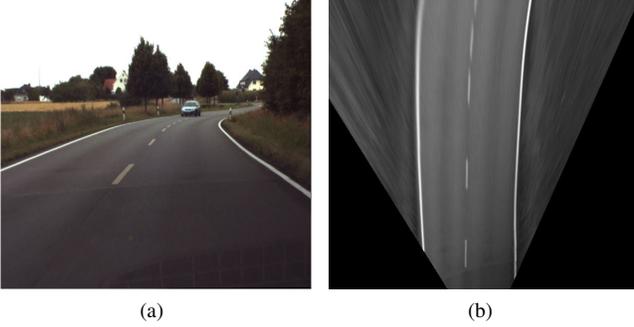


Fig. 1: (a) An image from the data set, where the lane markers are not parallel. (b) The transformed image after applying IPM shows a top-view of the scene and the lane markers appear nearly parallel.

flexible number of particles, which reduces the computational complexity and increases the system's robustness as shown in our experiments. We achieve b), the introduction of an improved observation model, by using multi-kernel density estimation [8] to model the probabilistic distribution of the model parameters and by considering gradient information as local descriptor of the image surface. By contrast, many other approaches use only color and position information. We show that an accurate estimate of a probabilistic distribution of the linear-parabolic model parameters can be derived with this additional information.

Finally, We apply our algorithm to images which show a top-view of the original scene, see Fig. 1. These images are obtained using a well-known algorithm called Inverse Perspective Mapping (IPM) [9], [10], and we thus refer to them as IPM-images. The paper is organized as follows: Section II introduces the local image descriptors that are used in the algorithm's observation model. In Section III, the multi-kernel density estimation for the straight line, parabolic and linear-parabolic model are presented. Section IV describes the PPF for linear-parabolic parameter estimation. Results and analysis with different road situations are given in Section V. Finally, the work is concluded in Section VI.

## II. LOCAL IMAGE DESCRIPTORS

To detect lane markers in the IPM-images, certain image descriptors can be used, such as color  $C$ , position  $(x, y)$  and gradient  $(I_x, I_y)$ . Furthermore, we can also derive advanced descriptors based on these basic ones, e.g., the magnitude, alignment, and direction of the gradient:  $\Delta I$ ,  $\rho$  and  $\theta$ . In an urban environment lane markers are not always of the same color, they may even be missing, such that color information  $C$  cannot be used as a reliable cue alone, however, position and gradient information are still useful in this case. Our algorithm for lane parameters estimation uses local descriptors  $(x, y, \theta)$  as observation and models them by kernel density including both position and gradient information [8]. Thus, the observation space for kernel density estimation for each image is  $Q_{xy\theta} = \{x_i, y_i, \theta_i, i = 1, \dots, n\}$ , where  $n$  denotes the total number of pixels that are used for estimating the lane

parameters. But for the observation model of the PPF, the color information is also used by modeling it as a Gaussian function.

## III. LANE PARAMETER ESTIMATION BY KERNEL DENSITY

In [8] and [11] the probabilistic distribution of straight line parameters is modeled as multi-kernel density, and the candidates are found by comparing the probability of different line parameters. This process is called Statistical Hough transform (SHT). The SHT works on the entire image using the full information that the image offers to find possible lines (a so-called dense-method in contrast to sparse methods which work on sparse information like image edges or corners). This method is more robust to noise than the standard HT. In [12] the SHT method was extended to detect parabolic structures. However, the gradient angle  $\theta$  was modeled as a constant, which is not realistic.

In a similar manner we use multi-kernel density to estimate the statistical distribution of parabolic lane parameters. But the gradient angle  $\theta$  of the parabolic lane is not constant, such that an additional constraint is needed for the estimation. In this section, we first introduce the parameter estimation of the linear part using multi-kernel density, then we show the new estimation method for parabolic lane estimation, and finally we extend the work to linear-parabolic parameter estimation.

### A. Straight line model

A straight line model is given by

$$\rho = x \cos \theta + y \sin \theta, \quad (1)$$

where  $x$  and  $y$  are horizontal and vertical image coordinates,  $\rho$  and  $\theta$  are the line parameters which need to be estimated. These parameters  $(x, y, \rho, \theta)$  can be interpreted as random variables and described by a probability density function which is conditioned on  $Q_{xy\theta}$ , i.e.,  $p(\rho, \theta, x, y | Q_{xy\theta})$ . This is written according to the Bayes rule as

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

When  $x$ ,  $y$  and  $\theta$  are known, the first term  $p(\rho | x, y, \theta, Q_{xy\theta})$  is determined by (1), and the second term  $p(x, y, \theta | Q_{xy\theta})$  can be modeled by kernels. Thus (2) becomes

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

where  $\delta$  is the Dirac delta function,  $K_x = \mathcal{N}(x_i, \sigma_x^2)$ ,  $K_y = \mathcal{N}(y_i, \sigma_y^2)$ , and  $K_\theta = \mathcal{N}(\theta_i, \sigma_\theta^2)$  are Gaussian kernels and  $\sigma_x^2$ ,  $\sigma_y^2$  and  $\sigma_\theta^2$  are the variances of  $x_i$ ,  $y_i$ , and  $\theta_i$ , respectively. The distribution  $p(\rho, \theta | Q_{xy\theta})$  is obtained by integrating (3) over  $(x, y)$

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

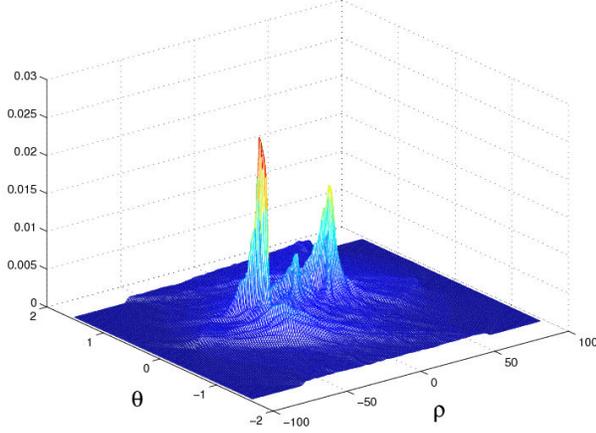


Fig. 2: Given observation space  $Q_{xy\theta}$  of Fig. 1b, the statistical distribution  $p(\rho, \theta | Q_{xy\theta})$  of straight line parameters is estimated using kernel density.

where

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

A detailed description of (5) can be found in [8]. Given the observation space  $Q_{xy\theta}$  of an image, the statistical distribution of the line parameters is calculated using (4). One example result is shown in Fig. 2.

### B. Parabolic model

Similar to the straight line model estimation, the parabolic lane model can also be estimated using kernel density. However, the gradient angle  $\theta$  of parabolic lanes is not constant. To model the kernel density of  $\theta$  correctly, we need more constraints and the parabolic lane model is defined as

$$x = c + dy + ey^2, \quad (6)$$

where  $c$ ,  $d$  and  $e$  are the parabolic lane parameters. The gradient angle is derived from (6)

$$\theta = \text{atan}(-2ey - d). \quad (7)$$

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

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

In the above equation, the first probability  $p(c, d, e | x, y, \theta, Q_{xy\theta})$  is determined by (6) and (7). The second probability  $p(x, y, \theta | Q_{xy\theta})$  can be modeled by multiple Gaussian kernel density. Thus, equation (8) becomes

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

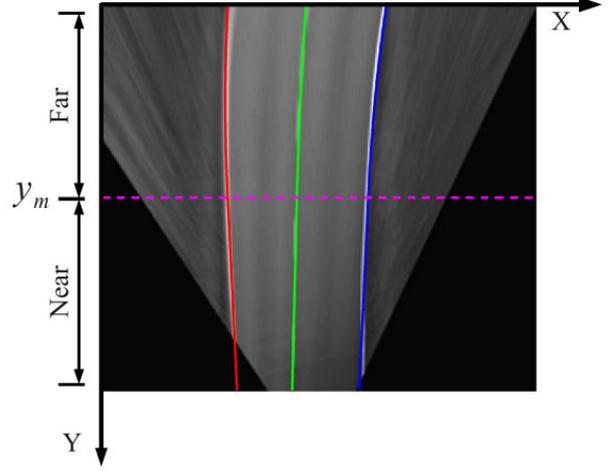


Fig. 3: The coordinate system of the image and the linear-parabolic model.  $y_m$  is the separate line between the near vision field and the far vision field.

where  $\delta_1 = \delta(c + dy + ey^2 - x)$  and  $\delta_2 = \delta(\theta - \text{atan}(-2ey - d))$  are Dirac functions,  $K_x$ ,  $K_y$ , and  $K_\theta$  are Gaussian kernels which are same as in (3). The distribution  $p(c, d, e | Q_{xy\theta})$  is obtained by integrating (9) over  $(x, y, \theta)$

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

where

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

Equation (11) is also known as Radon transform. Because  $x$  and  $\theta$  are represented by  $y$  using  $\delta_1$  and  $\delta_2$  functions, the three-fold integration over  $(x, y, \theta)$  in (11) is simplified to a single integration over  $y$

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

where

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

$$K'_\theta = \frac{1}{\sqrt{2\pi\sigma_{\theta_i}^2}} \exp\left(-\frac{(\text{atan}(-2ey - d) - \theta_i)^2}{2\sigma_{\theta_i}^2}\right). \quad (14)$$

The analytic expression of the Radon transform (12) cannot be obtained. To solve it, the Gauss-Hermite numerical method is used to calculate  $G_{pi}$  [12].

### C. Linear-parabolic model

The linear-parabolic model includes two parts: the linear 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 lanes in the far vision field [2], [4], as shown in Fig. 3.

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

where  $y_m$  is the border between the near and the far vision fields, and the observation space  $Q_{xy\theta}$  is divided into two subspaces:  $Q_{near}$  and  $Q_{far}$ .

Function (15) imposes continuity and differentiability conditions on point  $y_m$  such that  $f(y_m^-) = f(y_m^+)$  and  $f'(y_m^-) = f'(y_m^+)$ . By combining it with (1), we can further obtain  $a = \frac{\rho}{\cos(\theta)}$ ,  $b = -\tan(\theta)$ ,  $c = \frac{\rho}{\cos(\theta)} + \frac{y_m}{2}(-\tan(\theta) - d)$  and  $e = \frac{1}{2y_m}(-\tan(\theta) - d)$ . The linear-parabolic model is totally determined by three parameters:  $(\rho, \theta, d)$  given  $y_m$ .

To estimate the linear-parabolic parameters  $(\rho, \theta, d)$ , we first determine the linear part  $(\rho, \theta)$  by using (4) conditioned on observation  $Q_{near}$ , then we estimate the parabolic part  $d$  by (10) conditioned on observation  $Q_{far}$ . Because  $c$  and  $e$  are functions of  $(\rho, \theta, d)$ , we have

$$p(d|Q_{far}) = \frac{1}{n_{far}} \sum_i G_{pi}(c(\rho, \theta, d), d, e(\rho, \theta, d)). \quad (16)$$

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

#### IV. PARTITIONED PARTICLE FILTER FOR LANE DETECTION AND TRACKING

In [8] and [12], the authors estimate line and parabolic parameters by the SHT. However, the SHT is computational expensive for curved lane detection. Here, we use the PPF to estimate the lane parameters and introduce the multi-kernel density of the SHT as observation model with additional color information. The whole state space is subdivided into several subspaces, such that only a small number of particles are needed for each subspace. In addition, the number of particles for each subspace can be flexibly chosen. This way, the PPF saves computational time and increases the robustness of the system.

The whole algorithm is shown in the Fig. 4. In the image preprocessing part, the perspective effect is removed by the IPM algorithm, and the size of the image is decreased to ease off the computation time, then the vertical edge image is obtained by the Sobel edge extractor. Afterwards, the initialization samples drawn from the default lane model are introduced to the particle array. As we mentioned in the Section I, those initialization samples can be used to recover the state from tracking failure. Finally the linear and parabolic part are estimated hierarchically using the PPF with the observation model based on the multi-kernel density.

##### A. State definition

In the previous Section III-C, we introduced the probability estimation of linear-parabolic lane parameters. The linear-parabolic lane model is defined by three parameters  $(\rho, \theta, d)$  given  $y_m$ . For the situation that the road has three lanes, the entire state probability distribution at time  $t$  is given by a set of  $N$  particles  $X_t = \{x_t^j, j = 1, \dots, N\}$ , where  $x_t^j = (\rho_l^j, \theta_l^j, d_l^j, \rho_m^j, \theta_m^j, d_m^j, \rho_r^j, \theta_r^j, d_r^j)$  is the state space of a single particle which consists of the left, middle and right lane parameters.

The number of particles depends on the dimension of the particle state. The state  $x_t^j$  has nine parameters that must

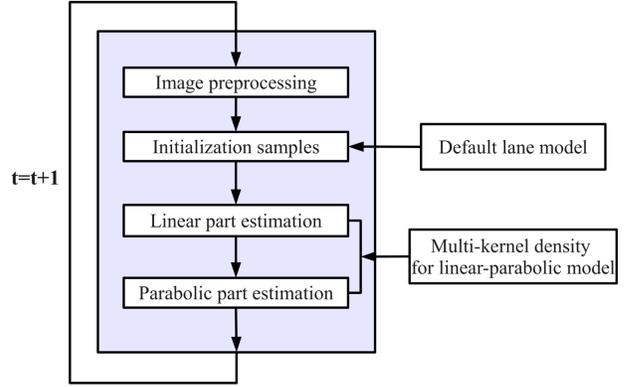


Fig. 4: The outline of our linear-parabolic lane shape estimation algorithm based on PPF with multi-kernel density observation model.

be estimated, this means that a large number of particles is necessary to accurately model their probability distribution. If the entire state space can be grouped into several subspaces, then for each subspace only a small number of particles are needed. Hence, we group the particle state space into two subspaces, and different numbers of particles are assigned to these subspaces. In short, the hierarchical procedure of the PPF is as follows: first the linear part  $S_l = \{\rho_l, \theta_l, \rho_m, \theta_m, \rho_r, \theta_r\}$  is estimated, then the parabolic part is estimated, which is  $S_p = \{d_l, d_m, d_r\}$ . The linear part  $S_l$  must be assigned more particles than the parabolic part, because the linear part is more important and has three more parameters in the state vector.

##### B. Initialization samples

When lanes suddenly disappear, e.g., are occluded by a car, a robust algorithm must maintain the best hypothesis or find the lanes again when possible. The solution is to introduce a constant percentage of initialization samples into the state distribution at every iteration, which are drawn from the distribution of the default lane models. In our algorithm, we use  $N' = 20\%N$  initialization particles. This mechanism can recover the lane state from failures [5]. Here we use the straight lane model as default lane model, such that the value of  $d$  in random samples are defined as  $d = b = -\tan(\theta)$ .

##### C. 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., multi-kernel density model. Finally, only the particles which have high weight value are kept. This resampling process follows the idea of survival of the fittest [13].

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  $j_{th}$  particle as

$$p(\hat{x}_t^l | x_{t-1}^l) = \mathcal{N}(A^l x_{t-1}^l, \Sigma^l), \quad (17)$$

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

$$\Sigma^l = \text{diag}(\sigma_\rho^2, \sigma_\theta^2, 0, \sigma_\rho^2, \sigma_\theta^2, 0, \sigma_\rho^2, \sigma_\theta^2, 0), \quad (18)$$

where  $\text{diag}$  is the diagonal matrix, and  $\sigma_\rho^2$  and  $\sigma_\theta^2$  are the covariances of the line parameters  $\rho$  and  $\theta$ . The predicted particle set is  $\hat{X}_t^l = \{\hat{x}_t^{lj}, j = 1, \dots, N_l\}$ , where  $N_l = N' + N$  is the number of particles used in the linear part estimation.

2) *Linear part resampling*: The observation space  $Q_{near}$  of the linear part describes the local features in the image. For the  $j_{th}$  predicted particle, the measurement of the left lane is  $z_{lt}^{lj}$  which includes  $n_{near}$  pixels in  $Q_{near}$  that are near the predicted left lane. The observation model is derived from (4) with additional color information

$$p(z_{lt}^{lj} | \hat{x}_t^{lj}) = \frac{1}{n_{near}} \sum_i K_{ci} \cdot K_{\theta_i^j} \cdot G_{li}(\rho_{lt}^j, \theta_{lt}^j), \quad (19)$$

where  $K_{ci} = \mathcal{N}(\mu_c, \sigma_c^2)$  is the Gaussian kernel model of the color information, and  $\mu_c$  and  $\sigma_c^2$  are mean and covariance of this model. The other measurements  $p(z_{mt}^{lj} | \hat{x}_t^{lj})$  and  $p(z_{rt}^{lj} | \hat{x}_t^{lj})$  for middle and right lanes are computed in the same way as described above. The weight for the  $j_{th}$  particle used in above observation model is

$$w_t^{lj} = \eta p(z_{lt}^{lj} | \hat{x}_t^{lj}) p(z_{mt}^{lj} | \hat{x}_t^{lj}) p(z_{rt}^{lj} | \hat{x}_t^{lj}), \quad (20)$$

where  $\eta$  is a normalization factor, which ensures that the weights sum to one. Finally, the new particle set  $X_t^l = \{\hat{x}_t^{lj}, j = 1, \dots, N_p\}$  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 a high weight will be kept and the others that have lower weight will be removed [11]. To avoid impoverishment effect of original PPF algorithm [14], [15], and enforce the importance of linear part, the weights for particles in  $X_t^l$  are reset to be equal.

#### D. Parabolic part estimation

The algorithm of 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 curve part:

$$p(\hat{x}_t^p | x_{t-1}^p) = \mathcal{N}(A^p x_{t-1}^p, \Sigma^p) \quad (21)$$

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

$$\Sigma^p = \text{diag}(0, 0, \sigma_d^2, 0, 0, \sigma_d^2, 0, 0, \sigma_d^2) \quad (22)$$

where  $\sigma_d^2$  is the covariance of the parabolic parameter  $d$ , and the three lanes have the same covariance. As a result, the predicted particle set  $\hat{X}_t^p = \{\hat{x}_t^{pj}, j = 1, \dots, N_p\}$  is obtained.

2) *Resampling of parabolic part*: For the  $j_{th}$  particle, the measurement of the left lane in the far vision field observation space  $Q_{far}$  is  $z_{lt}^{pj}$ , which corresponds to  $n_{far}$  pixels in  $Q_{far}$  that are near the predicted left lane. The observation model can be derived from Eq. (16):

$$p(z_{lt}^{pj} | \hat{x}_t^{pj}) = \frac{1}{n_{far}} \sum_i K_{ci} \cdot G_{pi}(c(\rho_{lt}^j, \theta_{lt}^j, d_{lt}^j), d_{lt}^j, e(\rho_{lt}^j, \theta_{lt}^j, d_{lt}^j)) \quad (23)$$

The measurement distributions  $p(z_{mt}^{pj} | \hat{x}_t^{pj})$  and  $p(z_{rt}^{pj} | \hat{x}_t^{pj})$  for middle and right lanes are computed in the same way as above. The weight for the  $j_{th}$  particle using the above observation model is:

$$w_t^{pj} = \eta p(z_{lt}^{pj} | \hat{x}_t^{pj}) p(z_{mt}^{pj} | \hat{x}_t^{pj}) p(z_{rt}^{pj} | \hat{x}_t^{pj}) \quad (24)$$

where  $\eta$  is normalization factor. After resampling on  $\hat{X}_t^p$  by the weights, the new particle set  $X_t = \{\hat{x}_t^{pj}, j = 1, \dots, N\}$  for the next iteration is obtained, and the weights are reset to be equal.

## V. RESULTS

To demonstrate the performance of our algorithm, we applied it to the EISATS dataset [16], which contains a variety of challenging situations like strong shadow, high-curvature roads, partly marked and occluded lanes, etc. In the experiments, we use  $N = 250$  particles to track the linear part and  $N_p = 200$  particles to estimate the parabolic part. For every iteration  $N^l = 50$  initialization samples, drawn from default lane models, are introduced into the particle set. The covariances of the lane model parameters in the test are  $\sigma_\rho = 1$ ,  $\sigma_\theta = 0.02$ , and  $\sigma_d = 0.05$ . The estimation result is defined as a weighted average of the particle state value in the set  $X_t$ .

For comparison, we also test the kernel density observation model with the general PF, which uses 50 particles for initialization and 250 particles for tracking. The tracking results are shown in Fig. 5, which shows that the general PF is more sensitive to noise, especially in the far vision field. However, the proposed PPF algorithm gives better estimation results using the same amount of particles, as shown in Fig. 6. The Fig. 7 shows the lanes back-projected into the original image. Furthermore, our algorithm also succeeds in tracking sequences that have strong illumination artifacts and brightness differences, where only right and middle lanes are estimated using the PPF, as shown in Fig. 8. These results demonstrate the algorithm's robustness with respect to these difficult situations. A video is also available online at [17].

The advantage of the hierarchical processing in the PPF is shown in Fig. 6c, where the car is on a bridge and the ground is not flat anymore, such that there are no lane markers in the far vision field of the IPM-image. The hierarchical processing first finds the straight lanes in the near vision field, and further finds the parabolic part based on the straight lane detection results. As shown in Fig. 6c, the noisy information on the parabolic part has no effect on the straight lane part which is more important for driving. Although the parabolic estimation is random in Fig. 6c, after a few frames, the PPF

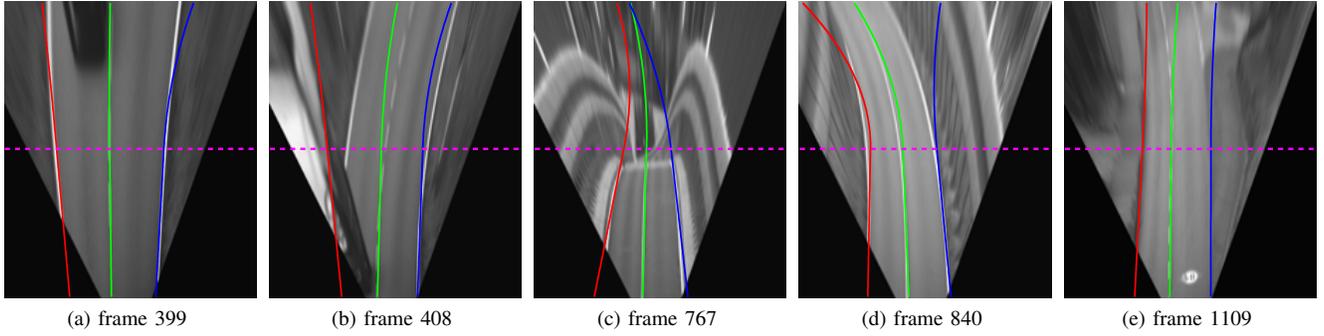


Fig. 5: Estimation results on the IPM images using multi-kernel density model as observation model of the general Particle filter. Cases are: (a) dashed lanes, (b) occluded by car, (c) non-flat ground plane, (d) high curvature image, (e) partly marked lanes.

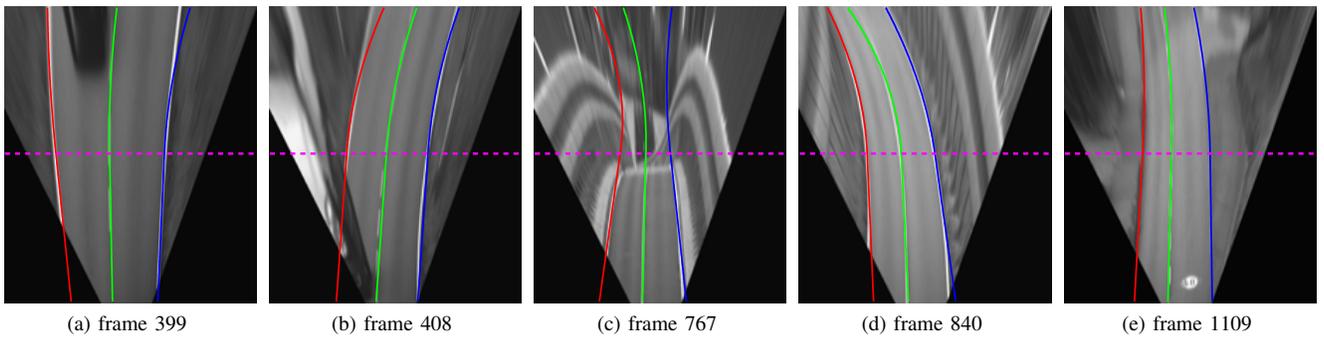


Fig. 6: Estimation results on the IPM images using multi-kernel density model as observation model of the Partitioned Particle filter.

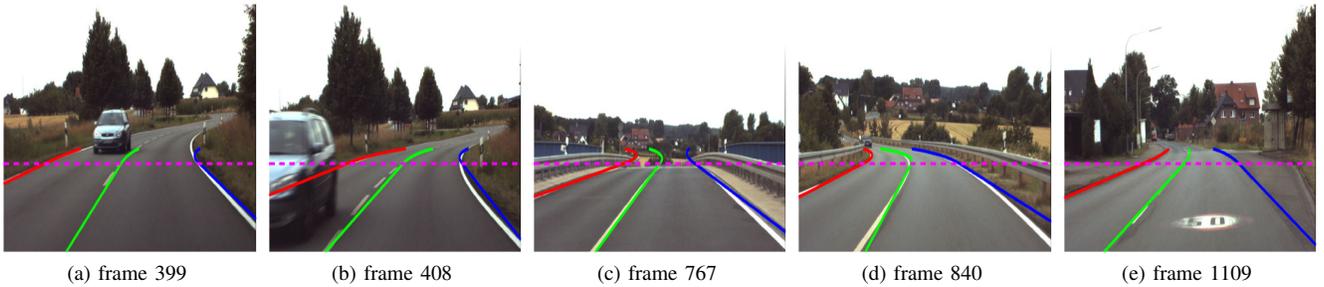


Fig. 7: Tracking results in Fig.6 are projected back to the original images.

still can give accurate estimations when the parabolic part is available again in the image as shown in Fig. 6d. This shows the robustness of our algorithm.

The current algorithm was programmed in Matlab on a Laptop with INTEL QUAD CORE (2.5GHz) for simulation and analysis purpose. Because the Gauss-Hermite numerical integral method is very slow in Matlab, the computations are expensive and our algorithm cannot be used for online processing yet (e.g., 10s for Fig. 6 and 5s for Fig. 8). However, this can be improvable substantially since the algorithm structure allows for implement using parallel programming. For example, the pixels in the observation space can be processed independently. Another limit of our algorithm is

that if the lanes in the near vision field have high curvature, the estimation error will increase, which can be seen in Fig. 7b and Fig. 7d. This limit comes from our linear assumption and the constant  $y_m$ . To solve this problem, a self adaptive algorithm for updating  $y_m$  can be developed in the future.

## VI. CONCLUSION

We presented our idea to combine multi-kernel density based parameter distribution estimation with the PPF, and showed the application of this idea for linear-parabolic lane detection and tracking on IPM-images. The algorithm succeeded in tracking with many challenge scenes, which shows

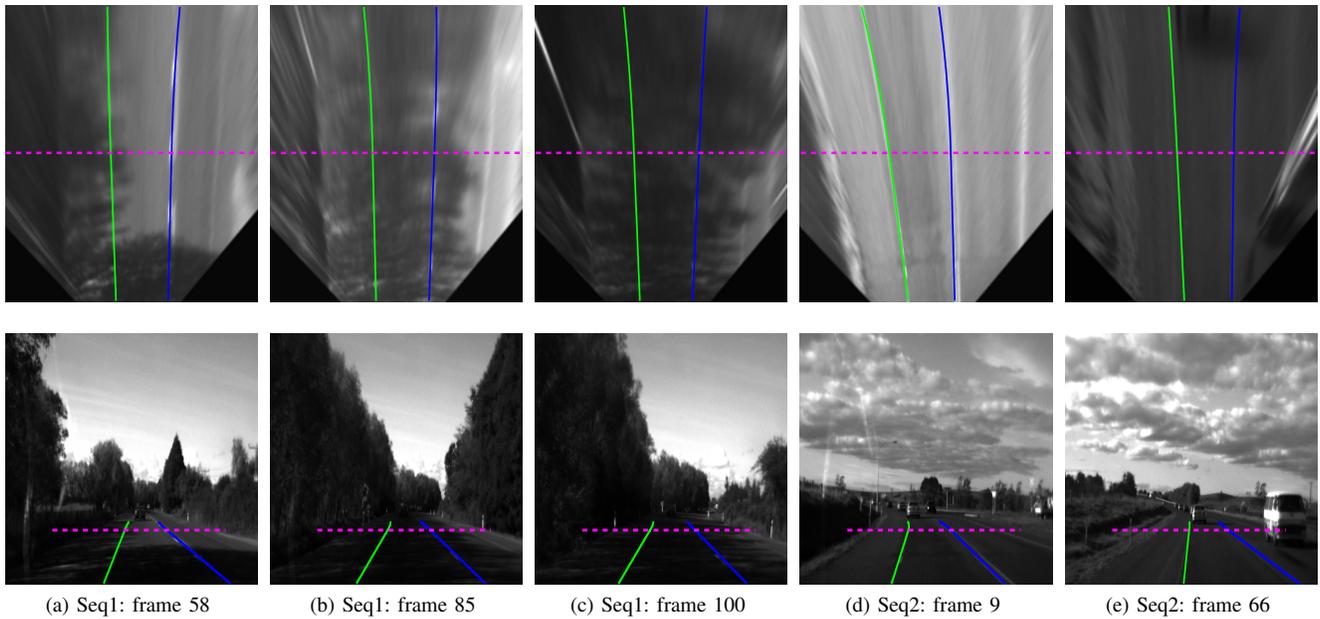


Fig. 8: Tracking results on challenging situations, e.g., strong illumination artifacts (Seq1) and brightness differences (Seq2).

that our algorithm is very robust concerning challenging scenes.

The advantages of our algorithm, as compared to the state of the art in this field, are threefold. First, we do not only use position and color information of the pixels, but also employ gradient information in the multi-kernel density to improve performance. Second, we estimate the linear-parabolic lane model in a hierarchical fashion. This reduces computation time and enhances the robustness of the algorithm. Third, the PPF maintains multiple hypotheses of lane state and can recover from failure by introducing initial particles, whereas the KF and least-squares method do not support this feature.

Future work will require testing our algorithm on more and larger databases, as well as optimizing computation using faster programming approaches. This effort is justified since the suggested combination of methods can be used to estimate also other shapes like circles, ellipses, and conics. Thus, our novel algorithm can become useful also in other fields where tracking of curved lines is important and its application is not limited to the driving scenario.

#### REFERENCES

- [1] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, Mar. 2006.
- [2] C. R. Jung and C. R. Kelber, "A robust linear-parabolic model for lane following," in *Proc. 17th Brazilian Symposium on Computer Graphics and Image Processing*, Oct. 17–20, 2004, pp. 72–79.
- [3] R. Danescu and S. Nedevschi, "Probabilistic lane tracking in difficult road scenarios using stereovision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 272–282, June 2009.
- [4] K. H. Lim, K. P. Seng, L.-M. Ang, and S. W. Chin, "Lane detection and kalman-based linear-parabolic lane tracking," in *Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on*, vol. 2, 26–27 2009, pp. 351–354.
- [5] Y. Zhou, R. Xu, X. Hu, and Q. Ye, "A robust lane detection and tracking method based on computer vision," *Measurement Science and Technology*, vol. 17, no. 4, pp. 736–745, 2006.
- [6] B. Southall and C. Taylor, "Stochastic road shape estimation," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, 2001, pp. 205–212 vol.1.
- [7] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *International Journal of Computer Vision*, vol. 39(1), pp. 57–71, 2000.
- [8] R. Dahyot, "Statistical hough transform," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 8, pp. 1502–1509, Aug. 2009.
- [9] T. Bergener and C. Bruckho, "Compensation of non-linear distortions in inverse-perspective mappings," Institut für Neuroinformatik, Ruhr-Universität Bochum, Tech. Rep., April 1999.
- [10] M. Bertozzi and A. Broggi, "Real-time lane and obstacle detection on the gold system," in *Proc. IEEE Intelligent Vehicles Symposium*, Sept. 19–20, 1996, pp. 213–218.
- [11] G. Liu, F. Wörgötter, and I. Markelić, "Combining statistical hough transform and particle filter for robust lane detection and tracking," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, jun. 2010, pp. 993–997.
- [12] X. Liu, Q. Song, and P. Li, "A parabolic detection algorithm based on kernel density estimation," in *5th International Conference on Intelligent Computing*, 2009, pp. 405–412.
- [13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [14] S. Duffner, J.-M. Odobez, and E. Ricci, "Dynamic partitioned sampling for tracking with discriminative features," in *Proceedings of the British Machine Vision Conference*, September 2009.
- [15] K. Smith and D. Gatica-perez, "Order matters: a distributed sampling method for multi-object tracking," in *In Proc. BMVC*, 2004.
- [16] <http://www.mi.auckland.ac.nz/EISATS>.
- [17] <http://www.youtube.com/watch?v=NSS5EzGu4Lw>.