Monocular Multi-Kernel Based Lane Marking Detection

Wenjie Lu, Sergio Alberto Rodriguez Florez, Emmanuel Seignez, Roger Reynaud

Monocular Multi-Kernel Based Lane Marking Detection

Wenjie Lu[1,2], Sergio A. Rodriguez F.[1,2], Emmanuel Seignez[1,2] and Roger Reynaud[1,2]

Abstract—**Lane marking detection provides key information for scene understanding in structured environments. Such information has been widely exploited in Advanced Driving Assistance Systems and Autonomous Vehicle applications. This paper presents an enhanced lane marking detection approach intended for low-level perception. It relies on a multi-kernel detection framework with hierarchical weights. First, the detection strategy performs in Bird's Eye View (BEV) space and starts with an image filtering using a cell-based blob method. Then, lane marking parameters are optimized following a parabolic model. Finally, a self-assessment process provides an integrity indicator to improve the output performance of detection results. An evaluation using images from a public dataset confirms the effectiveness of the method.**

I. INTRODUCTION

On-vehicle lane marking detection plays an important role in Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles (AV). In this context, this function constitutes a key part of low-level perception, which is intended to help scene understanding [1] by estimating the shape and the localization of the lane markings. Such information can be exploited within more complex systems, for instance, Automatic Cruise Control [2], drowsiness detection systems [3], and vehicle localization systems [4].

This subject has been widely studied for decades and recent remarkable progresses have been summarized in [5]. Most of existing lane detection approaches follow a common strategy composed of three stages: sensor type, feature extraction, and model fitting.

Sensor type. Different kinds of vision systems are used within the state-of-the-art. Fisheye cameras [6] are well suited for multi-marking detection thanks to wide capture range. However, the marking shapes are deformed because of distortion, making lane fitting a complex task. Stereo vision systems [7] grant access to dense 3D data where road structure, such as road curbs, can be detected. Laser-based approaches [8] provide a clear signature (i.e. reflectiveness) of the lanes invariant to light changes. This kind of sensor, however, is rarely adopted because of its high integration cost. Compared to laser sensors, monocular vision remains a mature technology at a lower cost. The sensitivity of vision systems to light changes can be alleviated by image processing algorithms as stated in [9]. In this study, a facing-forward camera on board of a vehicle is adopted.

Feature extraction. Feature extraction methods enfold lane marking descriptors, as well as different techniques employed to discriminate potential lane markings. Color is the most common property for marking detection, but color based descriptors do not deal with light changes. Color-gradient based methods are better suited to complex outdoor conditions. For instance, Histogram of Oriented Gradients (HOG) can be used to describe pixel features in [10]. Other approaches rely on multi-cue fusion for increasing robustness in advance. For example, in [11], color, color gradient and road shape are combined to model a confidence in a particle filter. In this paper, second order derivative filter and cell-based blob algorithm inspired from [12] are considered.

Model fitting. Several geometry models in BEV space have been proposed: straight lines [13], parabolic curves [14], and splines [9]. In semi-urban environments, the use of a naive straight line model is not appropriate, since only a few part of the road scenarios can be correctly fitted. Curve models and multi-models fit the majority of lane markings in real road conditions, however the increasing model complexity implies a higher computation time. A parabolic model is a trade-off between model complexity and time cost in our method.

Different fitting methods have been studied intended to retrieve high precision and low computation time. For instance, straight lines are fitted by Hough transform in [15]. A multi-kernel based framework is adopted to determine all the three parameters of a parabola model together in [16]. At first, the "distance" between a single pixel and a specific parabola is described as a probability density function. The probability distribution of the parabolic model is then obtained by integrating the probability density function. This multi-kernel based method is introduced in our method and improved in two aspects. For one thing, a parametric Gaussian model based initialization is proposed to provide a prior guess of the parabola's zero order component. This initialization step greatly reduces the computation time because only two parameters are optimized in multi-kernel method. For the other, hierarchical weights are introduced into probability density function, in order to ensure robustness.

To improve the output performance of marking detection method, a self-assessment indicator is proposed. This indicator describes quality of a detected marking. An indicator threshold is set according to Receiver Operating Characteristic (ROC) curve to determine a qualified detection.

The remainder of this paper is structured as in Fig. 1. Section II introduces image processing stage. Section III explains parameter estimation stage. Section IV presents experimental results. The paper is concluded in Section V.

1. Université Paris-Sud, 91405, Orsay, FRANCE

2. Institut Fondamentale d'Electronique CNRS 8622, 91405, Orsay, FRANCE

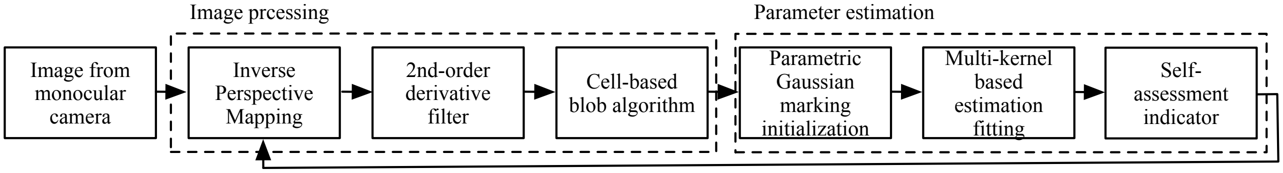
Image prcessing
Parameter estimation
Image from monocular camera
Inverse Perspective Mapping
2nd-order derivative filter
Cell-based blob algorithm
Parametric Gaussian marking initialization
Multi-kernel based estimation fitting
Self-assessment indicator

Fig. 1. Paper structure.

II. IMAGE PROCESSING

In image processing part, a second-order derivative filter is utilized to extract marking features in BEV space at first, and a cell-based blob algorithm is employed to eliminate potential outlier features.

A. Inverse Perspective Mapping

The road images are captured by a monocular camera, which is modeled using a pinhole projective model assuming no distortion and zero skew. The images are processed through Inverse Perspective Mapping (IPM), from perspective space to BEV space. This transformation provides vertical and paralleled lane markings in BEV space, which greatly facilitates marking detection strategies. In addition, BEV images are stabilized in case of distortion caused by road slopes, according to [17]. At first, the detected parabolic markings of last frame are approximated to two straight line markings, which is intended to estimate the vanishing point. The derived vanishing point assists to obtain adjusted pitch and yaw angle.

B. Second-order Derivative filter

A second order derivative filter along the horizontal direction is applied to process BEV images where lane markings are nearly vertical. This filter is presented as follows:

$$I_{xx} = G_x * G_x * I, \tag{1}$$

where I is input image, $G_x = [+1, 0, -1]$ is horizontal Sobel descriptor, I_{xx} is the filtered image.

Second-order derivative is able to detect “black-white-black” transition, if the width of the “white” part is less than 5 pixels. Assuming that a marking width is between 10-35 cm in reality, the width of projected marking in BEV image is 1-4 pixel, with the resolution 0.1m/pixel. So second-order derivative filter is able to detect the marking pixels in BEV images. In addition, time consuming of this method is much less than complex extraction methods such as CANNY.

C. Cell-based blob algorithm

To eliminate the outliers included in the image, a cell-based blob algorithm inspired from [12], is introduced. As depicted in Fig. 2, the image is divided into 4×4 cells at first, then all the blobs in the image are searched. The blob directions are classified into 9 bin levels (20° per bin). So every blob has its own bin level. All the bin levels in a cell can vote for the main bin level of the current cell, as shown in Fig. 2(b). All the cells' main bin levels can vote for the main bin level of the entire image. Meanwhile, a bin level template, as depicted in Fig. 2(c), is constructed offline according to various detected marking results. This template represents possible bin levels in different cells. For instance, as in Fig 2(a), left-heading markings appear only at the left side part of a BEV image, and is unable to appear at the right side part. In the end, outlier blobs are excluded according to three conditions: the main bin level of the image, the bin level template, and the size of blob. The output result is shown in Fig. 2.

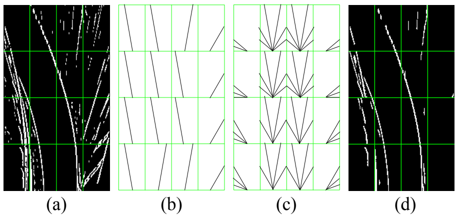

Fig. 2. Blob algorithm. (a) input binary image, (b) voted bin levels in each cell, (c) bin level template, (d) output image.

By the end of image processing part, a binary image is produced. White pixels represent lane marking candidates. Considering the complexity and diversity of outdoor traffics, some noise pixels still exist in the image. The next section explains approaches to estimate lane marking parameters from a $M \times N$ binary image with a few noise pixels.

III. PARAMETER ESTIMATION

A parabola: $x = c + d \cdot y + e \cdot y^2$ is chosen as marking model in BEV space. On one side, the marking shape variation is restricted because of IPM range in our method, so a parabola is more suitable than complex models (i.e. spline-like model). On the other side, a straight line model, which appears frequently in real situations, is included in parabola model. The marking initialization step determines the zero order component c. The first and second order components d and e are estimated through an improved multi-kernel based method with hierarchical weights.

A. Marking Initialization

This part aims to determining the zero order components c_l and c_r of left and right markings together using a parametric Gaussian model based method. To this end, the intersections of both lane markings at x-axis in BEV images are estimated using multi-cues. These two intersections are exactly model parameters c_l and c_r. First, a Region Of Interest (ROI) I_{ROI} with the size of $M_{ROI} \times N$, is cropped from the binary image, as shown in Fig. 3(a). The length M_{ROI} should be set carefully. The markings cannot be approximated as straight lines if M_{ROI} is too long, the discontinuous markings do not appear in I_{ROI} if M_{ROI} is too short. Then the following distributions are considered: previous detection distribution X_1, white pixels distribution X_2, Hough lines distribution X_3, prior data distribution X_4 and lane width distribution

X_5. The normalized distributions $\overline{X_1}$ to $\overline{X_5}$ are shown as in Fig. 3(c)-(d).

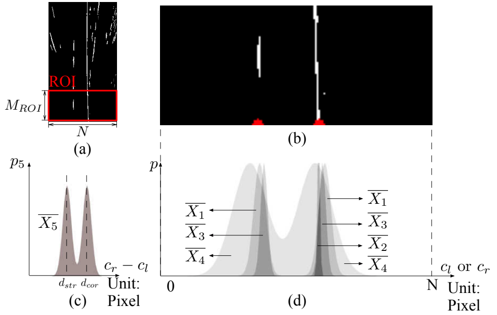

Fig. 3. Parametric Gaussian model based Initialization. (a) ROI, (b) amplified ROI, (c) lane width distribution $\overline{X_5}$, (d) Gaussian distributions $\overline{X_1}$ to $\overline{X_4}$ corresponding to (b). Estimated results c_l^* and c_r^* are depicted as two red dot in (b).

Normally, in BEV space the intersections move smoothly along the first row of I_{ROI} if the input frame frequency is high enough. So the positions of both intersections are strongly related to the previous position c_l^{pre} and c_r^{pre}. Distribution X_1 is therefore represented as:

$$X_1 \sim 1/2 \cdot \left[N\left(c_l^{pre}, \sigma_1^2\right) + N\left(c_r^{pre}, \sigma_1^2\right)\right]. \tag{2}$$

White pixels at the bottom row of I_{ROI} are also potential candidates. The white pixels distribution is:

$$X_2 \sim 1/n_w \cdot \left[N\left(\mu_{w,1}, \sigma_2^2\right) + \cdots + N\left(\mu_{w,n_w}, \sigma_2^2\right)\right], \tag{3}$$

where $\mu_{w,1}, \cdots, \mu_{w,n_w}$ are locations of the white pixels along x-axis.

Considering that dashed markings may have no white pixel cues, for instance the left marking in Fig. 3(a), Hough transform is helpful to find the intersections of these discontinuous markings. Let $\mu_{ht,1}, \cdots, \mu_{ht,n_{ht}}$ denote the intercepts of these Hough lines on x-axis, the Hough line distribution is:

$$X_3 \sim 1/n_{ht} \cdot \left[N\left(\mu_{ht,1}, \sigma_3^2\right) + \cdots + N\left(\mu_{ht,n_{ht}}, \sigma_3^2\right)\right]. \tag{4}$$

The prior distribution $N\left(\mu_{4,l}, \sigma_{4,l}^2\right)$ and $N\left(\mu_{4,r}, \sigma_{4,r}^2\right)$ are fitted offline according to numerous historical data which is deinfed as:

$$X_4 \sim 1/2 \cdot \left[N\left(\mu_{4,l}, \sigma_{4,l}^2\right) + N\left(\mu_{4,r}, \sigma_{4,r}^2\right)\right]. \tag{5}$$

The lane width distribution is represented as:

$$X_5 \sim 1/2 \cdot \left[N\left(d_{str}, \sigma_5^2\right) + N\left(d_{cor}, \sigma_5^2\right)\right], \tag{6}$$

where d_{str} and d_{cor} denote the average lane width of straight sections and corner sections respectively. The probability distribution of c_l and c_r is then given as

$$p_t(c_l, c_r) = \left[\sum_{j=l,r}\left(\sum_{i=1}^{3} k_i \cdot p_i\left(c_j\right)\right) \cdot p_4(c_j)\right] \cdot p_5(c_l - c_r), \tag{7}$$

where $p_i(x)$ is probability density of $\overline{X_i}$, k_i is a coefficient of $p_i(x)$. The probability distribution $p_t(c_l, c_r)$ of the input image Fig. 3(a) is shown in Fig. 4.

Finally the intersections of left and right lane markings c_l^* and c_r^* , are determined as follows:

$$c_l^*, c_r^* = \underset{c_l, c_r}{\arg\max}\, p_t(c_l, c_r). \tag{8}$$

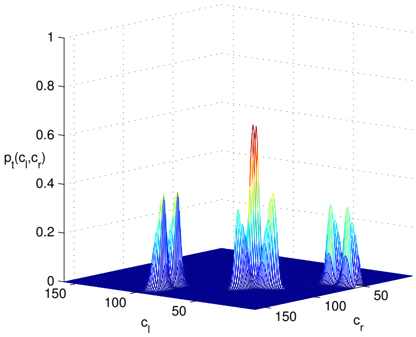

Fig. 4. The probability distribution of $p_t(c_l, c_r)$.

B. Multi-Kernel Based Estimation with Hierarchical Weights

A multi kernel density based method with hierarchical weights is introduced to estimate the model parameters. The basic descriptor of this algorithm is the similarity between an image pixel (x_i, y_i) with the slope $\tan\theta_i$ and the model (c, d, e), presented as:

$$G_{pi}(c, d, e, x_i, y_i) = \int_{-\infty}^{+\infty} K'_x K_y K'_\theta dy, \tag{9}$$

where

$$K_y = \frac{1}{\sqrt{2\pi\sigma_{yi}^2}} exp\left(-\frac{(y - y_i)^2}{2\sigma_{yi}^2}\right) \tag{10}$$

$$K'_x = \frac{1}{\sqrt{2\pi\sigma_{xi}^2}} exp\left(-\frac{(c + dy + ey^2 - x_i)^2}{2\sigma_{xi}^2}\right) \tag{11}$$

$$K'_\theta = \frac{1}{\sqrt{2\pi\sigma_{\theta i}^2}} exp\left(-\frac{(atan(-2ey - d) - \theta_i)^2}{2\sigma_{\theta i}^2}\right). \tag{12}$$

GaussHermite quadrature method [18] is employed to compute the numerical solution of G_{pi}. When G_{pi} is derived, the probability of a specified model $p_{Gpi}(d, e)$ is defined as:

$$p_{Gpi}(d, e) = \frac{1}{n_t}\sum_{i=1}^{n_{total}} w(x_i, y_i) \cdot G_{pi}(c_m, d, e, x_i, y_i), \tag{13}$$

where n_t is the total number of white pixels in the image. $w(x_i, y_i)$ is a hierarchical weight, which offers corresponding coefficient according to different areas of an BEV image. The assignment of $w(x_i, y_i)$ is depicted in Fig. 5. In the figure, the current lane markings are more likely to appear in darker areas than in lighter areas, therefore, the pixels in darker areas are valued with a higher weight. Relatively the effect of noise pixels (e.g. vehicles on the road) in light areas are alleviated, with a lower weight.

The left and the right marking probabilities are computed together as in equation (14) to obtain the optimized parameters d_l^*, e_l^*, d_r^*, e_r^*, with a geometrical constraint presented in equation (15) and (16).

$$d_l^*, e_l^*, d_r^*, e_r^* = \underset{d_{j_l}, d_{j_r}, e_{j_l}, e_{j_r}}{\arg\max}[p_{Gpi}(d_{i_l}, e_{j_l}) + p_{Gpi}(d_{i_r}, e_{j_r})]. \tag{14}$$

$$d_{j_r} \in (d_{j_l} - \triangle d, d_{j_l} + \triangle d) \tag{15}$$

$$e_{j_r} \in (e_{j_l} - \triangle e, e_{j_l} + \triangle e). \tag{16}$$

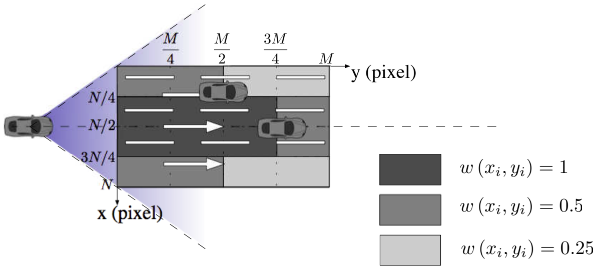

Fig. 5. Assignment of hierarchical weight $w(x_i, y_i)$ in BEV space.

Therefore, c_l^* and c_r^* are estimated in the initialization step, and d_l^*, e_l^*, d_r^*, e_r^* are optimized using multi-kernel method. The current lane markings are detected.

C. Self-assessment Indicator

When a marking in a frame is estimated, a self-assessment indicator is designed to qualify the detection. To this end, a confidence $conf$ is employed to measure a detected marking. At first, a set Φ_c of pixels with higher contribution to the detected markings (c^*, d^*, e^*) are defined as follows:

$$\Phi_c = \{(x_i, y_i) | G_{pi}(c^*, d^*, e^*, x_i, y_i) \geq G_{thres}\}, \tag{17}$$

where $i = 1, ..., n_{con}$, G_{thres} is confidence threshold, n_{con} is the total number of high contributed pixels in the image.

Then the frame is divided into L average blocks along y-axis. The set Φ_c^l of pixels in l^{th} block is defined as:

$$\Phi_c^l = \left\{(x_i, y_i) | (x_i, y_i) \in \Phi_c, \frac{M}{L} l \leq y_i < \frac{M}{L}(l+1)\right\}, \tag{18}$$

where $l = 0, 1, ..., L-1$, $i = 1, ..., n_{con,l}$, $n_{con,l}$ is the number of high contributed pixels in l^{th} block.

When the number of high contributed pixels in all the blocks are derived, the confidence of a detection can be defined as:

$$conf = \frac{n_{con}}{L} \sum_{l=1}^{L} p_{con,l}, \tag{19}$$

where

$$p_{con,l} = \begin{cases} \frac{L \cdot n_{con,i}}{n_{con}} & \frac{n_{con,i}}{n_{con}} \leq \frac{1}{L} \\ \frac{L \cdot n_{con,i}}{(1-L) \cdot n_{con}} + \frac{L}{L-1} & else \end{cases}. \tag{20}$$

In confidence $conf$, both the number of contributed pixels (n_{con}/L in equation (19)) and the distribution of these pixels ($\sum_{l=1}^{L} p_{con,l}$ in equation (19)) are considered. A detection result which has more contributed pixels and whose contributed pixels are more equally distributed earns a higher confidence.

With confidence $conf$ and an optimized threshold T_{conf}^*, the detected markings can be classified. If both left and right detections are qualified, accept these two detections, and apply local parameter estimation in the next frame. If one detection is qualified while the other is not, accept the qualified detection and estimate the detection on the other side according to the qualified result, meanwhile apply local parameter estimation in the next frame. If neither detection is qualified, reject both of them and estimate the lane markings according to previous detections, and apply global parameter estimation in the next frame. The estimation of T_{conf}^* is introduced in IV-B.

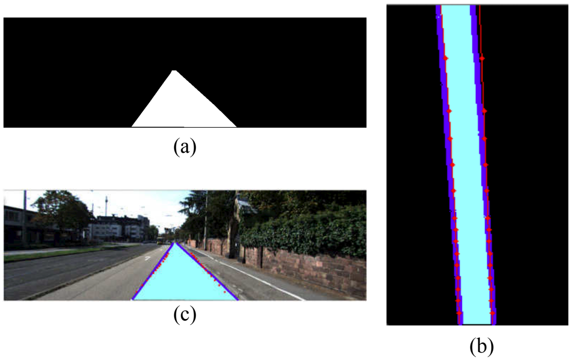

Fig. 6. The procedures from road surface benchmarks to lane marking benchmarks. (a) road surface benchmark in perspective space. (b) road surface benchmark (cyan), reference markings (blue) and the markings (red) in BEV space. (c) road surface benchmark (cyan), reference markings (blue) and detected markings (red) in perspective space.

IV. RESULTS

A. Ground truth from KITTI database

The lack of a standard and unified evaluation environment is one of the most critical problems in quantifying the performance of marking detection, as stated in [5]. To assess our result in a public dataset and provide a universal evaluation for comparison, we make use of the road surface ground truth frames included in KITTI database [19], enriched with a lane marking annotation presented below. The ground truth dataset addresses three different road types: urban unmarked (UU), urban marked two-way road (UM), and urban marked multi-lane road (UMM). UM images are chosen to evaluate our method because the boundaries of road surface benchmark in this type represent exactly the reference markings. The annotation procedure is illustrated in Fig. 6. When a benchmark image is inputed as Fig. 6(a), the benchmark road surface is projected into BEV space as cyan area in Fig. 6(b). Then the left and the right boundaries of the road surface are considered as a pair of reference markings, which are shown as blue pixels in Fig. 6(b). For contrast, a pair of markings detected by our method are depicted as red pixels in Fig. 6(b) as well. Fig. 6(c) illustrates the road surface benchmark (cyan), the reference markings (blue) and the detected markings (red) in perspective image. When road surface is occluded by objects in some UM images, the road surface is annotated manually.

B. Indicator Threshold Estimation

One usage of the reference markings is to derive the optimized indicator threshold T_{conf}^*. To this end, a confusion matrix is constructed as Table I, including the required confusion matrix elements: true positive (TP), false positive (FP), true negative (TN) or false negative (FN).

After that, True Positive Rate $TPR = \frac{TP}{TP+FN}$ and False Positive Rate $FPR = \frac{FP}{FP+TN}$ with different T_{conf} are calculated, shown as the green stars in Fig. 7. A function

		self-assessment indicator	
		$conf \leq T_{conf}$	$conf > T_{conf}$
with ground truth	$MAE > 4pxl$	TN	FP
	$MAE \leq 4pxl$	FN	TP

TABLE I
DEFINITION OF CONFUSION MATRIX ELEMENTS

$y(x) = (a^x - b^x)/(a - b)$ is fitted as ROC curve according to the sample points, depicted as red curve in Fig. 7. The optimized point is obtained as the nearest point on ROC curve to the perfect classification point $(0, 1)$ [20], depicted as the red dot in Fig. 7. Optimized threshold of self-assessment indicator T^*_{conf} is estimated corresponding to the optimized point. This estimation process is executed offline.

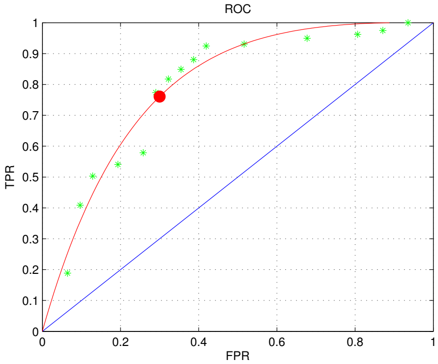

Fig. 7. ROC curve. Green stars are samples, red curve is the fitted exponential function, and the red dot is the optimized point.

C. Marking detection

This algorithm has been tested on 12 different challenging scenarios in KITTI database, including more than 1900 frames. Fig. 8 shows several detected lane markings in various outside conditions. The image processing area is marked within blue lines. The pink pixels are the lane marking candidates after image processing. The green pixels are the corresponding contributed pixels in Φ_{con} from (17). The detected lane markings are depicted as red dots in the image processing area. In Fig. 8(a), the number of candidate pixels for the right lane is limited because of the strong contrast near tunnel entrance, but the the constraint of parallel markings ensures both detections. Fig. 8(b) shows the detection of dashed markings, which is more difficult than continuous marking detection in practice. Marking initialization step helps this detection, because the zero order components of marking model is estimated before multi-kernel fitting. The detection in heavy traffic conditions are shown as Fig. 8(c), thanks to the cell-based blob algorithm, numerous outlier pixels caused by vehicles are eliminated. Fig. 8(d) depicts the detection result in a corner, the parabolic model fits the markings. Fig. 8(e) and (f) are more complicated road situations including curves, multi markings, heavy traffic or irregular shadows, our approach derives detection results in these environments. Fig. 8(g) shows an special condition, the primitive result (purple dots) is detected, but this result is rejected by self-assessment indicator, a pair of estimated markings (yellow dots) are adopted as output result. The results above perform the marking detection ability of the proposed method in highway and road scenes.

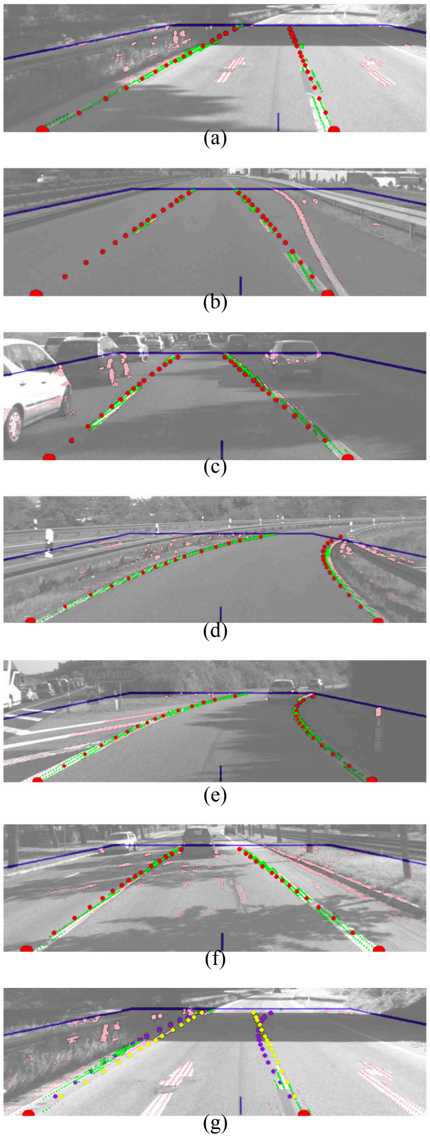
(a)
(b)
(c)
(d)
(e)
(f)
(g)

Fig. 8. Example detection results within various outdoor conditions.

The average processing time per frame is 22.7ms. About 98% of the frames adopt local parameter estimation (7 scales for both model parameter d and e). About 2% of the frames apply global parameter estimation (40 scales for both model parameter d and e). The maximum time consuming in global detections is 62ms. According to the time cycle of KITTI database frames (100ms), the proposed method is adequate to process the frame streams in real-time.

Meanwhile, two lane detection algorithms, Method A [9] and Method B [16], are introduced to compare with the proposed method. In Method A, selective oriented Gaussian filters are applied as image processing strategy. RANSAC is then used to fit Bezier Splines. All the lane markings in the frame are detected. Method A and our method use totally different marking models and fitting methods. The contrast of these two methods is regarded as the comparison between methods with different frameworks.

TABLE II
METHODS COMPARISON USING KITTI DATASET.

	Time	Correct	MAE	Success
	(second)	(frame)	(pixel)	ratio (%)
Method A [9]	0.0210	117	12.84	61.58
Method B [16]	1.5580	113	11.84	59.47
Proposed method	0.0489	161	4.38	84.74

In Method B, a multi-kernel based framework is introduced to determine parameters of a parabolic model. Method B and our method have the same multi-kernel estimation framework. Different from Method B, our method proposes a parametric Gaussian initialization step, and hierarchical weights. The comparison between Method B and our method can imply the effect of the improvements in our methods.

The source code of Method A is provided by the author. Only the IPM parameters of KITTI experiment environment are set before the program runs. The code for Method B are programed according to [16]. Source codes of all three methods are run using KITTI benchmark images. Obtained markings from detection methods are compared to reference markings which are obtained from 95 benchmark images in IV-A. Comparison results of the three methods are reported in Table II, where **Time** is the run time per frame, **Correct** is the number of correct markings (a marking's Mean Absolute Error (MAE) is less than 4 pixels in BEV space is defined as a correct detection), **MAE** means the MAE of all the detected markings, and **Success ratio** is the percentage of correct detections in all the detections.

The run time of Method A is 0.0210s, which is close to the average time 0.02s mentioned in [9]. Some detections of Method A are markings from neighborhood lane or road barriers, rather than the current lane marking. So the success ratio of Method A is relative low. The contrast shows that our method has a higher success ratio, but is slower than method A. Compared to Method B, our method has a less time cost, and higher success ratio. This comparison indicates that the initialization step and hierarchical weights help to reduce average run time and improve marking detection performance. Considering that the benchmark images we used are not continuous, global detection is applied on all the images, the computation time of our method in Table II is an average time of global detection. In addition, self-assessment indicator is not able to provide estimated detections in discontinuous frames. The performance of our method can be improved in continuous scenarios.

V. CONCLUSIONS

This paper demonstrates a real-time, reliable, and precise lane marking detection method applied in real traffic conditions, thanks to an improved multi-kernel based estimation method with hierarchical weights. A second-order derivative filter is proposed to extract lane marking pixels. The time cost is less than complex extraction method. A cell-based blob algorithm is introduced to alleviate outlier pixels (i.e. heavy traffic, disturbed markings and shadows). A parametric Gaussian model based initialization is used to estimate zero order model parameters, which greatly reduces the time consuming of curve fitting. A multi-kernel based parameter estimation with hierarchical weights is recommended to obtain the rest model parameters. In the end, a confidence indicator is applied to improve the qualification of output markings.

REFERENCES

[1] S. A. Rodríguez F, V. Fremont, P. Bonnifait, and V. Cherfaoui, "Multimodal object detection and localization for high integrity driving assistance," *Machine Vision Applications*, vol. 25, no. 3, pp. 583–598, April 2014.

[2] B. Kim and K. Yi, "Probabilistic states prediction algorithm using multi-sensor fusion and application to smart cruise control systems," in *Proc. IEEE Intelligent Vehicles Symp.*, June 2013, pp. 888–895.

[3] X. Li, E. Seignez, and P. Loonis, "Reliability-based driver drowsiness detection using dempster-shafer theory," in *Proc. IEEE Conf. on Control Automation Robotics & Vision*, December 2012, pp. 1059–1064.

[4] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Proc. IEEE Intelligent Vehicles Symposium*, June 2013, pp. 449–454.

[5] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

[6] S. Li and Y. Shimomura, "Lane marking detection by side fisheye camera," in *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems*, September 2008, pp. 606–611.

[7] V. Pradeep, G. Medioni, and J. Weiland, "Piecewise planar modeling for step detection using stereo vision," in *Workshop on Computer Vision Applications for the Visually Impaired*, 2008.

[8] A. von Reyher, A. Joos, and H. Winner, "A lidar-based approach for near range lane detection," in *Proc. IEEE Intelligent Vehicles Symp.*, June 2005, pp. 147–152.

[9] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intelligent Vehicles Symp.*, June 2008, pp. 7–12.

[10] A. Linarth and E. Angelopoulou, "On feature templates for particle filter based lane detection," in *Proc. IEEE Conf. on Intelligent Transportation Systems*, October 2011, pp. 1721–1726.

[11] N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symp.*, June 2003, pp. 558–563.

[12] R. Jiang, R. Klette, T. Vaudrey, and S. Wang, "Lane detection and tracking using a new lane model and distance transform," *Machine Vision and Applications*, vol. 22, no. 4, pp. 721–737, June 2011.

[13] S. Vacek, S. Bergmann, U. Mohr, and R. Dillmann, "Rule-based tracking of multiple lanes using particle filters," in *Proc. IEEE conf. on Multisensor Fusion and Integration for Intelligent Systems*, September 2006, pp. 203–208.

[14] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, March 2006.

[15] M. Javadi, M. Hannan, S. Samad, and A. Hussain, "A robust vision-based lane boundaries detection approach for intelligent vehicles," *Information Technology Journal*, vol. 11, no. 9, pp. 1184–1192, 2012.

[16] G. Liu, F. Worgotter, and I. Markelic, "Lane shape estimation using a partitioned particle filter for autonomous driving," in *Proc. IEEE International Conf. on Robotics and Automation*, May 2011, pp. 1627–1633.

[17] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *Proc. IEEE Intelligent Vehicles Symp.*, June 2007, pp. 315–320.

[18] X. Liu, Q. Song, and P. Li, "A parabolic detection algorithm based on kernel density estimation," in *Emerging Intelligent Computing Technology and Applications*, 2009, pp. 405–412.

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, September 2013.

[20] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.