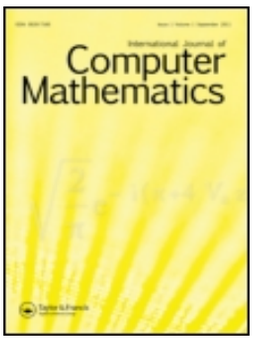

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: https://www.tandfonline.com/loi/gcom20

# Extraction of lane markings using orientation and vanishing point constraints in structured road scenes

Weirong Liu, Shutao Li & Xu Huang

**To cite this article:** Weirong Liu, Shutao Li & Xu Huang (2014) Extraction of lane markings using orientation and vanishing point constraints in structured road scenes, International Journal of Computer Mathematics, 91:11, 2359-2373, DOI: 10.1080/00207160.2013.813020

**To link to this article:** https://doi.org/10.1080/00207160.2013.813020

Accepted author version posted online: 18 Jun 2013.
Published online: 01 Aug 2013.

Submit your article to this journal

Article views: 128

View related articles

View Crossmark data

Citing articles: 4 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=gcom20

*International Journal of Computer Mathematics*, 2014
Vol. 91, No. 11, 2359–2373, http://dx.doi.org/10.1080/00207160.2013.813020

Taylor & Francis
Taylor & Francis Group

# Extraction of lane markings using orientation and vanishing point constraints in structured road scenes

Weirong Liu[a,b], Shutao Li[a,b]* and Xu Huang[c]

[a]*The State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, People's Republic of China;* [b]*College of Electrical and Information Engineering, Hunan University, Changsha 410082, People's Republic of China;* [c]*Faculty of Education, Science, Technology & Mathematics, University of Canberra, Bruce, Australian Capital Territory 2601, Australia*

(*Received 25 January 2013; revised version received 30 March 2013; accepted 4 June 2013*)

Extraction of lane markings is still a challenge due to the poor quality of lane markings and interferences from external circumstances. In this paper, we utilize line segments as low-level features to detect lane markings on structured road scenes. Our novel algorithm can be highlighted in four items as follows. Firstly, a road surface region is reasonably located using adaptive segmentation method, and then most of the interferences from the external circumstances are eliminated by Laplacian filter. Secondly, the line segments detected by the line segment detector are applied to represent the structural information of lanes. Thirdly, non-lane candidate markings are removed through orientation and vanishing point constraints. Finally, the lanes are accurately extracted from the remaining candidate lane marking. Experimental results on complex structured road scenarios in urban streets are shown and the effectiveness and robustness of our novel algorithm are underpinned by the experimental results.

**Keywords:** lane detection; Laplacian filter; line segment detector; *K*-means clustering; intelligent vehicle

*2000 AMS Subject Classifications*: 68T45; 62H30; 68U10

## 1. Introduction

Lane boundary detection techniques are used to recognize painted lane markings or boundaries from clutter road scenes and then to produce an accurate estimation of the lane boundary positions. As a crucial element in intelligent vehicles, lane detection plays an important role in driver assistance systems [3].

Since we have to deal with stained lane markings, clutter shadows, various illumination conditions and other interferences from the external circumstances at the same time, vision-based lane detection is still a challenge. Many different types of sensor have been proposed for lane markings detection over the last decades, such as Global Positioning System [4], laser scanner [25], radar [1] and visual camera [2,3,5,6,10,11,14–17,19,20,23,24]. With the development of digital image processing technology, vision-based lane detection and recognition algorithms have been an active research area [17]. Up to now, various lane markings extraction approaches using vision sensors have been developed. Broadly, these techniques can be divided into two categories: the model-based approach and the feature-based approach.

Since the geometry relationships of lanes on road surface are normally assumed as straight-line model or curve model [2,14,23], the model-based approaches are robust against noise and

*Corresponding author. Emails: shutao_li@yahoo.com.cn, shutao_li@hnu.edu.cn

© 2013 Taylor & Francis

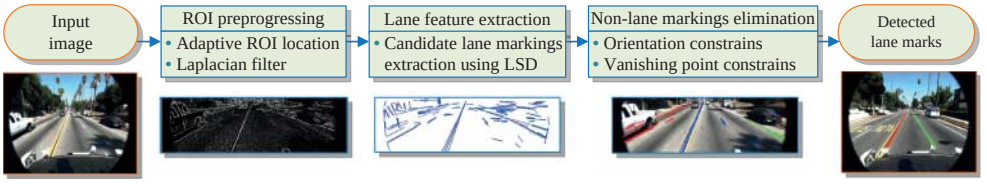


Figure 1. Flowchart of the proposed algorithm.

missing data [24]. But the model constructed for a scene may fail in other scenes, which makes the model-based approach less adaptive. Hence, the accuracy rate of lane detection depends on the compatibility between the model and actual road scenes.

Based on the low-level features such as edges [16], texture [19], colour [6,11] and frequency domain features [15], the feature-based methods are efficient and adequate effective. However, the feature-based methods are often sensitive to road conditions, lighting changes and shadows, since features are traditionally extracted at the pixel level [5].

In most cases, the shape of lanes on structured roads can be approximated by well-defined straight lines or some short straight lines [20]. Therefore, the key steps for lane markings extraction from structured roads based on the model-based approach or the feature-based approach are as follows. Firstly, the inverse perspective mapping (IPM) is applied to remove the perspective effect in the input image sequences. Then, the commonly used methods for line detection techniques are the random sample consensus (RANSAC) algorithm and the Hough transform (HT) [3,10] and its variations [9,18,20,21,26]. Due to the strict flat road assumption in IPM and plenty of multiplications and trigonometric operations to each pixel in HT on rapidly changing scenarios, these methods have high false detection rate and computational cost.

In order to improve validity of lane markings detection and reduce the computation complexity, we present a novel effective and robust algorithm for lane detection in this paper. The main steps involved in our scheme are summarized in Figure 1.

The main novelties and contributions of our algorithm include in three aspects. Firstly, we reasonably segment road surface region using the adaptive segmentation method and then use the Laplacian filter to remove most of the interferences from the external circumstances. Secondly, candidate lane markings in rapidly changing complex scenarios are extracted by the line segment detector (LSD) algorithm in [22], which lead to high efficiency with a low false detection rate. Finally, unlike the traditional model-based approach, we take the line segments as low-level features to analyse geometry information of lanes on structured road scenes. Therefore, less prior knowledge (only including orientation and vanishing point constraints) is necessary for the lane detection. As a result, our algorithm is more adaptive and efficient to complex scenes compared with other methods.

The remainder of this paper is organized as follows. Section 2 presents the effective lane detection algorithm in detail. Experimental results on complex structured road scenarios from real-world data sets are provided in Section 3 and Section 4 concludes this paper.

## 2. Lane detection algorithm

### 2.1 *Preprocessing operations of region of interest*

#### 2.1.1 *Segmentation of ROI*

In general, all candidate lane markings lie in the middle region of the image captured from the vision camera. In order to save computation times and reduce interferences from the external

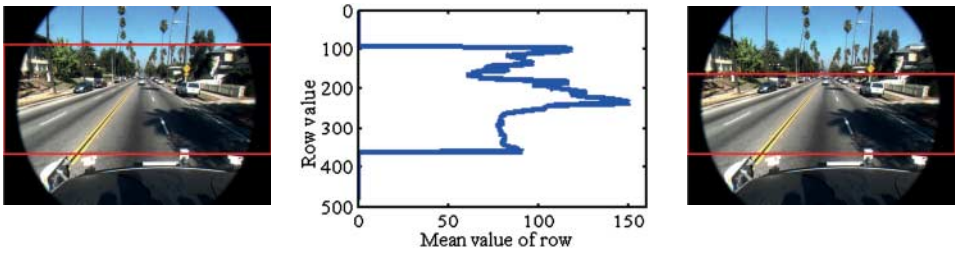


Figure 2. Results of ROI segmentation. (a) The rough ROI within the red rectangle box is extracted by some priors, (b) the mean value of each row in rough ROI and (c) the ROI within the red rectangle box is extracted by the adaptive ROI location algorithm (colour online only).

circumstances such as impacts from sky, buildings, trees etc., the non-road surface region at the top of input image should be eliminated.

Let $\mathbf{I}_c \in \mathbb{R}^{m\times n}$ be an image captured from vision camera, where $m$ and $n$ are the number of rows and columns in $\mathbf{I}_c$, respectively. Based on prior knowledge, the upper boundary of the region of interest (ROI) is determined by a fixed ratio and the lower boundary of the ROI is located at the position of the front bumper. Hence, we can easily extract the rough ROI $\mathbf{I} \in \mathbb{R}^{m_r\times n}$, where $m_r$ is the number of rows of $\mathbf{I}$. As shown in Figure 2(a), a rough ROI, $\mathbf{I}$, marked by the red rectangle box covers all candidate lane markings.

Although the fixed ratio and fixed position segmentation methods mentioned above can decrease the sizes of the region to be processed, there are still non-road surface scenes in the rough ROI $\mathbf{I}$. In order to precisely capture upper boundary of $\mathbf{I}$ in dynamic scenes with vehicle moving, we analyse the relationships between the upper boundary and mean value of each row of $\mathbf{I}$. From the mean value of each row of $\mathbf{I}$ shown in Figure 2(b), we can see that the reasonable upper boundary of $\mathbf{I}$ locates at a row with minimum mean value. Therefore, we propose an adaptive ROI location algorithm as follows.

Firstly, the mean value $\bar{\mathbf{I}}_i$ of each row in $\mathbf{I}$ can be calculated by the following equation:

$$\bar{\mathbf{I}}_i = \frac{\sum_{j=1}^{n} \mathbf{I}_{ij}}{n} \quad (i = 1, \ldots, m_r). \tag{1}$$

Secondly, row value $d$ with a minimal mean value can be calculated by the following equation:

$$d = \mathrm{E}(\min(\bar{\mathbf{I}}_i)) \quad (i = 1, \ldots, m_r), \tag{2}$$

where the operator E extracts the row value with minimal mean value $\min(\bar{\mathbf{I}}_i)$.

Finally, the reasonable ROI $\mathbf{I}_a \in \mathbb{R}^{(m_r-d+1)\times n}$ is extracted by the following equation:

$$\mathbf{I}_a = \mathbf{I}_{ij} \quad (i = d, \ldots, m_r, j = 1, \ldots, n). \tag{3}$$

If the adaptive ROI segmentation algorithm mentioned above fail in few complex scenes, such as a large shadow region lies on the end of road surface, the reasonable ROI $\mathbf{I}_a$ becomes set as $\mathbf{I}$.

Compared with the rough ROI $\mathbf{I}$ in Figure 2(a) extracted by some priors, the reasonable ROI $\mathbf{I}_a$ within the red box in Figure 2(c) extracted by the adaptive ROI segmentation algorithm is more accurate. As a consequence, a more accurate ROI will reduce computational cost and eliminate interferences from the non-road surface region.

Figure 3. Results of ROI filtered by the Laplacian filter. The images in the first row are the ROI $\mathbf{I}_a$, and the images in the second row show the ROI $\mathbf{I}_d$ filtered by Laplacian.

### 2.1.2 *ROI filtering*

Edge distribution function [7], Gaussian filter [2], Canny filter [13] and Sobel filter [12] are used to detect the candidate lane boundaries in the ROI, but there are a large number of short line segments from regions with rapid intensity change in the filtered images. As a result, these short line segments will frequently lead to false lane detection results. While the Laplacian filter is able to accurately detect the candidate lane boundaries in the ROI, the filtered results from regions with rapid intensity change appear as irregular noises. Hence, we utilize the Laplacian filter to extract the edges of multiple types of the lane markings and remove the interferences from shadows, writings and road marks.

Since the Laplacian filter in Equation (4) is a two-dimensional isotropic measure of the second spatial derivative of an image, the Laplacian operator on an image highlights regions of rapid intensity change and is usually used for edge detection:

$$\nabla^2 = \frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2}. \tag{4}$$

The discrete Laplacian in the following equation is used to filter the reasonable ROI $\mathbf{I}_a$:

$$\mathbf{I}_d = \nabla^2 \otimes \mathbf{I}_a, \tag{5}$$

where $\mathbf{I}_d$ is the filtered ROI, $\otimes$ is the convolution operation and

$$\nabla^2 = \frac{4}{(\alpha+1)} \begin{bmatrix} \frac{\alpha}{4} & \frac{(1-\alpha)}{4} & \frac{\alpha}{4} \\ \frac{(1-\alpha)}{4} & -1 & \frac{(1-\alpha)}{4} \\ \frac{\alpha}{4} & \frac{(1-\alpha)}{4} & \frac{\alpha}{4} \end{bmatrix}$$

is the discrete Laplacian.

From the filtered ROI $\mathbf{I}_d$ shown in the second row of Figure 3, we can see that the edges of lane markings are accurately extracted and the interferences from clutter shadows are removed.

## 2.2 *Candidate lane markings extraction using LSD*

Line segments can be used as low-level features to capture the structural information of images and help solving several problems in image analysis and computer vision [22]. Traditional line segment detection methods for lane markings extraction are the RANSAC algorithm [8] and the linear HT and its variations, such as the randomized HT [26], hierarchical additive HT [20], and HT with graphical techniques [21].

Since each pixel in image is applied with the trigonometric and multiplications operations and the orientation information is ignored, the disadvantages of the line detection methods based on HT

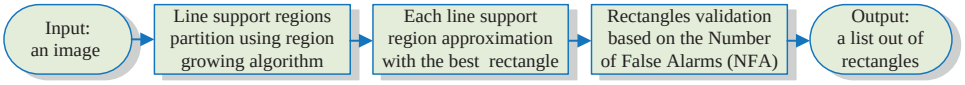


Figure 4. The key steps of the LSD parameters.

are high computational complexity and false detection. Meanwhile, the disadvantages of RANSAC techniques for irregular curve fitting are similar to the disadvantages of HT methods [14].

Based on the best approximation with the rectangle and the expected number of false alarms (NFA) constraint, a fast LSD algorithm in [22] produces accurate results and controls the number of false detections at the same time. The key steps of the LSD algorithm are illustrated in Figure 4.

The LSD algorithm starts at forming a level-line field by the region growing method and then segments the image into line-support regions according to the same level-line angle up to a certain tolerance. After that, in order to find the most similar rectangle of each line-support region, the main orientation of a rectangle is set as the principal inertial axis of line-support region and the size of the rectangle is chosen to cover the line-support region. Finally, each rectangle is validated by the contrario approach and the Helmholtz principle, and then the number of false detections is controlled by the NFA of a rectangle.

There are three parameters $\rho$, $\tau$ and $\varepsilon$ involved in LSD, where $\rho = 5.2$ is the threshold on the gradient magnitude, $\tau = 22.5$ is the angle tolerance used for line-support regions search and $\varepsilon = 1$ is a non-critical detection parameter. The parameter settings mentioned above are supported by tests on thousands of images of very different kinds and origins [22].

The LSD algorithm is used to fast extract line segments from the ROI $\mathrm{F}_d$, and line segments as low-level features are used to analyse and detect candidate lane markings on structured road scenes. These procedures can be formulated by the following equation:

$$S = \mathrm{LSD}(\mathbf{I}_d), \tag{6}$$

where $S = \{s_1, s_2, \ldots, s_k\}$ is a set of line segments extracted from the ROI $\mathbf{I}_d$, and the operator LSD is the lines extraction algorithm shown in Figure 4.

Each line segment $\mathrm{s}_i$ $(i = 1, \ldots, k)$ is defined as

$$s_i = \{x_i^1, y_i^1, x_i^2, y_i^2, \theta_i\} \quad (i = 1, \ldots, k), \tag{7}$$

where $(x_i^1, y_i^1)$ and $(x_i^2, y_i^2)$ are the starting and ending point coordinates of the line segment $s_i$, respectively. $\theta_i \in [-90°, 90°]$ is the angle of the line segment $s_i$ and is calculated by the following equation:

$$\theta_i = \frac{180}{\pi} \times \arctan\left(\frac{y_i^2 - y_i^1}{x_i^2 - x_i^1}\right) \quad (i = 1, \ldots, k). \tag{8}$$

As shown in the first row of Figure 5, the test images collected from different scenarios are used to evaluate the LSD algorithms. There are different types of lanes in test images, such as straight lanes and dash lanes under the circumstance with shadows, writings, road marks and moving vehicles. The line segments extracted by the HT and LSD are shown in the second row and the third row of Figure 5, respectively.

As shown in the first row of Figure 5, the quality of boundaries detected by the HT method is poor, and even some obvious traces on the road are missed by the HT method. From the second row of Figure 5, we can easily get that the LSD algorithm accurately detects the full boundaries of different lanes in the ROI $\mathbf{I}_d$, and the geometric structure of ROI $\mathbf{I}_d$ is well expressed by the detected line segments $S$. Since the region growing method in the LSD is sensitive to noise, a long line segment in a noisy region may be cut into two or more short line segments. But these small line segments will be clustered or eliminated according to orientation and vanishing point constraints which will be discussed as follows.

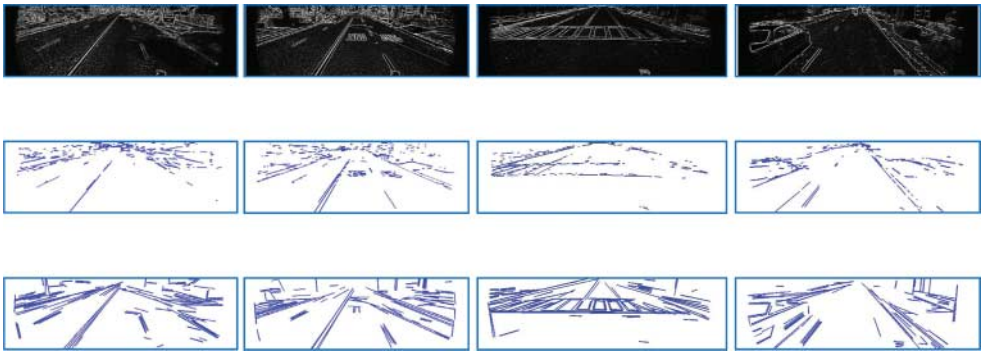

Figure 5. Comparison of line segments detection results. The first row: the ROI $\mathbf{I}_d$ filtered by Laplacian. The second row: line segments detection results using HT. The third row: line segments detection results using LSD.

### 2.3 *Non-lane markings removed by orientation and vanishing point constraints*

Vehicles on the road usually run on a relative fixed lane following the traffic rules, and the lane markings appear on a captured image with limited directions as shown in Figure 5. We observed that lane markings in the left of ROI $\mathbf{I}_d$ have an orientation angle between $-20°$ and $-70°$, and lane markings in the right of ROI $\mathbf{I}_d$ have an orientation angle between $20°$ and $70°$, which will be discussed in detail in Section 3.2. Hence, the line segments at the left (right) of ROI with an orientation angle out of $-20°$ to $-70°$ (or $20°$ to $70°$) are the non-lane markings. In order to remove irrelevant line segments that represent non-lane markings, the lines $S$ are filtered by the following equations:

$$S^{\mathrm{L}} = \left\{ s_i^{\mathrm{L}} \mid x_i^1 < \frac{n}{2}, x_i^2 < \frac{n}{2}, \theta_i \in [-20°, -70°] \right\} \quad (i = 1, \ldots, p), \tag{9}$$

$$S^{\mathrm{R}} = \left\{ s_i^{\mathrm{R}} \mid x_i^1 \geq \frac{n}{2}, x_i^2 \geq \frac{n}{2}, \theta_i \in [20°, 70°] \right\} \quad (i = 1, \ldots, q), \tag{10}$$

where $p$ and $q$ are the numbers of retained line segments $S^{\mathrm{L}}$ in the left of ROI $\mathbf{I}_d$ and $S^{\mathrm{R}}$ in the right of ROI $\mathbf{I}_d$.

As shown in the third row of Figure 5, the non-lane markings included in the line segments $S$ with blue are removed according to Equations (9) and (10), and the retained line segments $S^{\mathrm{L}}$ with green and $S^{\mathrm{R}}$ with red are drawn in Figure 6. Comparing the line segments $S$ with the retained line segments $S^{\mathrm{L}}$ and $S^{\mathrm{R}}$, we can see that the candidate lane marks reserve in $S^{\mathrm{L}}$ or $S^{\mathrm{R}}$ and non-lane markings from road outside are removed.

Moreover, in order to extract the valuable lanes for driver assistance applications, we should extract the line segments in $S^{\mathrm{L}}$ or $S^{\mathrm{R}}$ close to vehicle and remove the line segments in $S^{\mathrm{L}}$ or $S^{\mathrm{R}}$ relatively far from vehicle. Based on direction constraint, the $k$-means algorithm is utilized to partition the retained lines $S^{\mathrm{L}}$ or $S^{\mathrm{R}}$ into two categories.

Given a set of line segments $S^{\mathrm{L}} = \{s_1^{\mathrm{L}}, s_2^{\mathrm{L}}, \ldots, s_p^{\mathrm{L}}\}$ (or $S^{\mathrm{R}} = \{s_1^{\mathrm{R}}, s_2^{\mathrm{R}}, \ldots, s_q^{\mathrm{R}}\}$), $k$-means algorithm aims to divide the $p$ (or $q$) line segments into two sets $S^{\mathrm{L}} = \{S_1^{\mathrm{L}}, S_2^{\mathrm{L}}\}$ (or $S^{\mathrm{R}} = \{S_1^{\mathrm{R}}, S_2^{\mathrm{R}}\}$) and

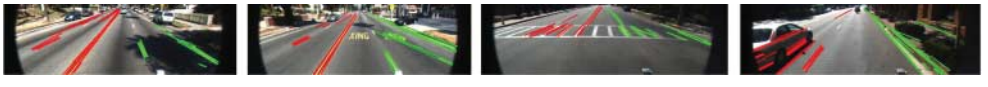

Figure 6. Non-lane markings removed by orientation constraints (9) and (10) (colour online only).

Figure 7. Non-lane markings removed by $k$-means clustering (colour online only).

minimize the within-cluster sum of squares, that is,

$$\arg\min_{S^{\mathrm{L}}} \sum_{i=1}^{2} \sum_{s_j^{\mathrm{L}} \in S_i^{\mathrm{L}}} \|s_j^{\mathrm{L}} - \mu_i^{\mathrm{L}}\|^2 \quad (j = 1, \ldots, p), \tag{11}$$

$$\arg\min_{S^{\mathrm{R}}} \sum_{i=1}^{2} \sum_{s_j^{\mathrm{R}} \in S_i^{\mathrm{R}}} \|s_j^{\mathrm{R}} - \mu_i^{\mathrm{R}}\|^2 \quad (j = 1, \ldots, q), \tag{12}$$

where $\mu_i^{\mathrm{L}}$ $(i = 1, 2)$ is the mean value of points in $S_i^{\mathrm{L}}$ $(i = 1, 2)$ and $\mu_i^{\mathrm{R}}$ $(i = 1, 2)$ is the mean value of points in $S_i^{\mathrm{R}}$ $(i = 1, 2)$.

The line segments $S_{\mathrm{C}}^{\mathrm{L}}$ (or $S_{\mathrm{C}}^{\mathrm{R}}$) are chosen by the following equations because the lanes close to a vehicle are more useful than the lanes relatively far from the vehicle in driver assistance applications:

$$S_{\mathrm{C}}^{\mathrm{L}} = \{S_i^{\mathrm{L}} \mid \mu_i^{\mathrm{L}} = \max(\mu_1^{\mathrm{L}}, \mu_2^{\mathrm{L}})\} \quad (i = 1, 2), \tag{13}$$

$$S_{\mathrm{C}}^{\mathrm{R}} = \{S_i^{R} \mid \mu_i^{\mathrm{R}} = \min(\mu_1^{\mathrm{R}}, \mu_2^{\mathrm{R}})\} \quad (i = 1, 2). \tag{14}$$

As shown in Figure 7, we can see that all lanes close to a vehicle are accurately captured by the line segments $S_{\mathrm{C}}^{\mathrm{L}}$ and $S_{\mathrm{C}}^{\mathrm{R}}$ in blue. In addition, interferences from the characters and irregular patches close to a vehicle are reserved in the line segments $S_{\mathrm{C}}^{\mathrm{L}}$ and $S_{\mathrm{C}}^{\mathrm{R}}$ at the same time. The lane marks intersect at vanishing point, but the interferences from the characters and irregular patches may appear in any direction. Therefore, the line segments indicating interferences are removed using vanishing point as follows.

Let $V = (x_{\mathrm{v}}^1, y_{\mathrm{v}}^1, x_{\mathrm{v}}^2, y_{\mathrm{v}}^2)$ defines the vanishing point and $V$ is calculated by

$$V = G(S_{\mathrm{C}}, \mathbf{I}_{\mathrm{V}}), \tag{15}$$

where $S_{\mathrm{C}} = \{S_{\mathrm{C}}^{\mathrm{L}}, S_{\mathrm{C}}^{\mathrm{R}}\}$ is the reserved line segments from Equations (11)–(14), the operator $G$ extrapolates $S_{\mathrm{C}}$ across the ROI $\mathbf{I}_{\mathrm{V}}$ and selects the point with maximum intersection number. Since vanishing point $V$ usually lies in the top of image $\mathbf{F}_{\mathrm{C}}$, the ROI $\mathbf{I}_{\mathrm{V}}$ is set as $\mathbf{I}_{\mathrm{V}} = \mathbf{I}_{\mathrm{C}}^{(m/4)\times n}$ to reduce computational times.

Finally, we retain the final line segments $S_{\mathrm{V}}^{\mathrm{L}}$ (or $S_{\mathrm{V}}^{\mathrm{R}}$) in the left of ROI $\mathbf{I}_d$ (or in the right of ROI $\mathbf{I}_d$) by

$$S_{\mathrm{V}}^{\mathrm{L}} = \{s_{\mathrm{V}_i}^{\mathrm{L}} \mid \|s_{\mathrm{V}_i}^{\mathrm{L}} - V\| < \delta\} \quad (i = 1, \ldots, u), \tag{16}$$

$$S_{\mathrm{V}}^{\mathrm{R}} = \{s_{\mathrm{V}_i}^{\mathrm{R}} \mid \|s_{\mathrm{V}_i}^{\mathrm{R}} - V\| < \delta\} \quad (i = 1, \ldots, w), \tag{17}$$

where $\delta$ is the tolerance between the line segment and the vanishing point.

We plot the reserved lines $S_{\mathrm{V}}^{\mathrm{L}}$ and $S_{\mathrm{V}}^{\mathrm{R}}$ with purple and mark the selected vanishing point with the red '+' in Figure 8. Comparing with the red line segments in Figure 7 (or the green line segments in Figure 8) relatively far from a vehicle, we can see that the remaining purple line segments $S_{\mathrm{V}}^{\mathrm{L}}$ and $S_{\mathrm{V}}^{\mathrm{R}}$ covers all lane markings close to the vehicle.

Figure 8. Non-lane markings removed by vanishing point constraints (colour online only).

### 2.4 *Lane extraction*

Although the non-lane markings are eliminated by orientation and vanishing point constraints in Section 2.3, the left lane $L_{\mathrm{L}}$ (or the right lane $L_{\mathrm{R}}$) is represented by some line segments $S_{\mathrm{V}}^{\mathrm{L}}$ (or $S_{\mathrm{V}}^{\mathrm{L}}$). The final step in our algorithm is the extraction and fitting the left lane $L_{\mathrm{L}}$ (or the right lane $L_{\mathrm{R}}$) as follows:

$$L_{\mathrm{L}} = \{L_{\mathrm{L}_i} \mid \min(x_{V_i}^{L_1}), \max(y_{V_i}^{L_1}), \max(x_{V_i}^{L_2}), \min(y_{V_i}^{L_2}), \mathrm{mean}(\theta_{V_i}^{L})\} \quad (i = 1, \ldots, u), \tag{18}$$

$$L_{\mathrm{R}} = \{L_{\mathrm{R}_i} \mid \max(x_{V_i}^{\mathrm{R}_1}), \max(y_{V_i}^{\mathrm{R}_1}), \min(x_{V_i}^{\mathrm{R}_2}), \min(y_{V_i}^{\mathrm{R}_2}), \mathrm{mean}(\theta_{V_i}^{\mathrm{R}})\} \quad (i = 1, \ldots, w), \tag{19}$$

where $\min(\mathrm{x}_{\bullet}^{\bullet})$ (or $\max(\mathrm{x}_{\bullet}^{\bullet})$) is the minimum (or maximum) of horizontal coordinate in all candidate lines $S_{\mathrm{V}}^{(\bullet)}$, $\max(y_{(\bullet)}^{(\bullet)})$ (or $\max(\mathrm{y}_{(\bullet)}^{(\bullet)})$) is the minimum (or maximum) of vertical coordinate in all candidate lines $S_{\mathrm{V}}^{(\bullet)}$ and $\mathrm{mean}(\theta_{(\bullet)}^{(\bullet)})$ is the mean of the angle in all candidate lines $S_{\mathrm{V}}^{(\bullet)}$.

## 3. Experiments

### 3.1 *Experimental setting*

We have completed the proposed algorithm in MATLAB R2010b without optimal and conducted the experiments on a PC with Intel Core 2 i3-2010 at 3.3 GHz. The correct detection rate, false detection rate and missing detection rate are used to assess the objective performance of the lane detection. Meanwhile, for the convenience of qualitative validation performance of the proposed algorithm, we paint the final detected left lanes $L_{\mathrm{L}}$ and right lanes $L_{\mathrm{R}}$ in red and green, respectively, and superimpose the detected lanes $L_{\mathrm{L}}$ and $L_{\mathrm{R}}$ on the original image.

Because the research on lane detection tends to be proprietary [3] and different complex scenes in the real world cannot be represented by a data set, standardized training and testing data sets are not available. In order to evaluate the performance of the proposed method on a structured road, first we use video sequences with different types structured road of urban streets from Caltech lane data sets in [2]. The Caltech lane data sets contain four colour video sequences totalling 1225 frames with size of $640 \times 480$, which include challenging traffic scenarios, such as different pavement types, passing cars, stained writings, lots of shadows and curvatures. Moreover, we collect two colour video sequences totalling 787 frames with size of $640 \times 480$ from urban roads in Changsha and use these data sets to test the robustness of the proposed method under different weather conditions.

### 3.2 *Discussion of parameters setting*

As discussed in Section 2, there are two parameters in our proposed algorithm, namely the orientation angle range $\theta$ in the orientation constraint and tolerance $\delta$ in the vanishing point constraint. We design a few experiments to find the optimal values for the accurate lane detection. The former 100 test images from 'washington1' clip in Caltech lane data sets are collected for these experiments, because there are lot of clutter shadows and various illumination conditions in the 100 images.

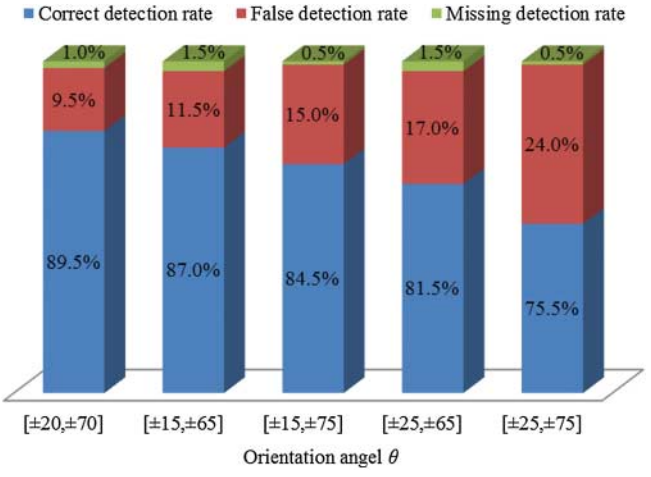


Figure 9. Effects of different orientation angles on the proposed method (colour online only).

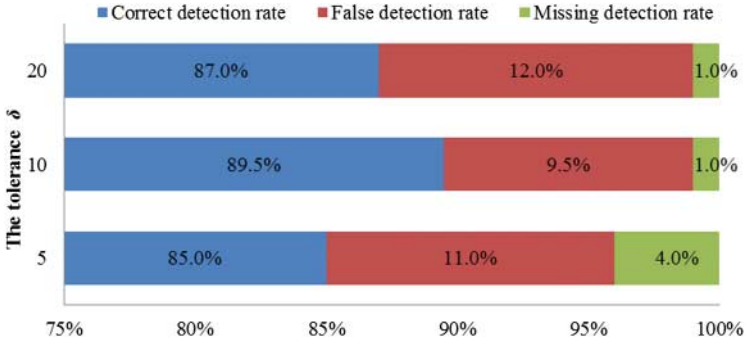


Figure 10. Effects of different tolerance on the proposed method (colour online only).

Firstly, we conduct experiments on the 100 test images with five orientation angle ranges $\theta$ of [±20,±70], [±15,±65], [±25,±75], [±25,±65] and [±25,±75], and then effects of different orientation angle range to performance of the proposed method are illustrated in Figure 9. Comparing with results from angle ranges $\theta \in [\pm 20, \pm 70]$, we observed that the correct detection rate decreases whether the orientation angle range $\theta$ increased or decreased. The reason for these results is that all lanes close to the vehicle lie in angle ranges $\theta \in [\pm 20, \pm 70]$. Furthermore, the correct detection rate from the orientation angle range $\theta$ increased is better than the correct detection rate from the orientation angle range $\theta$ decreased. When angle range $\theta$ increases, the lanes far from the vehicle included in the candidate line segments are satisfied with the vanishing point constraint and then lead to the false detection rate increased. On the contrary, when angle range $\theta$ decreases, the interferences from the road surface may not satisfy with the vanishing point constraint, and then the correct detection rate decreases slowly. Therefore, the angle range is reasonably set as $\theta \in [\pm 20, \pm 70]$.

Secondly, changing the tolerance $\delta$ values between the line segment and the vanishing point affects the performance of the proposed algorithm. The effects of the three different tolerance values are shown in Figure 10. Comparing with results from the tolerance $\delta = 5$ and $\delta = 20$, it is clear that the correct detection rate with the tolerance $\delta = 10$ is the best. Meanwhile, we can see that the missing detection rate increases from 1% to 4% with values of the tolerance $\delta$ increased from 5 to 20. The reason behind these results is that small values of the tolerance $\delta$ results in the

correct candidate line segments eliminated by the vanishing point constraint. Hence, the tolerance is set as $\delta = 10$, which leads to more robust results.

### 3.3 *Results*

After the parameters have been discussed above, we implement the experiments not only on washington1, washington2, cordova1 and cordova2 clips in Caltech lane data sets, but also on Changsha1 and Changsha2 clips in Changsha data sets collected by ourselves. The experimental results are presented in this section.

Since line segments as low-level features, our algorithm successfully extracts geometry information of lanes on structured road scenes with almost no or a little interference. Therefore, the left (highlighted in red) and right (highlighted in green) straight lane marks are completely detected as shown in the first row of Figure 11.

From the results shown in the second row of Figure 11, combining direction constraint with the *k*-means algorithm, we can observe that our algorithm can filter the interferences from the rutted tracks, stop lines, zebra stripes, lettering markings and irregular patches, and then the left and right straight lane marks close to the vehicle are accurately captured.

The results in the third row of Figure 11 prove that our algorithm also works well in the scenarios such as with messy shadows, moving vehicles and arrows indication markings, and the boundary lines of the lanes are all correctly detected by the proposed method. These results should be attributed to the interferences elimination by the vanishing point constraint.

As shown in Figure 11, we can see that all scenarios in Caltech lane data sets are under sunny weather conditions with good illumination. In order to prove the effectiveness of our algorithm under different weather conditions, we conduct the experiments on Changsha1 (under cloudy weather conditions) and Changsha2 clip (under rainy weather conditions), and then present the lane detection results in Figure 12.

Although cloudy weather will cause poor illumination conditions, the candidate lane boundaries with low contrast in ROI are enhanced by the Laplacian filter and then interferences from the isolation belts, broken marks, rhombus warning marks or vehicles ahead are filtered by direction

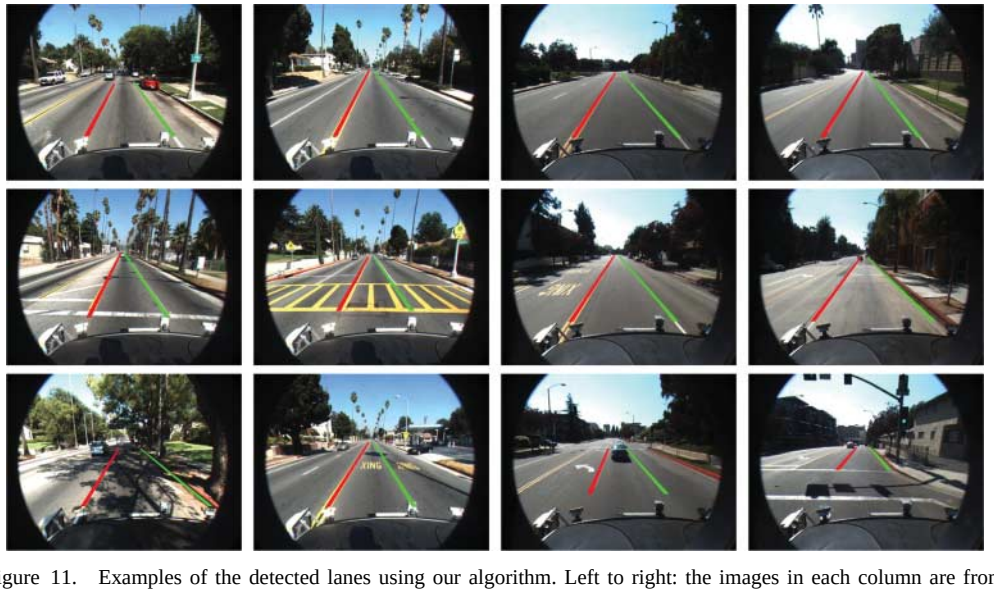

Figure 11. Examples of the detected lanes using our algorithm. Left to right: the images in each column are from Washington1, Washington2, Cordova1 and cordova2 clips, respectively (colour online only).

Figure 12. Examples of the detected lanes using our algorithm. The images in the first row are from Changsha1 clip and the images in the second row are from Changsha2 clip.

Table 1. The quantitative evaluation results for all clips in Caltech lane and Changsha data sets using our algorithm.

| Clips | Number of frames | Correct detection rate (%) | False detection rate (%) | Missing detection rate (%) | Computation time (s)/frame |
|---|---|---|---|---|---|
| Washington1 | 337 | 92.88 | 6.82 | 0.30 | 0.22 |
| Washington2 | 232 | 94.61 | 4.96 | 0.43 | 0.21 |
| Cordova1 | 250 | 95.80 | 3.80 | 0.40 | 0.19 |
| Cordova2 | 406 | 93.47 | 5.91 | 0.62 | 0.20 |
| Changsha1 | 331 | 95.17 | 4.38 | 0.45 | 0.20 |
| Changsha2 | 456 | 93.20 | 5.92 | 0.88 | 0.21 |
| Average | 2012 | 94.01 | 5.44 | 0.55 | 0.21 |

and vanishing point constraints. Hence, our algorithm accurately extracts the straight lane marks under cloudy weather conditions as shown in the first row of Figure 12.

Furthermore, interferences under rainy weather conditions such as pedestrians, tail light flares, indication marks, rutted watermarks caused by wheel rolling and passing cars are also eliminated by the direction and vanishing point constraints. The results in the second row of Figure 12 prove the effectiveness of our algorithm under rainy weather conditions.

Although some qualitative experimental results are presented in Figures 11 and 12, we use the correct detection rate, false detection rate and missing detection rate to quantitatively evaluate performance of our proposed algorithm on data sets in the real world.

The quantitative results for four clips totally have 2012 colour frames in all data sets, which are summarized in Table 1. The second column shows the numbers of frames in each clip, and the indexes for quantitative evaluation of detection results using our algorithm are listed in the third column to the fifth column. The efficiency of the proposed algorithm is assessed by the average computation time listed in the sixth column.

We can see that the average correct detection rate for all 2012 frames in Caltech lane data sets achieve at 94.01%, with a low average false detection rate and an average missing detection rate less than 0.55%. Meanwhile, the average computation time for our algorithm without optimization on the testing platform mentioned above is less than 0.21 s per image frame.

Note that results for the cordova1 clip achieve the highest correct detection rate, since the scenarios with interference in cordova1 clip are less than other clips. On the contrary, because there are clutter shadows, various illumination conditions, and other interferences from external circumstances at same time, we can easily get that false detection rate for the washington1 clip is higher than quantitative results for washington2, cordova1 and cordova2 clips. Nevertheless, the average correct detection rate for washington1 clip archives at 92.88% with a missing detection

rate less than 0.30%, which demonstrate that the candidate lane markings extraction algorithm using LSD is robust to the complex structured road scenarios. Moreover, since the strict flat road assumption in other methods is not necessary in our algorithm, we can see that our algorithm also works well in scenarios with a different slope of road in cordova2 clip.

Due to reflection of road surface under rainy weather condition, some lane markings far away from the car are invisible, which lead to the missed detection rate for Changsha2 clip is the highest in all experimental results. Comparing with correct detection rates of clips in Caltech lane data sets, we can see that the correct detection rates of results from Changsha1 and Changsha2 clips keep good even under cloudy or rainy weather conditions.

As shown in Table 1, we can see that the average computation time and the correct detection rate are proportional to the complexity of road scenarios. Even there are lots of interferences in washington1 clip, the average computation times are less than 0.21 s per image frame with higher resolution ($640 \times 480$), which demonstrate the efficiency of our algorithm.

Therefore, the qualitative and quantitative experimental results for all cases shown in Figures 11 and 12 and Table 1 clearly demonstrate the potential and practical utility of our algorithm for lane detection under different structured scenarios.

Furthermore, we compare our algorithm with well-known lane detection approach based on model [2] so as to prove the effect of our algorithm based on line segment features. The detection results [2] on four clips are available at http://www.vision.caltech.edu/malaa/research/iv08. We illustrate the different performance of our algorithm and Aly's algorithm [2] in Figures 13–15. Based on the RANSAC algorithm for the fitting Bezier splines model, Aly's algorithm achieves good results on detecting two boundary modes.

But we can see that Aly's algorithm misses detection of some lane boundaries due to the clutter shadows in Figure 13(e), lettering markings in Figure 13(f), short lanes in Figure 13(g) and slope roads in Figure 13(h). Since these complex scenes are not satisfied with the assumptions of model in [2], and then lead to missed detection of lanes. Compared with the assumptions of Aly's model, less prior knowledge (only including orientation and vanishing point constraints) are necessary in our algorithm. As a result, our algorithm successfully detects all boundaries as shown in Figures 13(a)–(d) and is more adaptive to complex scenes.

In addition, since messy interferences or inadequate lane markings may match the model in [2], Aly's algorithm produces false detection of some lane boundaries as shown in Figures 14(e)–(h).

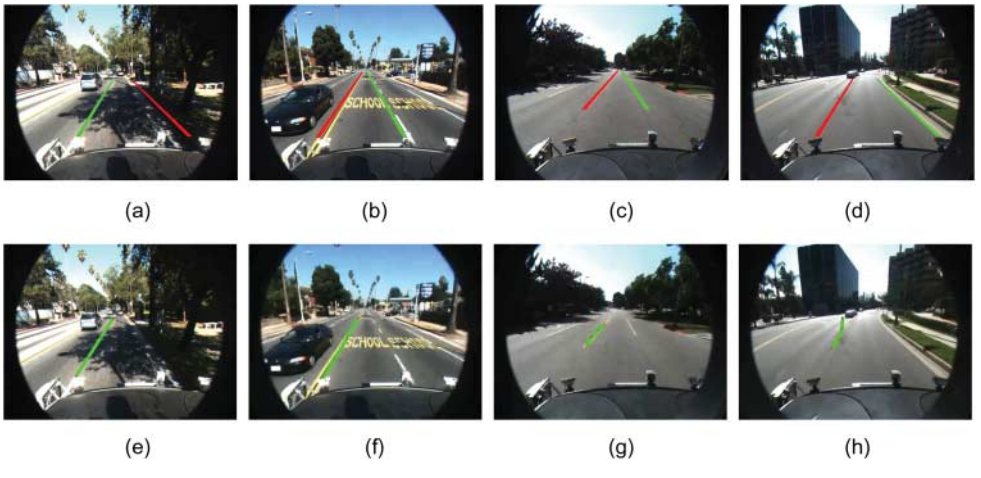


Figure 13. Experimental results comparison of our algorithm and Aly's algorithm [2]. Upper row: correct detection results using our algorithm. Lower row: missing detection results using Aly's algorithm. Left to right: the images in each row are from washington1 clip, washington2 clip, cordova1 clip and cordova2 clip, respectively.

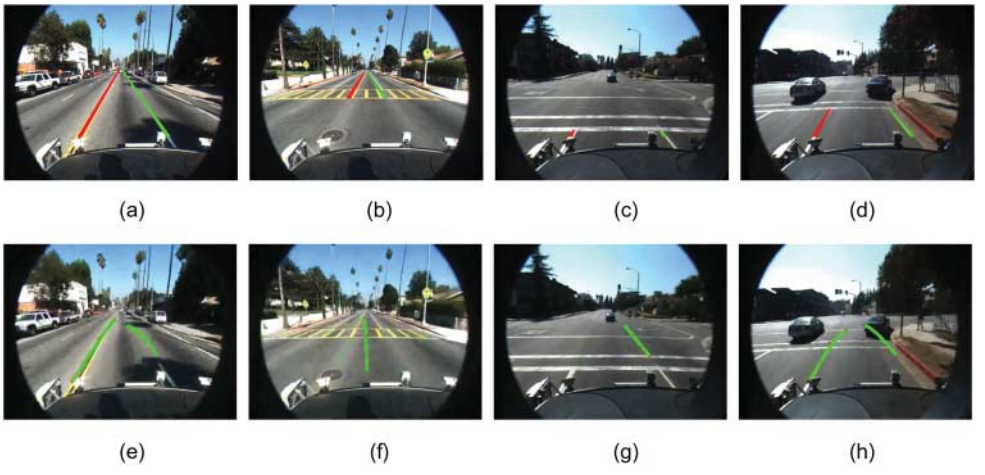


Figure 14. Experimental results comparison of our algorithm and Aly's algorithm [2]. Upper row: correct detection results using our algorithm. Lower row: false detection results using Aly's algorithm. Left to right: the images in each row are from washington1 clip, washington2 clip, cordova1 clip and cordova2 clip, respectively.

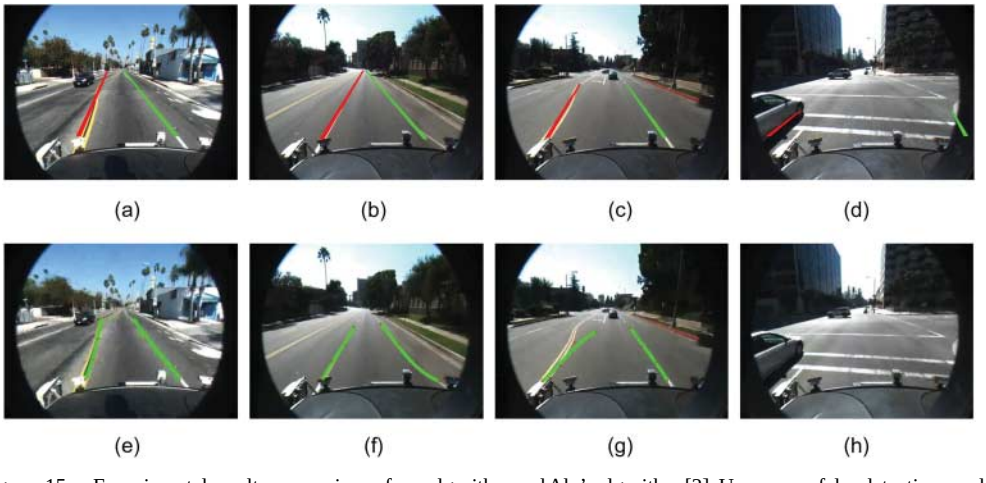


Figure 15. Experimental results comparison of our algorithm and Aly's algorithm [2]. Upper row: false detection results using our algorithm. Lower row: detection results using Aly's algorithm. Left to right: the images in each row are from washington1 clip, washington2 clip, cordova1 clip and cordova2 clip, respectively.

While our algorithm accurately detects all boundaries in the same road scenes, such as the clutter shadows in Figure 14(a), zebra stripes in Figure 14(b), short lanes in Figure 14(c) and passing vehicles in Figure 14(d).

As shown in Figures 15(a)–(c), the typical disadvantage of our algorithm is that the results for the curving lane detection are incomplete or false, since our algorithm adopts line segments as low level and does not use any lane model. But as shown in Figures 15(e) and 15(f), the curving lanes are accurately detected by Aly's algorithm. Meanwhile, Aly's algorithm may falsely fit the abnormal curving lane as shown in Figure 15(g). Another typical false detection of the lane is shown in Figure 15(d), in which the scenarios include very short lanes but lots of interferences similar to lanes, for example, the long shadow of the vehicle.

## 4. Conclusion

In this paper, we introduce an effective and efficient algorithm for lane detection in structured road scenes. Different from pixel level features or much prior knowledge required in other methods, we exploit line segments extracted by LSD as low-level feature to analyse the structural information of lanes, and then less prior knowledge (only orientation and vanishing point constraints) are necessary in our algorithm. We test our algorithm on challenging data sets and the experimental results prove effectiveness and robustness of the proposed method in complex traffic scenarios

## Acknowledgements

The authors thank the editors and the anonymous reviewers, whose comments helped to improve the paper greatly. This work was supported by the National Natural Science Foundation of China (No. 61172161), the Independent Research Funds of the State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body (Hunan University, No. 71165002) and the Commonwealth of Australia under the Australia-China Science and Research Fund (ACSRF02541).

## References

[1] C. Adam, R. Schubert, N. Mattern, and G. Wanielik, *Probabilistic road estimation and lane association using radar detections*, Proceedings of the 14th International Conference on Information Fusion, Chicago, IL, 2011, pp. 1–8.
[2] M. Aly, *Real time detection of lane markers in urban streets*, IEEE Intelligent Vehicles Symposium, Eindhoven, Holland, 2008, pp. 7–12.
[3] A. Borkar, M. Hayes, and M.T. Smith, *A novel lane detection system with efficient ground truth generation*, IEEE. Trans. Int. Transp. Syst. 13 (2012), pp. 365–374.
[4] Y. Chen and J. Krumm, *Probabilistic modeling of traffic lanes from GPS traces*, Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, 2010, pp. 81–88.
[5] H.Y. Cheng, B.S. Jeng, P.T. Tseng, and K.C. Fan, *Lane detection with moving vehicles in the traffic scenes*, IEEE Trans. Int. Transp. Syst. 7 (2006), pp. 571–582.
[6] K.Y. Chiu and S.F. Lin, *Lane detection using color-based segmentation*, IEEE Intelligent Vehicles Symposium, Las Vegas, NV, 2005, pp. 706–711.
[7] B. Fardi, U. Scheunert, H. Cramer, and G. Wanielik, *A new approach for lane departure identification*, IEEE Proceedings Intelligent Vehicles Symposium, Columbus, OH, 2003, pp. 100–105.
[8] M.A. Fischler and R.C. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Commun. ACM 24 (1981), pp. 381–395.
[9] X.B. Gao, Z.X. Niu, D.C. Tao, and X.L. Li, *Non-goal scene analysis for soccer video*, Neurocomputing 74 (2011), pp. 540–548.
[10] R. Gopalan, T. Hong, M. Shneier, and R. Chellappa, *A learning approach towards detection and tracking of lane markings*, IEEE Trans. Int. Transp. Syst. 13 (2012), pp. 1–12.
[11] Y. He, H. Wang, and B. Zhang, *Color-based road detection in urban traffic scenes*, IEEE Trans. Int. Transp. Syst. 5 (2004), pp. 309–318.
[12] M.S. Javadi, M.A. Hannan, S.A. Samad, and A. Hussain, *A robust vision-based lane boundaries detection approach for intelligent vehicles*, Inf. Technol. J. 11 (2012), pp. 1184–1192.
[13] R. Jiang, R. Klette, T. Vaudrey, and S. Wang, *Lane detection and tracking using a new lane model and distance transform*, Machine Vis. Appl. 22 (2011), pp. 721–737.
[14] Z. Kim, *Robust lane detection and tracking in challenging scenarios*, IEEE Trans. Int. Transp. Syst. 9 (2008), pp. 16–26.
[15] C. Kreucher and S. Lakshmanan, *Lana: A lane extraction algorithm that uses frequency domain features*, IEEE Trans. Robot. Autom. 15 (1999), pp. 343–350.
[16] A. Lopez, J. Serrat, C. Canero, F. Lumbreras, and T. Graf, *Robust lane markings detection and road geometry computation*, Int. J. Automot. Technol. 11 (2010), pp. 395–407.
[17] J.C. Mccall and M.M. Trivedi, *Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation*, IEEE Trans. Int. Transp. Syst. 7 (2006), pp. 20–37.
[18] Z.X. Niu, Q. Tian, and X.B. Gao, *Real-world trajectory extraction for attack pattern analysis in soccer video*, Proceedings of the International Conference on Multimedia, Florence, Italy, 2010, pp. 635–638.
[19] C. Rasmussen, *Grouping dominant orientations for Ill-structured road following*, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, Vol. 1, 2004, pp. 470–477.
[20] R.K. Satzoda, S. Sathyanarayana, and T. Srikanthan, *Hierarchical additive hough transform for lane detection*, IEEE Embedded Syst. Lett. 2 (2010), pp. 23–26.
[21] A.A. Sewisy, *Graphical techniques for detecting lines with the hough transform*, Int. J. Comput. Math. 79 (2002), pp. 49–64.

[22] R.G. Von Gioi, J. Jakubowicz, J.M. Morel, and G. Randall, *LSD: A fast line segment detector with a false detection control*, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010), pp. 722–732.
[23] Y. Wang, D. Shen, and E.K. Teoh, *Lane detection using spline model*, Pattern Recognit. Lett. 21 (2000), pp. 677–689.
[24] Y. Wang, E.K. Teoh, and D. Shen, *Lane detection and tracking using b-snake*, Image Vision Comput. 22 (2004), pp. 269–280.
[25] T. Weiss, B. Schiele, and K. Dietmayer, *Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids*, IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 2007, pp. 184–189.
[26] L. Xu, E. Oja, and P. Kultanen, *A new curve detection method: Randomized hough transform (rht)*, Pattern Recognit. Lett. 11 (1990), pp. 331–338.