A Lane Detection, Tracking and Recognition System for Smart Vehicles

by

Guangqian Lu

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

As important components of intelligent transportation system, lane detection and tracking (LDT) and lane departure warning (LDW) systems have attracted great interest from the computer vision community over the past few years. Conversely, lane markings recognition (LMR) systems received surprisingly little attention.

This thesis proposed a real-time lane assisting framework for intelligent vehicles, which consists of a comprehensive module and simplified module. To the best of our knowledge, this is the first parallel architecture that considers not only lane detection and tracking, but also lane marking recognition and departure warning. A lightweight version of the Hough transform, PPHT is used for both modules to detect lines. After detection stage, for the comprehensive module, a novel refinement scheme consisting of angle threshold and segment linking (ATSL) and trapezoidal refinement method (TRM) takes shape and texture information into account, which significantly improves the LDT performance. Also based on TRM, colour and edge informations are used to recognize lane marking colors (white and yellow) and shapes (solid and dashed). In the simplified module, refined MSER blobs dramatically simplifies the preprocessing and refinement stage, and enables the simplified module performs well on lane detection and tracking.

Several experiments are conducted in highway and urban roads in Ottawa. The detection rate of the LDT system in comprehensive module average 95.9% and exceed 89.3% in poor conditions, while the recognition rate depends on the quality of lane paint and achieves an average accuracy of 93.1%. The simplified module has an average detection rate of 92.7% and exceeds 84.9% in poor conditions. Except the conventional experimental methods, a novel point cluster evaluation and pdf analysis method have been proposed to evaluate the performance of LDT systems, in terms of the stability, accuracy and similarity to Gaussian distribution.

Acknowledgements

First and foremost, my sincerest gratitude goes to my supervisor, Dr. Azzedine Boukerche. With his patience and excellent expertise, he have dedicated his full effort to guiding me in achieving my goals as well as offering me financial support.

A special gratitude I would like to tribute to the leader of our Mobile Vision Group, Dr. Abdelhamid Mammeri, who is a nice friend and a very supportive tutor for me. He spares no effort in bringing me suggestions and encouragement on my work, especially with attentive and delicate guidance on my thesis writing.

Moreover, I also have to appreciate all the PARADISE team members, who keep supporting and assisting me during the past two years. I would like to give special thanks to my smart friend and colleague, Mr. Zongzhi Tang, who contributed in the implementation of the simplified module of my proposed system. Also I would like to thank my friend Mr. Mingchang Zhao, who shared a lot of knowledge on C++ and OpenCV with me. Thanks to Mr. Farzan Nowruzi for helping me propose the point cluster method. Thanks to Dr. Robson De Grande, Dr. Amir Darehshoorzadeh and all those who contributed to the possibility to complete my master thesis.

Last but not least, I would like to give my deep appreciate to my parents in China. They keep supporting me throughout my past two years in Canada.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Lane departure crashes count for the majority of highway fatalities and cause hundreds of human deaths, thousands of injuries, and billions of dollars in loss every year. It is reported in [93] that there were $15,307$ lane departure crashes resulting in $16,948$ fatalities in 2011, which counts for 51% of the total fatal crashes in the United States. In response to such stern problems, also to the trend of multimedia mobile application for ad hoc and telecommunication networks ([20], [16], [24], [18], [22], [29], [25], [17], [9], [39], [90], [27]), intelligent transportation systems (ITS) such as lane detection and tracking (LDT), lane departure warning (LDW) and lane marking recognition (LMR) systems are called for by industry. Tightly related to each other, these systems are briefly defined as follows. A lane detection and tracking system is a mechanism designed for localizing and tracking lane boundaries for road lanes. Lane departure warning system aims at providing a warning scheme for drivers when the vehicle crosses prohibited edge lines in an inappropriate moment, which usually functions based on lane detection results. As a supplement to LDT system, lane marking recognition systems are used to recognize and distinguish between lane marking categories.

Over the last decade, lane detection, tracking and lane departure warning systems have attracted extensive interests of the automobile industry and computer-vision community. Many architectures and commercial systems have been proposed in the literature, namely as Lane Keeping Assistance (LKA) ([34], [60], [68] and [115]); Lane Departure Warning

(LDW) ([62], [30], [74] and [76]); Lateral Control (LC); Intelligent Cruise Control (ICC); Collision Warning (CW) ([76]), and Autonomous Vehicle Guidance ([125]). However, lane marking recognition systems have received surprisingly little attention compared to LDT and LDW systems. From our exhaustive literature review, only one paper (patent) found was [63], where the authors extract coloured areas by computing the luminance intensity of each pixel, by means of which, white and yellow colors can be recognized respectively and fed back to detection stage. Another work relating to LMR system has been done in [97] to distinguish between solid and dashed lines, by counting the presence of lane marking in one scan band. There is no exited work combining the recognition of both lane marking types and colours.

1.1 System Overview

In this thesis, we proposed a real-time Lane Assist System for Intelligent Vehicles, aiming at lane detection, recognition and tracking integrated with Lane departure warning. Different segmentation algorithms are implemented, which makes the proposed system consists of two different modules, a comprehensive module which takes advantage of edge segmentation and a simplified module with MSER segmentation.

For the comprehensive module, it is worth to be mentioned that, after the detection and tracking of lanes, our system takes advantage of the parallel computation to perform lane departure warning and lane marking recognition. To the best of our knowledge, this is the first whole architecture that considers lane detection and tracking, lane departure warning and lane marking recognition systems. The comprehensive module not only helps to detect and track lanes, but also warns inattentive drivers on the nature of pavement marking, and on the potential lane departure crashes.

As a lightweight LDT system, the simplified module focuses only on lane detection and tracking, which occupies the main function of lane assist systems. Different from the comprehensive module, the advantages of using MSER segmentation for LDT system can be outlined as following:

- To provide a supplement preprocessing approach for night scene (it has been experimentally proved in Section 4.3 that MSER segmentation performs better than edge segmentation at night);
- A relatively simple way of extracting ROI by MSER algorithm, without projective model and Inverse Perspective Mapping (IPM);
- A relatively lightweight and efficient refinement scheme without ATSL and TRM (Section 3.2.2 and 3.2.3), which are essential parts of comprehensive module.

With the comprehensive and simplified module in hand, the proposed system stands out from our predecessor's work in the following aspects:

1.1.1 Thesis Contribution

- The refinement stages in both modules make use of hybrid information and is carried out based on both shape and texture information, which could result in an increase in the detection rate. For refinement of detection results, a novel Trapezoidal Refinement Method (TRM) is proposed in comprehensive module, while the simplified module finds ROIs based on a refinement scheme of MSER blobs.
- To fulfill the requirements of real-time systems, the Progressive Probabilistic Hough Transform (PPHT) has been used in both comprehensive and simplified modules, as well as Kalman filter. Outstandingly, the proposed system is the first LDT system which combines the results of PPHT with Kalman filter, by tracking both ends of each detected line-marking.
- A fairly new lane marking recognition scheme is proposed in the comprehensive module. It recognizes and distinguishes between the nature of line marking forms (dashed or solid) and the colors (yellow or white). This recognition scheme is based on the concentration of pixels in trapezoids computed by TRM.
- Different from previous departure warning mechanism, the LDW system in comprehensive module creatively uses both ends of each detected lane markings and splitting

the ROI (region of interest), to warn the driver when the vehicle is potentially crossing lane boundaries.

- This thesis also presents new methods of performance evaluation of LDT systems (in Section 4.4.2) based on the spatial distribution of point clusters and probability density of point locations of both left and right markings in each frame.

1.2 Thesis Outline

The rest of this thesis is organized as follows. We start the rest part by reviewing the main research works in Chapter 2. Chapter 3 elaborates all the modules and stages in our system, which are, preprocessing stage in Section 3.1 which consists of different segmentation methods for both comprehensive (Section 3.1.1) and simplified (Section 3.1.2) module, followed by detection stage in Section 3.2 and tracking stage in Section 3.3; the novel LMR and LDW system for comprehensive module will be in Section 3.4 and 3.5, respectively.

Several experiments are introduced in Chapter 4. As a start of the experimentation stage, the platform, video collection and experimental methods are elaborated in Section 4.1 and 4.2. The LDT performance evaluation will be covered respectively in Section 4.4 and 4.3, followed by the evaluation on edge detection performance (Section 4.5) and LMR& LDW performance (Section 4.6). The two new evaluation methods will be introduced in Section 4.4.2 and 4.4.2. We conclude our thesis and present future work in Chapter 5.

Chapter 2

Related Work

The existed research on Lane Marking Detection is presented in different forms. Systems that have been developed can perform different tasks, namely Lane Following (LF) ([55] and [56]), Lane Keeping Assistance (LKA) ([68], [116] and [115]), Lane Departure Warning (LDW) ([62], [30], [40], [67], [74] and [76]), Lateral Control (LC), Intelligent Cruise Control (ICC), Collision Warning (CW) ([76]), and Autonomous Vehicle Guidance (AVG) ([125]), which can be integrated with VANET applications ([3]). In general, research works can be divided into two types, which is demonstrated as follow:

The first systems are those without units of controllers, as Lane Departure Warning (LDW), Collision Warning (CW). This type of system only detects information and sends warning or reminding messages if necessary. These systems are usually implemented in Driving Assistant System (DAS), instead of impacting on vehicles in a Autonomous Vehicle Guidance. The second type of system is designed with feedback, and with the aim at impacting on vehicle behaviour, which functions the way as a controller. Systems as Lane Following (LF) and Lane Keeping Assistance (LKA) follow this type of systems. LKA systems apply some methods to control the vehicle to avoid potential departure or collision. It has to be noticed that, the second type has much more complexity of hardware than the first type, also more time-consuming than the first type.

The system proposed in this thesis, as a second type of Lane Marking Detection system, deals with lane marking detection and tracking, lane recognition and departure warning.

The proposed system in this thesis does not directly impact on vehicle behaviour. It only detects the location of lane markings and sends necessary warning messages. Based on this fact, related work can be presented in four sections: pre-Processing, detection, tracking and departure warning (this structure will also be used to discuss the proposed method in Chapter 3).

Classification	***References***	***Remarks***
LF	[55] and [56]	Lane Following
LKA	[68], [116] and [115]	Lane Keeping Assistance
LDW	[62], [30], [40], [67], [74] and [76]	Lane Departure Warning
CW	[76]	Collision Warning
AVG	[125]	Autonomous Vehicle Guidance

Table 2.1: Classification for lane Marking Detection System

2.1 Pre-Processing

Pre-processing is always mandatory as the initial stage of image processing. The purpose of pre-processing stage is to enhance the input image in order to increase the likelihood of the successful delivery of areas with useful information to subsequent stages. Conventionally, pre-processing consists of image smoothing and segmentation, which will be introduced as follows. Additionally, for the purpose of lane Detection, extraction of Region of Interest (ROI) and Inverse Perspective Mapping (IPM) are usually added into pre-processing steps.

2.1.1 Image Smoothing

Aiming at lane marking detection, image smoothing can be done by applying two main filters, Median Filter ([44], [6], [54], [91], [105], [43] and [129]) and Gaussian filter ([38], [123], [73] and [127]) or both ([113] and [114]), to blur the noisy details. Different from those who mostly use 2D-Gaussian filters, Shih et al. only uses a 1D-Gaussian filter in [52],

Classification		*References*	**Remarks**
Image Smoothing	**Median Filter**	[44], [6], [54], [91], [105], [43], [129], [113] and [114]	Median filter replaces every pixel value with the median of its neighboring entries
	Gaussian Filter	[38], [123][73], [127], [113], [114] and [52]	[52] uses 1D Gaussian filter exceptionally of all
	Other Filters	[48], [46], [53] and [13]	Including dilation and erosion filter, 2D high-pass filter and a temporal blurring
ROI	**Vanishing Point**	[130], [85], [121], [41], [47], [100], [101], [110], [77], and [123]	Vanishing points need to be detected and updated in real time, in order to provide accurate ROI for every frame
	Perspective Analysis and Projective Model	[85], [130] and [88]	Sometimes this method serves to detect vanishing point, as in [85]
	Sub-sampling	[102], [105], [127], [10], [59] and [84]	Different sub-sampling strategies need to be applied respectively for rural and urban areas
IPM		[38], [4], [13], [106], [71], [85], [12], [72], [96], [97] and [92]	IPM refers to inverse perspective mapping, which transforms image view into birds' eye view
Segmentation	**Edge-based Segmentation**	[49], [130], [104], [71], [82], [68], [36], [105], [77], [41], [107], [99], [13], [124] and [120]	Canny, Sobel and Prweitt are included in the literature. Relevant experiment on the selection of edge detectors is included in Section 4.5
	Color-based Segmentation	[32], [36], [35], [8], [46], [97], [85], [105], [70], [75], [94], [110], and [123]	Different from color-based detection (Section 2.2.2), colour-based segmentation has to be done in different colour spaces other than RGB.
	MSER-based Segmentation	[109]	A improvement of MSER Segmentation is proposed in Section. 3.1.2

Table 2.2: Classification for Methods of Preprocessing Stage

which respectively takes x-direction and y-direction into account and significantly smooths the details of input images.

Apart from Median and Gaussian filters, other filters are also used by researchers. Dilation and erosion filters are used in [48] and [46]; 2D high-pass filter is used in [53] to weaken the effect of shadow on road. A temporal blurring is deployed in [13] to make lane markings appear long and continuous, also to blur noisy details on road surface.

2.1.2 Region of Interest (ROI)

The extraction of region of interest (ROI) is an important task of pre-processing stage, aiming at reducing the computational cost due to the processing time. It is unnecessary to process the entire pixels of images. Computation should be focused on regions which contain important information (as the rectangular in Figure 2.1, which contains lane marking pixels). To get the ROI, three main approaches can be found in literature, which are vanishing point detection, perspective analysis and projective model, and sub-sampling.

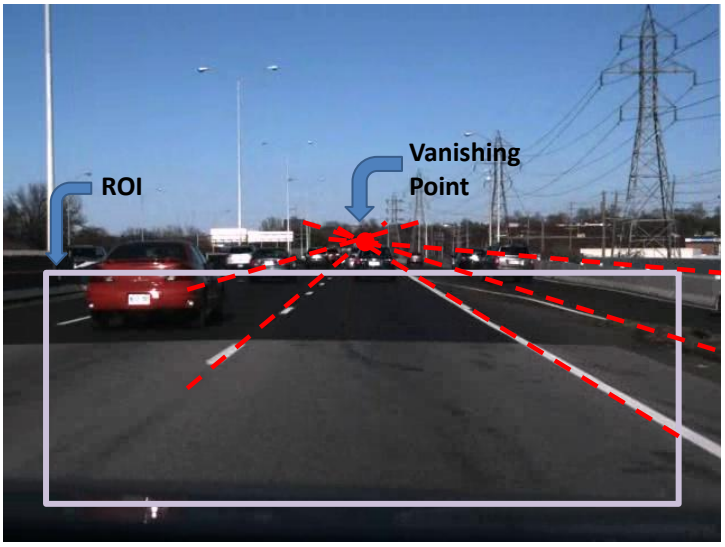


Figure 2.1: ROI and vanishing point: the red point where all lines intersect is vanishing point (as introduced in Section 2.1.2); the rectangular which specifies the detection range is reffered as region of interest (ROI)

Vanishing Point Extraction

Vanishing point detection is used to determine the region of interest in many papers, such as [130], [85], [121], [41], [47], [100] and [101]. However, as seeing from Figure 2.1, this has some strict situations, such as straight lanes with constant vanishing points; and the vanishing point does not fall outside the image frame.

In an attempt to combine vanishing point extraction with Hough transform, fardi et al. extract a vanishing point based on the intersection of a group of lines detected by Hough transform. However, this method can be only adapted to standard lane marking distributions with only one vanishing point. To improve the method of [41], David et al. extended the case from one vanishing point to multiple vanishing points, by applying a 2D-Hough transform in [100]. Even the 2D-Hough transform can only cooperate with straight lines, it still improves the fitness of lane marking detection for non-standard images. Apart from above, some methodologies took the part between the bottom of the image and vanishing point or vanishing line as region of interest ([110], [77] and [123]).

Perspective analysis and projective model

The lane marking width and shape change as a result of perspective effect. Parallel lane markings in real world plane intersects at vanishing point in the image plane. Usually, by analysing the perspective effect, detection range can be focused on a certain area, which can be region of interest. With a reasonable projection applied between image plane, real world plane and camera plane, not only the region of interest can be extracted, it also benefits overcoming the perspective distortion, getting the position of vanishing point and bird's eye view (which will be introduced in Section 2.1.3).

A projective model can be determined by computing the homography between real world plane and image plane. An example of projective model is shown in Figure 2.2. The X_c, Y_c, Z_c is camera plane, the X_w, Y_w, Z_w is world plane constructed on a road surface, while the u_i, v_i is image plane. The relation between the three planes is discussed in Section 3.1.1

Road plane needs to be transformed to image coordinate system through projection matrix given by camera calibration (which will be introduced in Section 3.1.1). It is noticeable that, as an essential component of a homography, the calibration matrix of camera is usually computed off line. As described in [85], a pinhole camera projection model is used to compute the position of vanishing point, by computing the unknown parameters of the rotation matrix.

RALPH (Rapidly Adapting Lateral Position Handler [88]) constructed a very basic projection model to get region of interest, which is a trapezoid, to focus on lane marking areas, so that irrelevant area (road surface) can be eliminated. Due to the perspective effect, the bottom is shorter than the top of the trapezoid, with their length vary based on vehicle velocity.

As discussed in [130], a projection model is constructed based on a 2D lane geometric model, as shown in Figure 2.2. This projection not only benefits to ROI extraction, it also helps to estimate lane model parameters and lane model matching for the refinement stage.

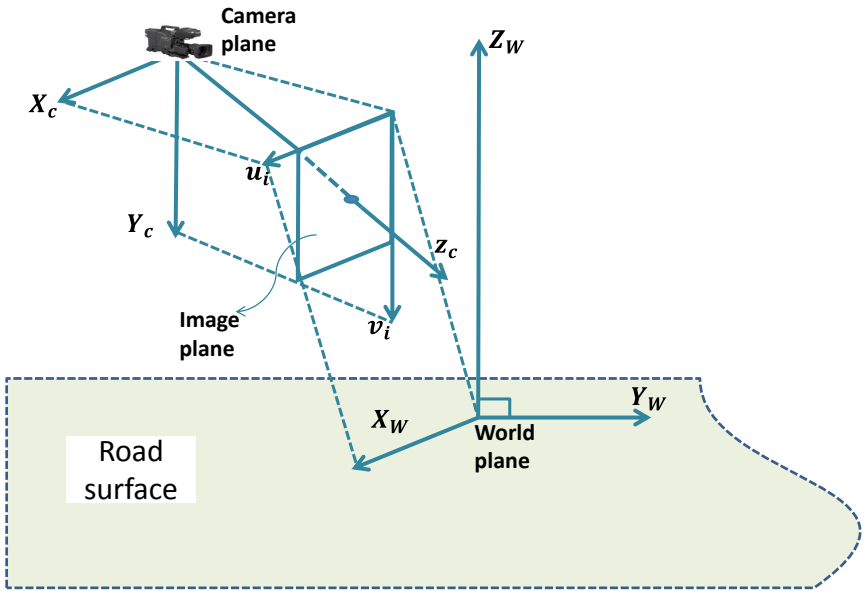


Figure 2.2: An example of projection model: X_c, Y_c, Z_c, X_w, Y_w, Z_w and u_i, v_i refer to camera plane, world plane and image plane respectively.

Sub-Sampling

It is also reasonable to split ROIs by sub-sampling ([102] and [105]). A predefined or adaptive percentage of the image can be used to determine the size of region of interest ([127]). Besides, some researchers divide the image horizontally ([10]) or vertically ([59]) into small parts. By conducting different strategies for rural and urban areas, Jeong and Nedevschi ([84]) applied predefined percentage to split ROIs for rural ways and adaptive percentage for highways.

2.1.3 Inverse Perspective Map (IPM)

After the generation of ROI, inverse perspective mapping (IPM) is usually deployed on the extracted area. This may be realized by remapping each pixel towards a different position, usually birds' eye view. Experimentally, it has already been proved that Lane detection within the image given by IPM performs much better than methods without IPM. Many researchers (e.g., [38], [4], [13], [106], [71], [85], [12], [72], [96], [97] and [92]) have used IPM to transform an image from a real world plane to a bird's eye view.

As seeing from Figure 2.3, by eliminating the perspective effect, the desired lane marking candidates present in the form of straight and parallel lines in the re-mapped image (Figure 2.3(b)). Moreover, noisy information can be removed since the remapped images put main focus on the road surface and ROI (as Figure 2.3(a)).

It is noticeable that, for the sake of refinement, proper projective model can be used to transform between normal view and birds' eye view (as indicated in Figure 2.3). Thus in the comprehensive module of the proposed system, inverse perspective map is transformed into normal image plane after hough transform, by using proper projective model.

2.1.4 Segmentation

To prepare images for detection stage, segmentation can be accomplished by extracting certain features from the input image. Colour and edge are two main features which are

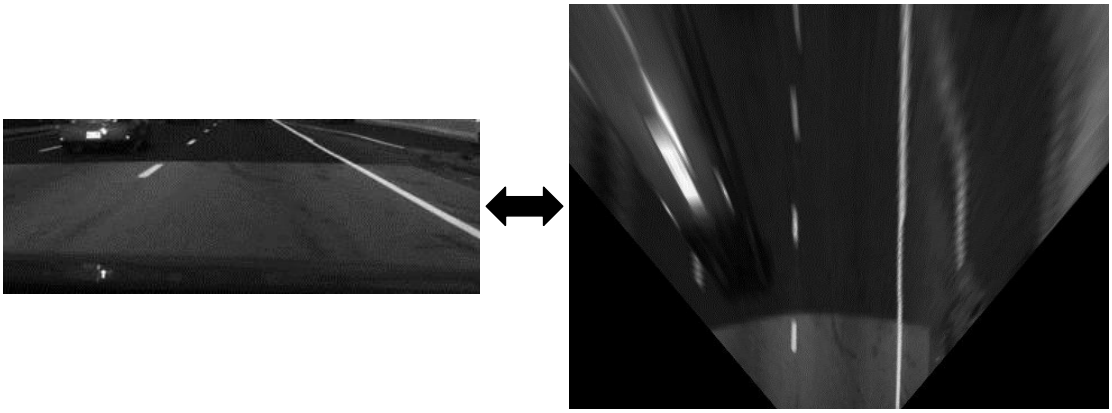


Figure 2.3: Inverse perspective mapping (IPM): (a) is the ROI of Figure 2.1, (b) is the birds' eye view of (a).

considered for lane detection segmentation. The major problem with edge segmentation is that a number of thresholds need to be adjusted, and the adjustments depend only on the image undergoing testing, while color segmentation techniques are sensitive to lighting conditions, especially those occurring on sunny days.

Edge-based Segmentation

Edge-based segmentation usually consists of edge detection and binarization. Different edge detection methods are proposed based on edge features, which usually include shape (distribution of edge points) and gradient information. Canny algorithm is used in [49], [130] and [104]. Hota et al. applied a Gaussian filter to smooth the input image based on gradient information in [49], followed by a Canny edge detector. Different from Canny, prweitt is used for edge detection in [71]. Steerable filter is another effective approach for edge segmentation. As demonstrated in [82] and [68], edges are extracted based on lane orientation characteristics with applying steerable filter.

A straightforward step after edge detection is to apply binarization, by doing this properly, effective details can be reserved while noisy points being neglected. Different methods have been proposed to find suitable threshold for binarization.

A predefined threshold based on massive experimental results is used for this task in

[36] and [105]. Because of the dependence on massive experimental results and previous experience, predefining the threshold can be inaccurate. Hence, adaptive threshold is proposed as a more effective approach for segmentation ([77], [41], [107], [99], [13] and [124]) to binarize images. Also as an adaptive way to threshold images, Otsu's method has been used in [120] to segment candidate areas from jumbled backgrounds.

Colour-based Segmentation

Most of the case, different color features can be presented based on different colour spaces (RGB, HSV, gray scale, etc.) In the literature, data structure of the images can be changed in order for better performance based on different colour spaces. For structured roads (mostly highways), it is better to use gray images for colour representation of images to be detected, because gray images make lane markings with light colours (white or yellow) distinguished from road surface (with dark gray for most of the time). Hence, gray-level conversion is considered in many papers, such as [32], [36], [35], [8], [46], [97], [85] and [105]. As in [85], a parametric multiple-class likelihood model of the road is proposed based on the gray level of pavement, lane markings and objects. Parameters are estimated by the Expectation-Maximization (EM) algorithm.

Different from directly converting coloured images into gray level, [70] only uses R and G channels of coloured images to form gray images. Furtherly, to make full use of the colour information of RGB images, some researchers deployed colour format conversion in their algorithm. For example, in [75], RGB images are converted to HSV. However, format conversion of coloured images may inevitably lose a certain amount of detail information in real scenarios. To solve this problem, authors of [94] and [110] converted RGB to HSI. Sun et al. revealed the advantage of combining HSI model with loosen thresholds in [110], in terms of keeping useful information and highlighting the difference between lane markings and road surface. The method proposed in [123] is based on the color consistency of the road surface. The road color is assumed to be Gaussianly distributed, by which the road surface colour and relevant variance can be computed.

MSER-based Segmentation

Hao Sun et al. proposed a method of describing MSER blobs using SIFT-based descriptors in [109], followed by a graphical model in order to localize lane markings. This method combines unsupervised learning algorithm with off-line training. It consists of 3 main steps: 1) obtain descriptors by extracting the feature of lane markings; 2) quantize descriptors into which relies much on visual words retrieving and 3) the image is described as a combination of visual words.

There are two drawbacks for this method: 1) the processing time varies in different scenarios, as the number of features detected varies; 2) the visual words for the feature of lane markings mostly consist of line segments, which takes both salient lane markings and noisy lines into account. This might increase the false positive results. Based on the idea of using MSER to extract the features of lane markings, we use a refinement method to eliminate noisy line candidates and feed the refined MSER results into next stage, which is detection with Hough Transform (HT).

2.2 Detection Stage

The second stage, i.e., detection, extracts lane markings from the ROI using feature extraction methods and refinement approaches. Three main feature extraction approaches can be categorized in the literature: edge-based methods, colour-based methods and hybrid (edge and color) methods.

2.2.1 Feature Extraction Based on Edges

Hough transform is the most commonly used edge extractor for this application, as shown in [4], [34], [38], [13], [101] and [47]. However, Hough transform has some drawbacks such as its computational complexity and the inevitably high false positive rate. To cope with this problem, variants of Hough Transform are used. Probabilistic Hough transform (PHT) and adaptive random Hough transform (ARHT) are both effective approaches in terms of

Classification		*References*	**Remarks**
Edge-based Detection Method	**Hough Transform**	[34], [38], [13], [101], [47], [123], [49], [70], [69] and [85]	Including SHT, PHT and ARHT
	Steerable Filter	[82], [96], [97], [119] and [106]	Effective for smooth road surface, e.g. highway
	Frequency Domain	[64]	IFT is used to choose diagonally dominant edges
Colour-based Detection Method		[108], [111] and [12]	The desired feature is usually enhanced based on colour segmentation (Section 2.1.4)
Hybrid Detection Method		[7], [32] and [100]	Taking both colour and edge information into account
Other Detection Method		[42], [45] and [35]	Including using wave raddar, "slice" modelling and particle filter for both tracking and detection
Refinement Method	**RANSAC**	[13], [4], [106], [118], [38], [85] and [60]	RANSAC refers to random sample consensus, which usually cooperate with spline, as in [60] and [4]
	Spline Fitting	[72], [38], [123], [4], [122] and [60]	Different from others using B-Spline, [60] uses cubic spline
	Quadratic Model	[118], [122], [121], [97], [130], [106], [70], [104]. [69] and [72]	Including hyperbolic and parabolic model
	Least Square	[72], [32], [85] and [13]	The least square estimation (LSE) fits the data points to a line on the inlier

Table 2.3: Classification for Methods of Detection Stage

reducing false positive rate. PHT has been used in [123] and [49] while ARHT performs well in [70] and [69]. Especially, Hota et al. shows some improved detection results in [49] by using the Probabilistic Hough Transform (PHT) followed by refinement based on clustering regression algorithm. Because of the importance of PHT, which is the core detection method of this thesis, a short survey on PHT will be presented in Section 2.2.1

Apart from Hough transform and its variants, another edge-based method based on steerable filter is used in many research papers such as in [82], [96], [97], [119] and [106]. A steerable filter is convolved with an input image to select the maximum response in the direction of lane marking. This method has good effects when road markings are clearly painted and consistently smooth. However, Steerable filter does not adapt to heavy traffic where the orientation of lane markings are not always dominant of all directions.

Another important edge-based method is introduced in [64], where the authors compute the image with the Inverse Fourier Transform, and choose diagonally dominant edges in the frequency domain to be lane markings.

A Short Survey on PHT

In the original paper on probabilistic Hough transform [61], Kiryati et al. shows that it is possible to obtain identical results to those of the standard Hough transform even if only a fraction (can be as low as 2%) of input points are chosen to vote.

Unfortunately the derived equations in [61] require a priori knowledge of the number of line length (numbers of points belonging to the line), which is very rare in real scenario. To solve this problem, in [11], Bergen shows that a Monte Carlo estimation of the Hough transofrm can interpret the probabilistic Hough transform. The number of voting points can be derived from the theory of Monte Carlo evaluation.

Besides above, to pre-select a poll size (fraction of voting points), an adaptive scheme which is called adaptive probabilistic Hough transform (APHT) has been used in [103], [2] and [126]. In [2] and [126], Kiryati and A. yla et al. developed an adaptive termination rule for probabilistic Hough transform. The difference between [2] and [126] is to choose the

highest peak or a certain number of largest peaks, respectively for [2] and [126]. But the draw back of this approach is that, it depends on the prominence of the most significant feature, which means it lacks of stability when it comes to lines with equal length.

2.2.2 Feature Extraction Based on Colours

These methods are quite efficient for unstructured roads, or rural roads without clear lane boundaries. However, unlike edge-based methods, color-based methods are not widely used by researchers. Color information has its own drawbacks as it is influenced by lightning. Hence Detection based on colour does not work well for night scenes or heavy and complex traffic.

For example, Sotelo et al. used a color-based method in the HIS color space by computing the cylindrical distribution of color features ([108]). Colour segmentation based on gray level is used in [111] and [32]. The authors of [12] enhanced the image by exploiting its vertical correlation, followed by an adaptive binarization.

2.2.3 Feature Extraction Based on Hybrid Information

Hybrid information is usually composed of edge information and colour information. This type of feature extraction scheme usually combines width, length, location (coordinates of pixels) of lines with gray level and brightness intensity of groups of points, which makes the extraction results better. Gray level is used in [32], to generate the threshold for next binarization stage. For the sake of deploying hybrid information, based on the binarization results containing lane marking pixels and noise, the width of possible lane markings is examined. The criteria is given that a qualified lane marking should be continuous in vertical and less than 30 pixels in horizontal width.

Zelinsky et al. uses hybrid information in [7] for both feature extraction and lane tracking stage. Instead of only using road edge information as a cue for particle filter, road and non-road colour information are also used to make the particle filter performs well. Another example of using hybrid information is in [100], lane boundaries are deemed as a

combination of four lines, which are left and right boundaries of left and right lane marking, respectively. lane marking is treated as a region with a certain thickness (space between left and right boundaries). The grading criterion for the hypothesis is first computed separately for the left and right line pairs, and then the two are combined together. The grading scheme are based on both edge and gray-level information, which are mean support-size of a pair; mean perpendicular gradient of a pair; similarity of a pair; mean continuity of a pair and mean lateral accuracy.

2.2.4 Other Detection Methods

An interferometric linear frequency modulated continuous wave radar is deployed in the road lane detection scheme in [42]. In the configuration, interferometry is set to estimate lane position. It is based on the fact that, The radial velocity difference between antennas is determined by the lane position. Phase difference are also considered for the received signals.

Another novel detection method employs road modelling by dividing roads as "slices". Foedisch et al. proposed a model-guided road recognition process in [45]. As the road model primitives, "slices" of a road can be determined by lane width (geometrical component) and the number of lanes and directions (topological component). This representation of road model primitives is consistent with the feature data extracted based on "road slices" perpendicular to the direction of the driving vehicle.

Instead of following the conventional lane detection approach that tracking stage comes after detection stage, Danescu et al. initialized the system by using particle filter to track lane marking candidates, followed by a validation stage in [35]. The reason why a particle filter can skip over detection stage is because a validation scheme is added to choose the valid lane candidates and lock the particles on lane.

2.2.5 Refinement Methods

For further refinement of the detection result, many techniques such as B-Spline fitting and RANSAC are used, as shown in [13], [122], [4], [106], [118], [60], [123] and [38]. Quadratic model is also used in detection-refinement in many research papers such as in [118], [122], [121], [97], [130], [106], [70], and [104].

Quadratic models usually cooperate with parameter estimation, to improve not only the efficiency but the robustness of the whole system. Different parameters can be initialized with estimation of road models and then updated by detection stage. As in [104], a novel model is constructed based on the morphological multi-structure elements, then after applying Hough transform to accurately locate lane markings, the model can be reconstructed for the sake of refinement. However, road modelling is a challenging task when a road has very complex shapes or rapidly changing curvatures.

As demonstrated in [123], at the end of a region growing scheme integrated with edge enhancement, Bezier spline algorithm serves as a refinement of detection stage. Additionally, different from traditional Bezier spline fitting, control points are optimized by minimizing the pixel count between the original and fitted boundaries. The Least Square Estimation (LSE) is used by Kuoyu for the estimation of quadratic model generated by Hough transform in [32]. Differently, a linear Least Square Estimation is deployed in [13], fitting the data points to a line on the inlier. Another example of using LSE can be found in [72]. A weighted least square constraint reduces the computational cost of the original EM-based vanishing point estimation algorithm.

Methods used in Refinement stage usually consists of more than one algorithm. Xiangyang et al. proposed a refinement method by combining parabolic road model and Least Square Estimation in [72], extending the B-Spline algorithm into a Parallel Snake algorithm for left and parallel lane markings. As an improvement of the Spline Fitting in [4], Jiayong et al. proposed a Optimized RANSAC Spline Fitting (as in [38]).

2.3 Tracking Stage

To enable the following of lane marking over time, a tracking stage is usually incorporated. This stage has the ability to decrease false detections and to predict future lane markings positioning in the image. A prior knowledge of the road geometric properties allows lane tracking to put confidently constraints on the possible location and orientation of lane markings for a new frame. Results of the detection on previous frames should be fed into the current frame as the main cue, to increase the accuracy of following steps. Some simple tracking methods (not involving a specific tracking algorithm, as in [41]) can be quite straightforward. By directly taking the position of vanishing point into account, Fardi et al. track the road border after detecting straight lines with Hough transform in [41].

Classification	***References***	***Remarks***
Kalman Filter	[117], [85], [38], [82], [96] [97], [33] and [112]	[112] use Extended Kalman Filter
Particle Filter	[60], [7] and [35]	Methods differ from each other in terms of different measurement cues
Other methods	[41]	[41] only considers the vanishing point position

Table 2.4: Classification for Lane Tracking Methods

2.3.1 Kalman Filter

The most common trackers used in LDT systems are Kalman Filters and Particle Filters. Previous work has been done on tracking the parameters of Hough Transform by using Kalman Filter as in [117], [38], [82], [96], [97], [85] and [33]. Usually, as the output of standard Hough transform (SHT as mentioned in Section 3.2.1), (ρ, θ) is used to initialize the Kalman filter. With proper setting of state transition matrix, the state vector can be

updated, which consists of (ρ, θ) and its derivative. Kalman filter is employed in this thesis as the tracking algorithm ([95]), which is elaborated in Section 3.3.

Despite the advantages of Kalman filter in terms of real-time performance, it is unable to reject outliers that cause failure of tracking, which in turn decreases the accuracy of detection. To alleviate these problems, an extended Kalman filter (EKF) for road tracking was introduced in [112]. EKF is designed for the tracking stage of a non-linear dynamic systems, such as lane detection in complex road environment where the noise might pose as a non-Gaussian distribution.

2.3.2 Particle Filter

Particle filter is another reliable option for lane tracking. It has been used in [60], [7] and [35]. Kim et al. proposed a method in [60], which combines lane border hypotheses obtained by RANSAC with hypotheses from a particle filter. Different from the method in [60], Danescu et al. takes more information into account, which are horizontal and vertical curvature, lateral offset and lane width. In terms of visual cues for robust lane tracking, [35] uses curbs and road edges as the cues, while [7] uses more than those, which consists of lane marking, road edge, road and non-road colour and road width. Usually, more reliable cues makes the lane tracking with particle filter performs well in more complex situations.

With a comparison between kalman filter and particle filter in [35], Danescu et al. pointed out that, particle filter and Kalman filter have their own advantages and disadvantages. Unlike a Kalman filter, a particle filter does not need initialization or measurement validation before updating the state matrix. Particles can evolve by themselves, and might have chance to cluster around the best lane estimation. However, the premise is the tracking system must be well designed and the measurement cues are relevant, which requires the system must know how to validate a lane candidate. Different from tracking with particle filter or traditional Kalman tracker, the tracking scheme in this thesis is to track the output of progressive probabilistic Hough transform (PPHT) by using Kalman filter (as introduced in Section 3.3.2).

2.4 Departure Warning

Classification	*References*	*Remarks*
Vanishing Point	[47], [5], [65] and [51]	Different from Section 2.1.2, the vanishing point position needs to be compared with previous frames
Lateral Offset	[97]	Lateral offset of lane markings need to be tracked in real time
Position of Origin Point	[40] and [34]	Optical center of CCD and vehicle origin in world plane are considered for [40] and [34] respectively.

Table 2.5: Classification for Departure Warning Methods

When a car is departing from the left side of a lane to the right side, visually for the driver, both of the lane markings move to left; on the other hand, both lane markings can appear to move to right if the car is moving towards the left side of a lane (as shown in Figure 2.4). Based on this observation, the detection scheme for lane changing can be formulated with respect to three informations: vanishing point position, lateral offset, optical center of CCD.

- **Vanishing point position**

  When it comes to Lane Departure Warning (LDW), the most common solution is to determine if a lane departure occurs by examining the horizontal location of the vanishing point, as performed in [47], [5], [65] and [51]. Different from the vanishing point detection for extracting ROI (Section 2.1.2), vanishing point position needs to be compared with those of previous frames, in order to find the car's horizontal displacement. The effects of these methods are desirable, but more time is required from them to detect vanishing points.

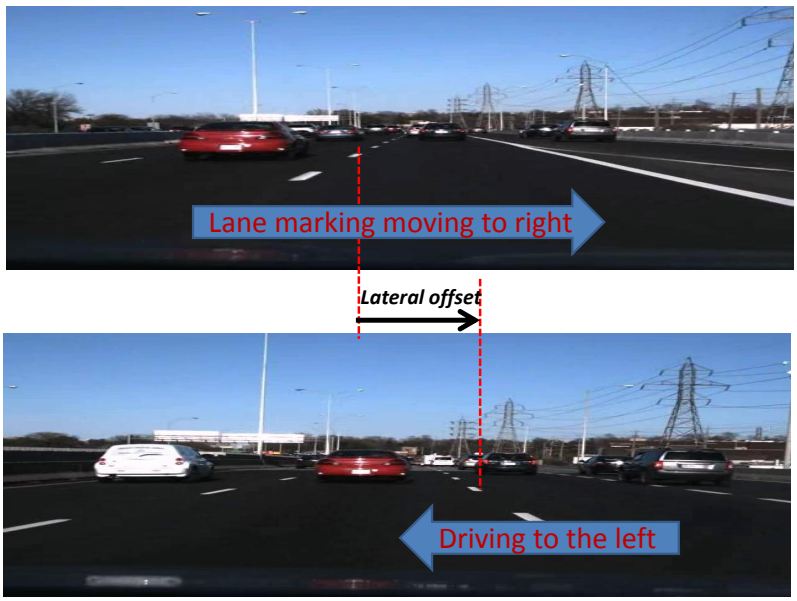


Figure 2.4: A lane departure case on Highway in Ottawa: driving to left leads to lane marking moving to right, with a lateral offset.

- **Lateral offset**

  As we can see from Figure 2.4, lateral offset reflects the distance difference of same lane marking in different frames. After employing Kalman filter for the tracking stage, the lateral offset of the left and right lane marking is being tracked in [97]. As well as the yaw rate change of the vehicle, the lateral offset of the vehicle due to yaw rate change is also used to cooperate with the determination of the occurring of lane departure.

- **Position of Origin Point**

  Fardi et al. proposed a Lane Departure Identification method in [40] based on the location of optical center of CCD and center of a lane, which require the optical center is always consistent with lane direction. Whereas in [34], lane departure is determined by monitoring the position of vehicle origin with respect to the detected lane boundaries. The vehicle origin is set to be a point below the camera in world plane. This method is more convenient and efficient as compared to vanishing point detection, because of less time consumption for getting the position of origin point.

2.5 Lane Marking Recognition

Recognition of lane markings is another very challenging task. It is notable that, the differentiation between lane marking colors and types are barely suggested in the literature, nor in the commercial systems. To recognize yellow and white lines on the road, there existed some patents on lane colour recognition, such as [63]. Besides, when it comes to lane type recognition (distinguish between solid and dashed lane markings), Satzoda et al. determines the lane type by monitoring the lane marking occurrence within a given number of frames ([97]). If the ratio of the number of frames containing lane markings over the entire number of frames is equal or greater than a threshold, the lane markings can be regarded as solid lines, otherwise dashed lines.

Although marking recognition is not as important as other components of DAS (Drving Assistant System), it is of great significance for Autonomous Driving Systems. In this thesis, an algorithm that recognizes and distinguishes between both lane marking types (dashed or solid) and colors (yellow or white) is proposed in order to provide possible recognition options for autonomous driving, which will be introduced in Section. 3.4

Chapter 3

System Architecture

As explained in Chapter 1, a real-time system is proposed in this thesis aiming at lane detection, recognition and tracking integrated with Lane departure warning. Apart from a comprehensive module, a simplified module with different segmentation scheme has been proposed. Because of the different segmentation methods used for comprehensive and simplified module, the setting of parameters in terms of detection and tracking algorithms (PPHT and Kalman filter respectively in this case) are different. This yields the implementation of both modules should be separately conducted.

The comprehensive module is organized as following (as Figure 3.1):

- preprocessing (based on edge segmentation),
- lane detection and tracking (LDT system),
- lane marking recognition (LMR system),
- lane departure warning (LDW system).

Meanwhile, with the advantage of using MSER algorithm, the simplified module comprises the following parts (as shown in Figure 3):

- preprocessing (based on MSER segmentation),

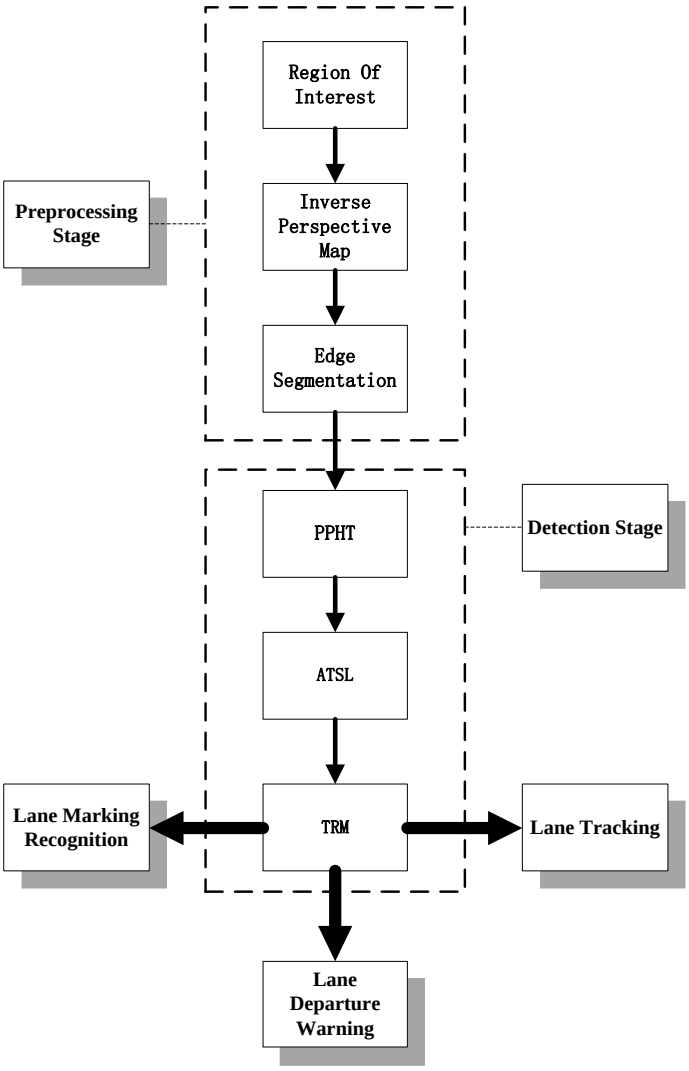


Figure 3.1: Flow Chart of Comprehensive Module

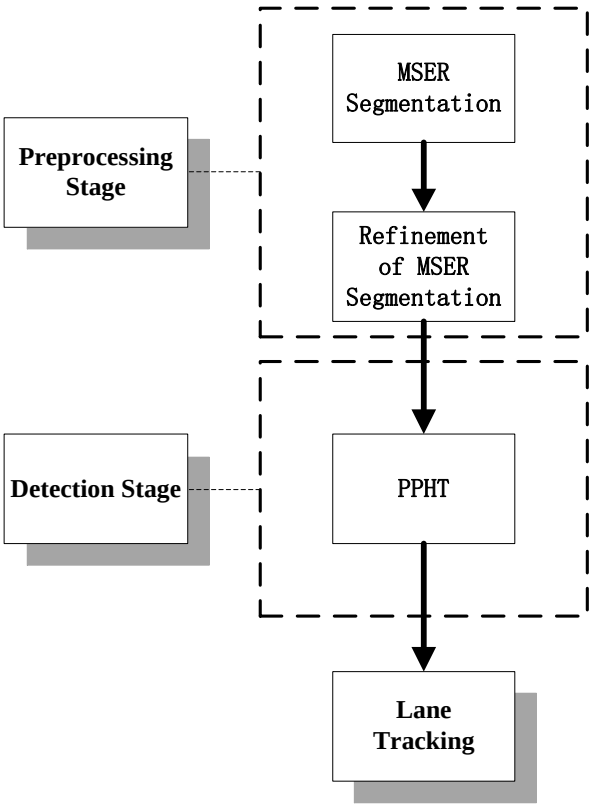


Figure 3.2: Flow Chart of Simplified Module

- lane detection and tracking (LDT system).

This chapter is organized as following: preprocessing, lane detection, lane tracking, lane recognition and departure warning. Both modules are discussed individually in Section 3.1.

3.1 Preprocessing

The reason for employing a preprocessing stage is because, the video clips taken in real scenario contain considerable outliers other than real lane markings. Frame images need to be properly prepared with outliers filtered out, depending on requirements of different computer vision systems. It can be realized by finding a Region of Interest (ROI) and applying segmentation. In this case, the concept of Region of Interest (ROI) is slightly different for edge segmentation and MSER segmentation. Edge segmentation needs to be applied after ROI extraction, followed by further refinement. On the contrary, MSER segmentation works based on MSER blobs, before offering refined results as ROI for the next detection stage.

Because of the different segmentation methods used in the comprehensive and simplified modules, the deployment of preprocessing stage are quite different for both modules.

3.1.1 Segmentation for Comprehensive Module (Edge-based)

When it comes to lane marking detection, edge-based segmentation is usually performed on the ROI in gray scale to enhance edges, and also to obtain pixels that belong to the desired lane markings. For most of the time, Inverse Perspective Mapping (IPM) needs to be involved before detection stage.

Projective Model

Worth to be mentioned here, in order for ROI extraction and IPM, a projective model is used, see Figure 3.3. We assume that the on-board camera is located at a height h above

a flat ground and a mapping needs to be constructed from a real world plane to an image plane. For that reason, we define a world frame F_w originated at the camera optical center, a camera frame F_c and an image frame F_i as follows (as Figure 3.3, [34, 66, 4]):

$$\{F_c\} = \{X_c, Y_c, Z_c, 1\}$$

$$\{F_w\} = \{X_w, Y_w, Z_w, 1\}$$

$$\{F_i\} = \{u_i, v_i, 1\}$$

The mapping between a 3D world point $(X_w, Y_w, Z_w, 1)$, which is located in world plane, and its 2D projection $(u_i, v_i, 1)$ can be expressed as in [34] or [66] following the equation:

$$(u_i, v_i, 1) = C.T.R(X_w, Y_w, Z_w, 1) \tag{3.1}$$

where C is the camera calibration matrix, T the translation matrix and R the rotation matrix.

We use the calibration method proposed in [128] to get the parameters of the camera calibration matrix C, i.e., focal length (f_x, f_y) and optical center (c_x, c_y), and extrinsic parameters such as pitch, yaw angle and the height h. The matrices R and T are defined according to the system requirement (having in mind that pitch angle, yaw angle and h are already known parameters). In the following section, these matrices (R, T and C) are defined according to the requirements of our system.

ROI Extraction Based on Projective Model

Region of interest (ROI) is always mandatory for edge-based segmentation. In order to not only reduce the surrounding noise but save running time, we need to narrow the detection range. In fact, electric poles, pedestrians, trees, and vehicles can all turn out to be extraneous entries for lane marking candidates. To avoid the interference resulted from surroundings or far-sighted view, and also for lower computational cost, detection area is better to be focused on a certain fraction of road surface.

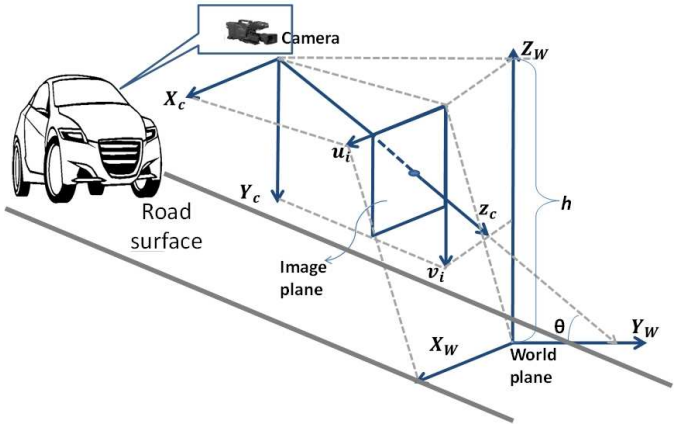


Figure 3.3: Projective Model

From another aspect, a minimum of detection distance in real world is needed for safety reason. As shown in Figure 3.4, ISO 17361:2007 and FMCSA-MCRR-05-005 legitimately allow the latest warning line be located $l = 0.3m$ outside the lane boundary, and that the minimum earliest warning line is $e = 0.75m$ inside the lane boundary (as shown in Figure 3.4). Accordingly, D.O.Cualain has already calculated a range of detection should be 16.18 m on the road surface in [34].

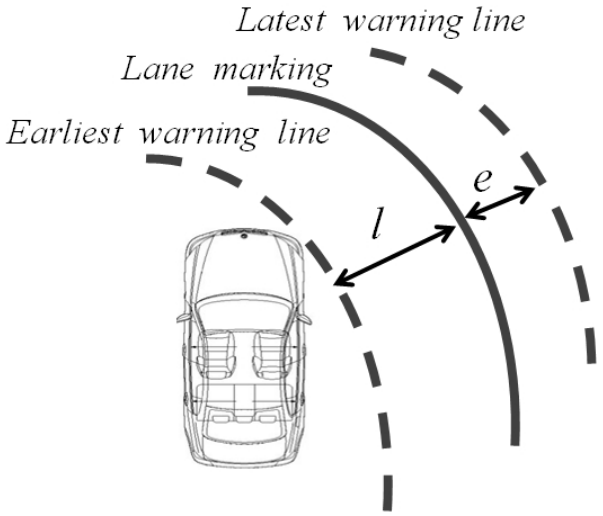


Figure 3.4: lane detection range for curve lanes

Different methods have been proposed to extract the ROI from the input images, as

in [34], [47] and [41]. Hanwell detects the position of vanishing point of a image in [47]. By doing this, the detection range (ROI) is set to be from camera to vanishing point. However, similar approaches only deal with straight lane markings. Because for curved lane markings, the position of vanishing point is not constant, it usually changes frame by frame.

To make our system more adaptable to real scenarios, and avoid collision on curbs or vertical departure on curve lane.(as shown in Figure 3.4), we need to take curved lanes into account. Hence for our scenario, we need to crop images at a distance of 16 m from the camera origin to the view in front of the car. Equation 3.1 can be used to project this distance to image plane.

Inverse Perspective Mapping (IPM)

The third step of conducting edge segmentation is to apply Inverse Perspective Mapping (IPM) on ROI, aiming at getting bird's eye view for road lines. In the literature, people use IPM to transform an image from real world plane to birds-eye-view (usually in 2D image plane), to look over the road surface from above.

There is one thing to be mentioned here, as the reason why IPM is required for lane marking detection. Because the core method of the detection stage (Section 3.2.1), Hough transform, asks for a very strict shape requirement. It is always good to make sure that the lines to be detected are parallel, straight, clear and continuous. Since birds' eye view images present all the lane boundaries as parallel and straight lines (as shown in Figure 2.3), this makes those strict requirement be better satisfied than just applying Hough transform on images in normal perspective. Further more, another benefit of IPM exists in the aspect of noise removal. By transforming an image from real world perspective into inverse perspective, we can narrow the image so that the detection range be concentrated and close to camera origin. Noisy points in far-sighted view and surroundings can be somehow reduced.

With the projective model expressed in Equation 3.1, to get the IPM of our system, we need to experimentally define camera parameter matrix C, rotation matrix R and

Translation matrix T (which are also used in [34], [66] and [4]) as follows:

Note: For this case, we don't need yaw angle because we set the camera right in the middle of a car so that it has no lateral bias to Y_w axis. Pitch angle θ can be set between 35.5 degrees to 90 degrees. Focal length(f_x, f_y) and optical center(c_x, c_y) are already obtained from camera calibration. So we have

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & sin(\theta) & -cos(\theta) & 0 \\ 0 & cos(\theta) & sin(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -h & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Edge Detection

After getting birds' eye view, some classical edge segmentation methods, such as Sobel, Prewitt, Robert or Canny operator, can be used to detect edges. It is common to know that Canny operator outperforms other operators such as Sobel and prewitt, for the general edge detection tasks. However, the authors in [87] have revealed that, for the purpose of lane detection, Canny filter is very sensitive to irrelevant objects as well as lane markings, which rapidly increases the number of false positives.

We compared Sobel and Canny as shown in Figure 3.5. It is obvious to find out from Figure 3.5 that Sobel performs better than Canny because of having much less noise (seeing from Figure 3.5(b) and (e)).

Figure 3.5: (a) and (d) are the same frame from Clip 4. (b) and (e) show the binarized picture of Sobel and Canny respectively. (c) is the final detection result of (b), while (f) is the result of (e).

Also comparing to other edge detection algorithms, the Canny operator stands out because of its high computational expensiveness. Conversely, the Sobel operator is less sensitive to noise and less complex than Canny, and it is able to detect the main edge points of markings. Hence Sobel operator is comprehensively better than the other methods as the edge detector for the proposed lane detection system. Further experimental results on the comparison of different edge segmentation methods are detailed in Section 4.5.

Sobel Algorithm As an edge detector, Sobel was first described and credited in a footnote "suggested by I. Sobel" in the book [31]. Dr. Irwin Sobel presented this algorithm at the Stanford Artificial Intelligence Project (SAIL) in 1968.

According to the presentation made by I. Sobel, the image function can be referred as a "density" function. For a 3×3 neighbourhood each simple central gradient estimate is a vector sum of a pair of orthogonal vectors. Each orthogonal vector is a directional derivative estimate multiplied by a unit vector which specifies the derivative's direction. A vector sum of the 8 directional derivative vectors is amounted by the vector sum of the 4 simple gradient estimations.

Hence for one pixel point, with its neighbourhood density value (usually as pixel values) indicated as following (Figure 3.6)

a	b	c
d	e	f
g	h	i

Figure 3.6: An example of the distribution of neighbouring density of point "e"

The magnitude of the directional derivative estimation vector V is defined for a given neighbour as

$$|V| = \frac{\Delta den}{dis}$$

where Δden stands for the density difference between two neighbours and dis equals to the distance of two neighbours.

Having all the neighbours group into pairs as $(a, i), (b, h), (c, g), (f, d)$, The direction of V is given by the unit vector (referred as $[1, 1]$, $[-1, 1]$, $[0, 1]$ and $[1, 0]$) to the relevant neighbours. The following the vector sum for the gradient estimate can be defined as following ([31]):

$$V = \frac{(c-g)}{4} \times [1, 1] + \frac{(a-i)}{4} \times [-1, 1] + \frac{(b-h)}{2} \times [0, 1] + \frac{(f-d)}{2} \times [1, 0]$$

Where we can have

$$V = [\frac{(c-g-a+i)}{4} + \frac{(f-d)}{2}, \frac{(c-g+a-i)}{4} + \frac{(b-h)}{2}] \tag{3.2}$$

It is conventional to divide the gradient by 4 in order to get the average gradient. However, these operations are typically done in fixed point on small integers and division

loses low order significant bits. Hence, instead of "divide by 4" (double shift to right), we "multiply by 4" (double shift to left). This will preserve low order bits.

So Equation 3.2 can be transformed as:

$$V' = 4 \times V = [(c - g - a + i) + 2 \times (f - d), (c - g + a - i) + 2 \times (b - h)] \tag{3.3}$$

where V' is the average gradient of V.

Figure 3.7 expressed weighted density summations for x and y components of the sobel operator. In this thesis, a point "e" can be recognized as an edge point if and only if:

$$|V'|^2 > T \tag{3.4}$$

where T is a pre-defined threshold.

1	2	1
0	0	0
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1

Figure 3.7: X and Y components of Sobel operators: the left graph is X component and the right graph is Y component.

3.1.2 Segmentation for Simplified Module (MSER-based)

MSER segmentation is implemented for the preprocessing stage of simplified module, as a substitution of projective model, Inverse Perspective Mapping (IPM) and edge detector. Different from edge segmentation, MSER is able to directly perform on the input

images and provides refined results as ROI for the next detection stage. Maximally Stable Extremal Region (MSER), which was proposed in [81], is an effective area-based segmentation method. Surprisingly, this algorithm has rarely been used for lane marking detection. In [109], a method of describing MSER patches using SIFT-based descriptors has been proposed, followed by a graphical model to localize lane markings. Different from [109], which requires unsupervised learning algorithm and off-line training, the proposed system directly takes advantage of MSER blobs for the feature extraction of lane markings.

MSER Algorithm

As discussed in [81], a gray image I can be described as a mapping: $(X, Y) \in Z^2 \rightarrow L$, where Z^2 stands for a set of pixel points (with the coordinate of (x, y)), and L stands for a set of luminance of pixel points ranges between $[0, 255]$. The term **region** used in MSER algorithm represents a contiguous subset S of the space Z^2 (especially for 4-neighbourhoods) which satisfies:

$$\forall p, q \in S, p, q \neq \emptyset, \exists series\{p, a_1, a_2, a_3, ..., a_{i-1}, a_i, q\}$$

, where a_1, a_2, a_3,...,$a_{i-1}, a_i \in$S, s.t. |p-a_1|=1, |a_i-q|=1, $\sum_{j=2}^{i}$ |a_j-a_{j-1}|=i-1.

A region S can be regarded as an **extremal region** when an arbitrary element of the region satisfies the mapping rule $S \rightarrow m \leq l$; where $m, l \in L$, m stands for the mapped value in L of an arbitrary element in S, and l is a pre-defined threshold which ranges between $[0, 255]$. A **stable extremal region** is an extremal region S that does not alter a lot while varies. Let:

$$R(S_l) = \{S_l, S_{l+1}, S_{l+2}, ..., S_{l+\Delta-1}, S_{l+\Delta}\}$$

which is a branch of tree rooted in S_l satisfied $S_l \subset S_{l+1} \subset S_{l+2} \subset ... \subset S_{l+\Delta-1} \subset S_{l+\Delta}$. In order to measure the stability of different extremal regions, the following equation needs to be used (as proposed in [81]):

$$q(l) = \frac{card(S_{l+\Delta} - S_l)}{card(S_l)} \tag{3.5}$$

where $card(S_l)$ represents the cardinality of a set S (one extremal region). An extremal region S_l can be chosen as a stable extremal region only in case that $q(l)$ of S_l is in the relevantly low level among the entire extremal regions. For certain $\Delta \in L$, **Maximally Stable Extremal Region** can be obtained by choosing the stable extremal region with the smallest $q(l)$ of all stable extremal regions.

Application of MSER Segmentation

As introduced in Section 2.1.1, as a part of edge detection, some smoothing methods (Gaussian Filter, Median Filters, etc.) are usually deployed before edge detection. This is designed to remove noise, blur the inner difference of desired regions and enhance regions with stable luminance. However, it makes edge segmentation might have a chance to miss some useful details, especially in the edge of prominent regions where luminance rapidly changes. To balance between keeping useful details and removing noise, we use MSER as the alternative of edge segmentation for preprocessing stage. Compared to segmentation based on edge information, an outstanding advantage of MSER is that, it only recognize stable extremal regions (e.g., lane markings, traffic signs, dark and stable parts of cars, etc.), which effectively filters out unpredicted noisy regions (such as potholes, obstacles on the road, etc.).

MSER blobs can be binarized for highlighting the stably extremal regions, as shown in Figures 3.9 and 3.8. However, due to heavy time-consumption and noise-sensitivity, MSER blobs need to be refined according to system requirements after being extracted. Apparently, the implementation of MSER algorithm is not enough to provide qualified images for detection because of containing noisy details as well as desired pixels. Besides, MSER experimentally turns out to be more computationally expensive than edge segmentation. This gives the necessity to the result refinement of MSER. In order to improve the time-efficiency and accuracy of the entire system, we propose a novel scanning method to reduce the pixel points in binarized MSER blobs, so that the number of input points for detection stage (pixel candidates for PPHT) could be decreased dramatically. This scanning method experimentally proves to allow the improvement of MSER and Hough transform results,

and the increasing of the detection rate of lane markings. (As discussed in Section 3.1.2.)

Refinement of MSER Segmentation

As stated in previous section, we find that there exists considerable noisy details as well as desired pixels in MSER-blobs (as shown in Figure 3.8). This requires some steps to refine the results of MSER segmentation. As a matter of fact, objects outside a lane are much more than objects (cars, pedestrians, etc.) within a lane (as shown in Figure 3.8). Therefore it is reasonable to say that lane marking blobs mainly distribute around the middle column (as the red dash line in Figure 3.8), comparing with other blobs which are outside lane boundaries.

Experimental observation reveals that the gap region between left and right lane markings is mainly composed of road surface, which has very weak luminance in gray scale images comparing with other objects. Since only stable extremal region can be extracted by MSER, noisy points within both lane markings can be rejected from MSER-blobs. This makes MSER different from edge detection, which extracts the information of both stable extremal regions and noisy regions. Accordingly we propose a scanning method performed on binarized pictures as described in Algorithm 1. The scanning process starts from the middle pixel of each row. It is carried out in left and right directions, respectively. MSER blobs are coloured in white, while the non-MSER areas are black. For each row in the image, when the first white pixel is encountered in both left and right area, the scanning process stops.

As the output of the proposed refinement step, MSER blobs can be shrunken into line pieces, which generally depict the contours of MSER blobs (as shown as 3.8). Since the scanning process starts from the middle column, these contours only belong to blobs that are close to middle column in left and right areas. Most likely, this method is able to portrait the contours of lane-marking-blobs which distribute near middle column. Moreover, the proposed scanning method makes the selected contours be one-pixel width, which weakens the interference from noisy blobs and makes qualified lines more prominent and easier to be detected by Hough transform.

Algorithm 1: Scanning Refinement of MSER

Input: Binarized images with MSER blobs

Output: Refined contours of MSER blobs

x and y: coordinates of a pixel point (x, y) in the binarized image

width and *height*: the width and height of binarized image

$P(x, y)$: pixel value of the point (x, y)

```
if Scanning for left area then
    for y = 0 to height do
        for x = width/2 to 0 do
            if P(x, y)! = 0 then
                x − − ;
                continue;
            else
                y + +;
                break;
else
    for y = 0 to height do
        for x = width/2 + 1 to width do
            if P(x, y)! = 0 then
                x + + ;
                continue;
            else
                y + + ;
                break;
```

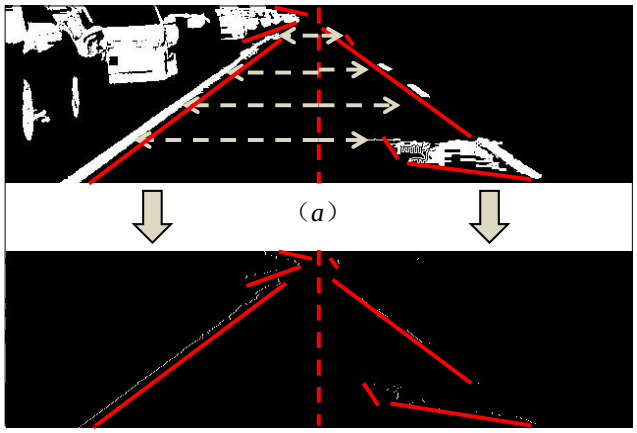


Figure 3.8: The working scheme of proposed scanning method. Red dash line in the middle indicates the "middle column" referred as above, while dash arrows indicate the scanning direction of each side (left and right side divided by middle column). For every row of each side, the scanning process stops when the first white pixel touched by the arrow (selected by scanning rule). Red solid lines generally depicts the entire contour after refinement.

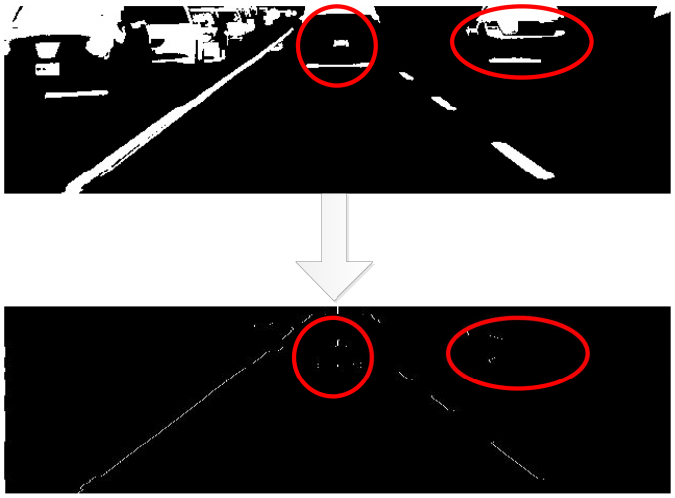


Figure 3.9: Two drawbacks of proposed scanning rule: red circular area: blobs between lane markings; red ellipse area: blobs between dashes.

However, there exists two drawbacks in the proposed method. First, for the area between left and right lane markings on the road, real scenarios might inevitably bring in cars or some stains which are likely to be detected as MSER-blobs (as shown as red circular area in Figure 3.9). As a result of this, the proposed scanning method might take the contours of noisy MSER-blobs as detection candidates, and feed those noise together with real lane marking pixels into detection stage. To eliminate noisy blobs mentioned above, as shown in Figure 3.8, the scanning method only selects at most two pixels in a row (one pixel per area), which makes the output lines have only one-pixel width. This significantly weakens the noisy contours (shorter and less straighter than lane markings) between lane markings, and also makes the continuous contours of lane marking blobs stands out from the background. Additionally, Noisy blobs can further be removed by PPHT with the help of proper threshold of length and angle, which is detailed in Section 3.2.1. As the second drawback of the scanning method, blobs outside lane boundaries (as shown as red ellipse area in Figure 3.9 may be encountered when scanning in rows between dashes. This might bring noisy contours and false positive results for dashed lane marking detection. Similar to the solution of first drawback, experimentally PPHT proves to be able to handle the above issues by angle and length threshold of line candidates, which is discussed in Section 3.2.1. By an appropriate threshold, lines located in irrelevant regions can hardly be selected as lane marking candidates.

3.2 Lane Detection

After preprocessing stage, detection stage is initialized by Hough transform (in this thesis we use Probabilistic hough transform, which is refered as PHT). Because of the benefit of preprocessing stage based on MSER segmentation, the simplified module performs well with Hough transform without refinement afterwards. While for comprehensive module, two refinement steps need to be conducted sequentially, which are angle thresholding and segment linking (ATSL) and trapezoidal refinement method (TRM). It is noticeable for the comprehensive module, where edge segmentation is used for preprocessing, that PHT is applied on bird's eye view. Hence, there are some works to be done between PHT and

ATSL, which is to transform the bird's eye view into real world plane (this transformation is very straightforward to get by applying inverse perspective mapping on bird's eye view). After getting real world plane image, the novel schemes of ATSL and TRM is proposed for the sake of refining PHT results.

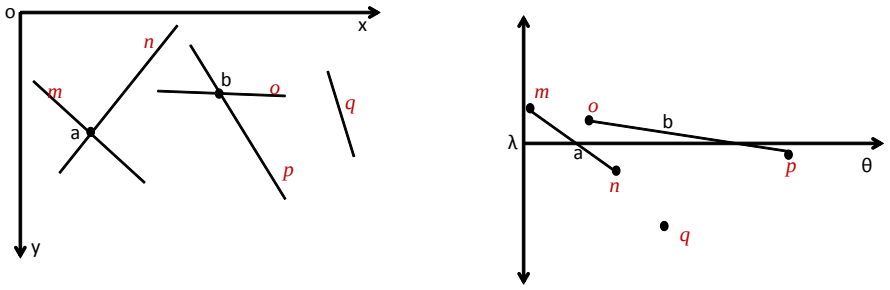
3.2.1 Probabilistic Hough Transform

Probabilistic Hough transform (PHT) is one of the most popular types of the classical line detection algorithm, Hough transform (HT), which was proposed [50] by Hough in 1962 and firstly used in Research in 1972. Hough transform is usually used to detect lines and circles, and it is used as the core method of lane marking detection in [34], [118] and [60]. Hough transform gives not only robust detection under noise but partial occlusion in many situations. The core formula of HT is

$$\lambda = xcos(\theta) + ysin(\theta) \tag{3.6}$$

λ is the length between the origin and the pedal of detected line and θ is the angle of its perpendicular line. A single point in xy-space corresponds a line in (λ, θ) space (as a and b in Figure 3.10(a) and Figure 3.10(b)). Similarly, A line in xy-space corresponds an intersection point which holds many lines in (λ, θ) space (as m,n,o,p and q in Figure 3.10(a) and Figure 3.10(b)).

In the literature, HT is usually referred as standard Hough transform (SHT). SHT works only based on Equation 3.6 and is different from other Hough transform (i.e. PHT and RHT). The detection scheme of SHT is firmly the same as the original idea proposed in [50], which can be generalized as following: an accumulator is constructed in hough space with respect to λ and θ. Every pixel points in the xy-plane image is selected to vote in the accumulator. Pre-defined thresholds are used to choose line segments which correspond to enough numbers of voting points in hough space (as in accumulator). Apparently, for pixel points that are intersections of several lines or just belong to one single line, this voting scheme is necessary; otherwise, for points that are noisy points (which don't belong to any lines or segments), being selected to vote in the accumulator is meaningless and a waste of time.



(a) XY-space (b) Hough space

Figure 3.10: An example of line distribution in XY-space (a) with the corresponding results in Hough space (b)

Analysis of PHT

To increase the time efficiency of SHT, Kiyara proposed a method in [61], which is called Probabilistic hough transform (PHT). Probabilistic Hough transform (PHT) improves the process of SHT by minimizing the number of voting pixels. Kiyara mathematically proved that, it is possible to obtain line results identical to those of the standard Hough transform if a reasonable fraction p of pixel points are chosen for the voting process, instead of choosing all the pixel points in image (as in xy-plane).

There are some difference between Standard Hough Transform(SHT) and Probabilistic Hough Transform(PHT).

***a.**)SHT is the most commonly used method for lane detection, while PHT is rarely used by researchers(only by Hota in [49]). This is because, as the standard form of Hough Transform, SHT provides comprehensive results and describes every lines with different λ and θ.*

***b.**)SHT and PHT produce different results. SHT produces (λ, θ) as results while PHT produces the coordinates of both ends of a line in xy-space.*

***c.**)PHT has the kernel of SHT, which yields, PHT randomly samples the starting and ending points based on the lines detected by SHT, while SHT only provides λ and θ of a line.*

***d.)** With the starting and ending points sampled in c.), PHT then threshold the length of line segments in order to eliminate weak line candidates.*

Probabilistic hough transform is initialized by randomly selecting a subset of points, followed by a standard Hough transform performed on the subset. In [61], the edge map with lines to be detected is considered as a "noise-dominated" stochastic model. This is because for the purpose of line detection, only edge points which form as lines are the results; most of the points within image area belong to noise. Assume we have an edge map with M points, S points belong to lines and $N = M - S$ noisy points. Suppose we sample m points out of M points, with selecting s points as the line results and n noisy points.

At the peak of accumulator (a pair of (λ, θ) with the most voting points in hough space), the random variable s distributes binomially. Hence, the probability that s belongs to a line is

$$P(s) = \binom{m}{s} (\frac{S}{M})^s (\frac{N}{M})^{m-s} \tag{3.7}$$

Also, to demonstrate the applicability of PHT, a random variable n^* needs to be introduced to represent the contribution of a selected noisy points to a certain location in accumulator array.

$$P(n^*) = \sum_{n=0}^{m} P(n) P(n^*|n) \tag{3.8}$$

where $P(n)$ is the probability of n selected noisy points and $P(n^*|n)$ is the conditional probability of n^*. Apparently $P(n)$ is also binomial,

$$P(n) = \binom{m}{n} (\frac{N}{M})^n (\frac{S}{M})^{m-n} \tag{3.9}$$

If m is large enough such that

$$m \frac{S}{M} \frac{N}{M} \ll 1 \tag{3.10}$$

the Gaussian approximation of Equation. 3.7 holds, with the expectation

$$\eta_s = \frac{m}{M} S \tag{3.11}$$

and we have the standard deviation

$$\sigma_s = \sqrt{m \frac{S}{M} \frac{N}{M}} \tag{3.12}$$

Similar to s, we can get the Gaussian approximation of n, where we have $\sigma_s = \sigma_n$ and the expectation

$$\eta_n = \frac{m}{M}N \tag{3.13}$$

It is also pointed out in [61] that the noisy points are uniformly distributed in image, which leads to non-uniform noise in the accumulator array. For a certain location (λ_0, θ_0)in the accumulator array, the conditional probability of n^* is

$$P(n^*|n) = \binom{n}{n^*} p^{n*}(1-p)^{n-n^*} \tag{3.14}$$

where p is the probability that a selected noisy point contributes to vote in a certain location (λ_0, θ_0) in accumulator array. p can be treated as the fraction of the image area which is projected onto a segment of length $d\Delta\lambda$. For $d\Delta\lambda \ll 1$,

$$p = \frac{2d\Delta\lambda}{\pi}\sqrt{1-\rho^2} \tag{3.15}$$

since $\sigma_n \ll \eta_n$ and $P(n^*|n)$ is not extremely sensitive to small variations in n, it is appropriate to approximate $P(n)$ in Equation. 3.9 as an impulse function at $n = \eta_n$. Hence, we have the Poisson approximation of $p(n^*)$ according to Equation. 3.8

$$p(n^*) \approx \exp^{-\eta_n p} \frac{(\eta_n p)^{n^*}}{n^*!} \tag{3.16}$$

Based on the comparison of Equation. 3.16 and 3.7, it can be seen that the random variables s and n^* are almost independent to each other. This means that, of the randomly selected m points, line results (represented by s line edge points) do not change a lot because of the contribution from n noisy points. Actually, successful experiments conducted in [61] with p as low as 2% revealed that, the poll size (a fraction of edge points that being randomly selected) is a parameter critically influencing the performance of probabilistic Hough transform.

However, the experiment in [61] still has one fatal drawback. For the noise-dominant case, only one single line is merged in the edge map, surrounded by some isolated noisy points. It is easy to know the number of points belonging to the line. Hence for the key step of Hough transform, thresholding in accumulator array, the author in [61] proposed

a formula to solve the poll size (sample fraction of all edge points)in accumulator array to identify a line, which requires a priori knowledge of the length of the line (number of points which belong to the line). This formula can be exclusively used in the experiment of [61]. However, it is almost impossible to know the length of all lines in real scenario, where all image frames are collected in real time.

Progressive Probabilistic Hough Transform (PPHT)

Regardless of the drawback of the experiment that Kiryati did in [61] in attempt to validate PHT's performance with priorly knowing the length of lines, PHT can be considered as a very efficient theoretical approach. Researchers hold the strong point of view that, PHT can be optimized appropriately in real scenario to detect lines. In 1999, J. Matas proposed progressive probabilistic Hough transform (PPHT) in [80] in order to solve the problem of randomly sampling points without knowing line length. PPHT has been commonly accepted as one of the best line detection methods based on PHT theory. PPHT proceeds as follows and can be demonstrated as Figure 3.11.

1. Randomly, a new point is selected for voting in the accumulator array, with contributing to all available bin (as referred in [80], bin stands for a pair of (λ, θ)). Then remove the selected pixel from input image.

2. Check if the highest peak (the pair of (λ, θ) with the most voting points) in the updated accumulator is greater than a pre-defined threshold $th(N)$. If not then go to Step 1.

3. Find all lines with the parameter (λ, θ) which was specified by the peak in Step 2. Choose the longest segment (which can be denoted by starting point Pt_0 and ending point Pt_1) of all lines.

4. Remove all the points of the longest line from input image.

5. Remove all the points of the selected line in Step 3 $(Pt_0 - Pt_1)$ from the accumulator, which means those points do not attend any other voting process.

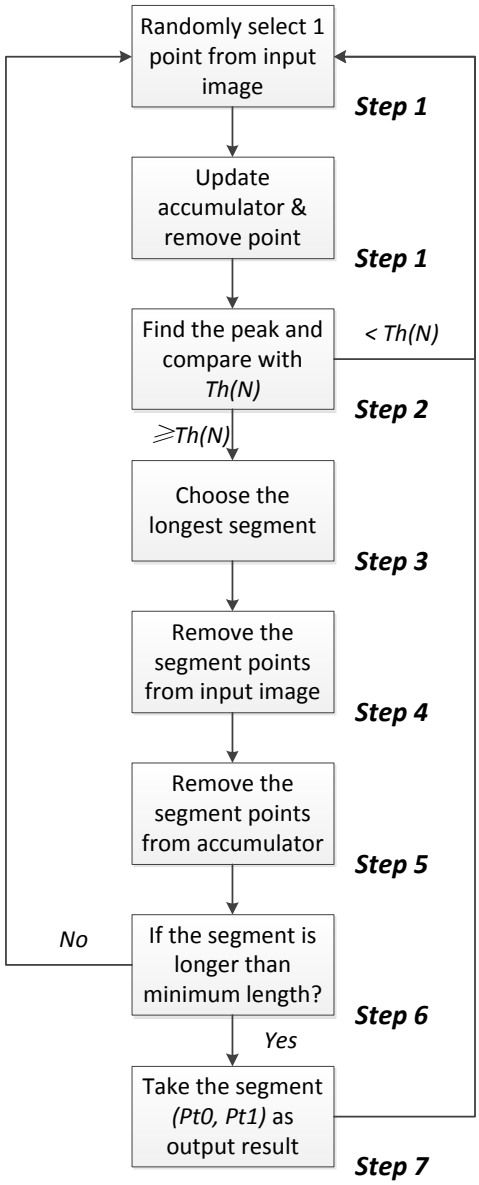


Figure 3.11: The flowchart of progressive probabilistic Hough transform (PPHT)

6. If the selected segment is longer than a pre-defined minimum length, then take the segment $(Pt_0 - Pt_1)$ as one of the output results.

7. Go to Step 1.

As we know, a full Hough transform (as standard Hough transform which transverse all points within input image) needs a stopping rule to avoid operations for all points. Without applying a full Hough transform, to avoid adding a stopping rule, PPHT stops when either the points have voted or have been assigned to a feature (recognize a point as belonging to a qualified line segment and remove this point from input image and accumulator array). Apparently this allows only a small fraction of points to be the candidate and significantly reduces the computation cost.

In terms of error estimation, without a stopping rule, the difference between the PPHT and the standard Hough transform (SHT) lies mainly on the number of false positives (some noisy points been taken as lines). As we can see from Figure 3.11, PPHT possesses the basic voting points as SHT. Hence if a line is detectable by the SHT, it should also be detected by the PPHT. False negatives (missing lines) are due to failed edge detection or previous detected results.

In this thesis, PPHT is used instead of SHT. Except for the purpose of minimizing computation cost, some reasons are explained as following: As talked above, SHT presents (λ, θ) of every detected line. However, this yields SHT is much too sensitive to all straight lines(regardless of some unwanted lines with short length), it also consumes much more time than PHT in order to present all lines. Plus, lane detection has its own requirement, which is, detector should only respond to lines with specific characteristics (lane markings). Recalling the two drawbacks in Section 3.1.2 (as shown in Figure 3.9), sometimes vehicles or part of the contours of surroundings appear in region of interest which can be erroneously detected by HT. Those lines with a variety of directions have short length comparing with real lane markings, hence are not eligible to be chosen as a detected lane marking candidate.In PPHT, constraint is given by setting a minimum line length, which only takes lines with qualified length as output.

For the comprehensive module, another reason to choose PPHT is because, to perform further refinement (ALS and TRM), starting and ending points are needed in next stages. One of the benefits of using PPHT is that, PPHT reduces computation cost by minimizing the number of voting points, hence common improvements that reduce the number of voting points (e.g., binarization based on gradient information) do not conflict with the result of PPHT. After applying PPHT on birds' eye view, some parallel straight lines are obtained. Of those lines, we have not only lane marking candidates, but some unwanted lines. It is necessary to apply a refinement stage on real world plane images, to further remove outliers and refine detection results. This is performed by Segments Linking, constructing trapezoid and other refinement methods, which is introduced in Sections 3.2.2 and 3.2.3.

For the comprehensive module, experimentally we found PPHT performs better than SHT and PHT on lane detection. Also for the simplified module, as the only step for detection stage, PPHT fulfils the task of lane marking extraction. Comparative results on both modules are introduced in Sections 4.3 and 4.4.

3.2.2 Angle Threshold and Segment Linking

As seeing from Section 3.2.1, PPHT does not take angles (θ) into account when thresholding the number of voting pixels in the accumulator, which may bring some unwanted lines together with qualified lines in real world plane(as shown in Figure 3.12(a)). At this step, refinement is to threshold and sift out left and right lane markers, which can be the input of the next step, trapezoidal refinement method (TRM). ATSL consists of three steps as following:

Angle Threshold

At this step, we need to firstly divide the ROI into right and left areas. Next, as we can obviously see from Figure 3.12(a)(b), experimentally we found that, suitable lane marking candidates within left and right areas should form angles with respect to bottom line of ROI as required as following:

$$20^\circ \leq \alpha \leq 70^\circ$$

$$20^\circ \leq \beta \leq 70^\circ$$

where α and β are the angle difference of lines with respect to the bottom of ROI in left and right area respectively. lines which do not satisfy the requirements above need to be removed(as shown in Figure 3.12(b)).

Segment Linking

In [83], Li Shunming et.al proposed a linking rule for curve lines to assemble relevant line segments and weaken unwanted lines. This is performed by examining their angle difference and distance to each other, then linking relevant lines into longer ones. For our case, we modified this method so that it adapts to straight lines and PPHT. Compared to method in [83], the proposed one is more efficient and adaptable to straight lines with starting and ending points. We can use this linking rule to not only strengthen potential line segments, also reduce the numbers of lane marking candidates. As shown in Figure 3.12(c), an arbitrary pair of line segments from the results of previous step is selected. We threshold the angle difference between two lines and the distance between one's ending point and the other's starting point. If the selected line pair has angle difference and point distance within a range, we need to take these two lines as relevant line pair and link them into a longer line (Algorithm 2). By using Algorithm 2, we make the lane marking candidates more prominent, clearer and longer, so that real location of lane markings could more likely be extracted and refined.

Optional Refinement

The third step can be optional and ignored if the system requires multiple lane detection. Most of the time in real scenario, usually drivers only need to focus on the lane where they are currently driving on, especially when a need of Lane Departure Warning is required. Considering the proposed system with a purpose of Lane Departure Warning, detection

Algorithm 2: Segment Linking Rules for PPHT

l_k: the k_{th} line segment

P_{ks} and P_{ke}: the starting and ending points of the k_{th} line segment

$\theta()$: the angle difference of two arbitrary lines

$d()$: the distance of two arbitrary points

TN: Total number of line segments

AT: Angle threshold

DT: Distance threshold

Event: Decide if lines are relevant to each other and link relevant lines

for *a = 1 to* TN **do**
- **if** l_a *is not labelled as linked* **then**
  - **for** $b = (a+1)$ *to* TN **do**
    - **if** l_b *is not labelled as linked* **then**
      - **if** $0 \leq \theta(l_a, l_b) \leq AT$ *and*
      - $0 \leq d(P_{ae}, P_{bs}) \leq DT$ **then**
        - link P_{as} with P_{be} into a line
        - Take this line as l_a
        - label l_a and l_b as *linked*
      - **else**
        - Return
    - **else**
      - Return
- **else**
  - Return

scheme needs to focus on only one lane. Hence, only one or two lane markings are needed to clearly mark this one lane (some lanes are formed with two lane markings, or single lane marking on one side and curb on the other side).

Therefore, for the case shown in Figure 3.12(e), we only need to select two lane markings (one lane marking for left area and one for right area) from lines with qualified angles and length (which are the results after angle threshold and Segment linking). Within each area (left and right areas divided by middle line in Figure 3.12(d)), the line with its both ends (starting and ending points) horizontally (by saying horizontally, it means in the direction of x-axis in image plane) closest to the middle line comparing with the others is chosen as the output of ATSL. Since the noisy lines and outliers with unqualified angles and length are removed after the first two steps of ATSL, experimentally we found this pair of lines chosen in optional refinement can most likely be lane marking candidates for the next steps.

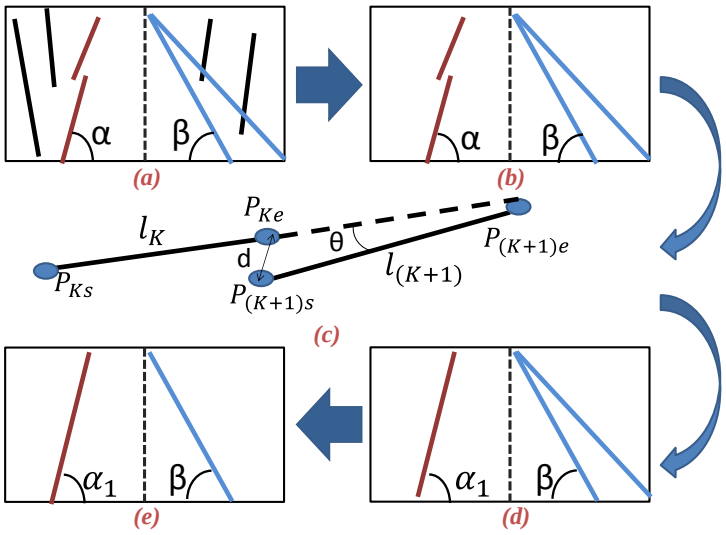


Figure 3.12: Angle Threshold and Segment Linking: (a) Results of PHT; (b) After angle threshold; (c) Apply segment linking; (d) Results after segment linking; (e)After choosing the line pair closet to middle line.

3.2.3 Trapezoidal Refinement Method

After ATSL, we have some qualified line pairs as lane marking candidates. For this stage, Trapezoidal Refinement Method (TRM), we use the line pairs of previous stage as input. Taking the possible detection failures in previous steps into account, somehow these two lines do not fit well with real location of lane markings, which means, starting and ending points are sometimes a few pixels away from real location (Figure 3.13).

Also in order to extract the colour information of lane markings for the sake of lane recognition (as required in Section 3.4), more pixel information is needed. Hence trapezoid is constructed for each line as follows. We choose the starting and ending points of one line as the middle point of the top and bottom bases of this trapezoid, respectively. These bases are linked together to form the lateral sides of the trapezoid (as shown in Figure 3.13). The top base of the trapezoid should be shorter than the bottom base, since a segment line (which shapes rectangular and is perpendicular to the horizontal axis of the optical coordinate of camera) appears as a trapezoid in images, as a result of perspective effect. Experimentally, the length of the bottom base of the trapezoid should be two times of the length of top-base. Both top and bottom bases of the constructed trapezoid are set to 20 an 40 pixels long for our case, respectively. In this way, the trapezoid contains more ground truth pixels than just covering the line segment which links starting and ending points.

After getting the trapezoid, we compute the average pixel value of this area, for 3 channels (R, G and B). Known to be much brighter than road surface, lane markings usually contains more yellow and white elements than background (which is, in this case, the rest part of constructed trapezoid). Averagely, R and G components of each pixel value are higher than the pixel values of background. Based on the fact above, for every y step, we scan the pixels from left to right with a 3×3 block (Figure 3.13). We determine if a point can be taken as lane marking pixel or not, by comparing the average value (for R and G channel) of a 3×3 block centering at each point in the trapezoid with the average value of the whole trapezoid. For one pixel, if the average value of its 3×3 block is greater than average of the trapezoid area, it can be taken as lane marking pixel, otherwise not.

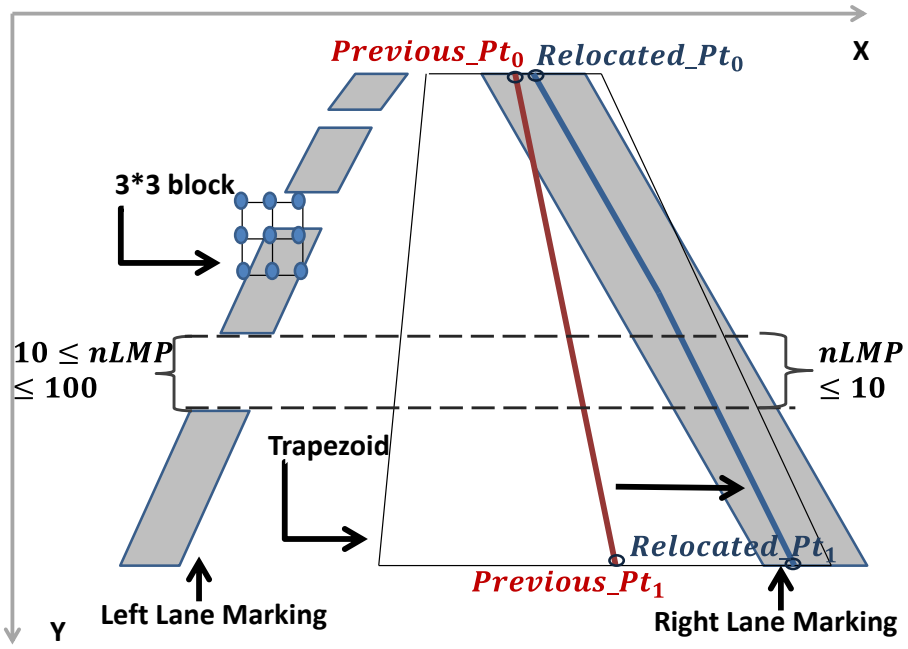


Figure 3.13: Trapezoidal Refinement Method

Experimentally, we found there are four situations about the location of real lane markers and our constructed trapezoids, as shown in Figure 3.14. The idealist situations are as Figure 3.14(b) and Figure 3.14(d) shown, trapezoid fully covering lane marking (blue parallelogram in image) and trapezoid being covered by lane marking. Figure 3.14(a) is the most common situation, partly covering, while Figure 3.14(c) is the worst situation, trapezoid and lane marking totally depart from each other. For the worst situation, because a detection failure (wrong starting and ending points) is brought in by previous steps, the proposed TRM does not work well with this situation, which actually does not often happen in our experiment. As demonstrated in Algorithm 3 and Algorithm 4, assuming we have both ends of a line, p0 and p1. By using TRM we can make detected points better fit with ground truth(Figure 3.13). The result of TRM is also used to serve the Kalman tracker with refined starting and ending points as explained in the following Section 3.3.

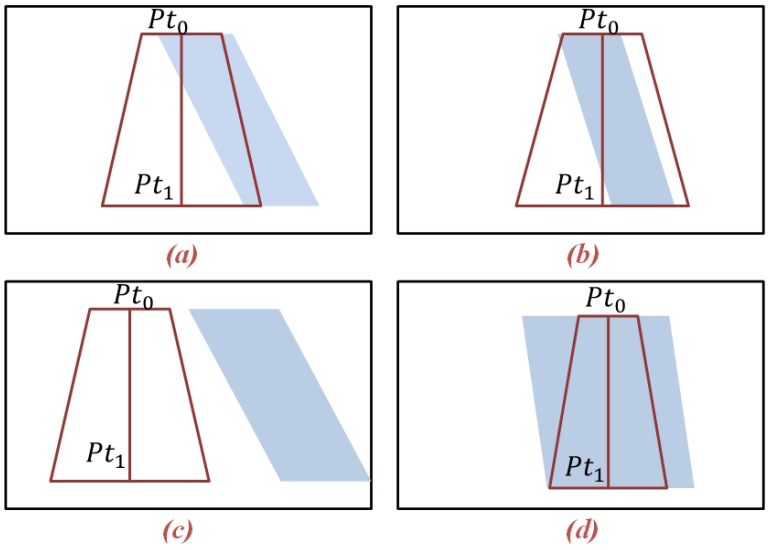


Figure 3.14: Location of Trapezoid and Lane Marking(as shown by blue parallelogram): (a):Trapezoid partly covers lane marking; (b):Trapezoid fully covers lane marking; (c):Trapezoid totally depart from lane marking; (d):Trapezoid fully covered by lane marking

Algorithm 3: Average of Trapezoid

$P0$: the top point(pt_0) of one line

$P1$: the bottom point(pt_1) of one line

t: the width($x_{max} - x_{min}$) of every y step of the trapezoid

AVG(RGB): get the average value of RGB for trapezoid

$P(RGB)$: get pixel values of RGB for one point

count: the amount of point in the trapezoid

Event: Get the Average pixel value of trapezoid

for $y = P0_y$ *to* $P1_y$ **do**

    t = 20 + 21 $\times \frac{y - P0_y}{P1_y - P0_y}$

    for $x = (\ (y - P0_y)\frac{P0_x - P1_x}{P0_y - P1_y} + P0_x - t)$ *to* $((y - P0_y)\frac{P0_x - P1_x}{P0_y - P1_y} + P0_x + t)$ **do**

        count++

        Get the pixel value for 3 channels(R,G and B) of a point $P(RGB)$

        SUM(RGB)= SUM(RGB)+$P(RGB)$

AVG(RGB) = SUM(RGB) / *count*

3.3 Lane Tracking

For most highway and ideal urban scenarios with smooth road texture, lane markings and background (road surface) are clearly distinguished from each other. This makes it easy for Hough transform and TRM to do their jobs. However, some challenges might come from situations with rough roads, or cloudy and rainy weather, detection result does not fit well with ground truth. Experiments have shown that the addition of a lane tracking stage after lane detection helps in coping with this issue. In order to improve the efficiency and robustness of lane detection system, Kalman filter (KF) is implemented as a tracker for both comprehensive and simplified modules.

3.3.1 Kalman Filter

In 1960, Kalman proposed Kalman filter (KF) in [57]. Working the way of a linear dynamic system based on Markov Chain model, KF predicts the post state based on previous state and current measurements, with updating the covariance matrix of state and measurement. Iteration keeps running by feeding corrected state matrix to next instance. For the case of lane tracking, previous researches have been done by using Kalman filter (KF) or Particle filter (PF) to track different parameters ([92], [38], [82], [97] and [106]). Also, KF has been used for SHT and PHT in [117] and [49], respectively.

Experimental results in comprehensive module have shown that PPHT has better performance than SHT for our case. Therefore in this thesis, KF is chosen to track both ends of each line, which is referred as starting and ending points of lane markings determined by PPHT, ATSL and TRM. As a contribution to robustly improve the fitness to ground truth, the correction results of KF is fed back to TRM for better construction of the trapezoids.

Kalman Model

According to [57], the inner connection between the state of a model at time k and the state of $(k-1)$ can be described as follow:

$$x_k = F_k x_{k-1} + B_k u_k + W_k \tag{3.17}$$

Where F_k is the state transition matrix which needs to be applied to the previous state vector x_{k1} in order for updating; B_k is the control matrix to update the external control vector u_k; W_k is the process noise with a covariance of Q_k, which is:

$$W_k \sim N(0, Q_k)$$

At time k, the measurement vector z_k of the state variable x_k can be acquired according to

$$z_k = H_k x_{k-1} + v_k \tag{3.18}$$

where H_k is the observation matrix; v_k is the measurement noise with a covariance of R_k, which is:

$$v_k \sim N(0, R_k)$$

. The initial value of state variables and the noise are all assumed to be discreetly independent to each other.

The mechanism of Kalman filter can be divided into two steps: prediction (also referred as estimation) and updating.

Prediction

In the prediction step, state variables are initially estimated by Kalman filter, with also initializing the process noise, priori estimation error. Meanwhile, the system keeps monitoring measurement information and feeding measurement matrix together with measurement noise into updating step.

The priori state estimate can be described as:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{3.19}$$

and the priori estimation error covariance:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{3.20}$$

Updating

At updating step, results of prediction need to be updated based on the computable weight of estimation results and measurement results (by utilizing the innovation which indicates the certainty). Trust from the system depends on the certainty, which will recursively influence the next instances.
Innovation:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{3.21}$$

Innovation covariance:

$$\mathbf{S}_k = H_k P_{k|k-1} H_k^T + R_k \tag{3.22}$$

Besides, Kalman gain and posteriori estimation error covariance need to be updated as well, contributing to compute the posteriori state variables, which evolves from the initial state variables and becomes the priori estimation of state variables for next instance.
Optimal Kalman gain:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{3.23}$$

Posteriori (after being updated) error estimation covariance:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{3.24}$$

Posteriori (after being updated) state estimation:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{3.25}$$

It is pointed out in [57] that, the formula for the updated estimation and error covariance above are valid only under the condition of the optimal Kalman gain (Equation 3.23). Thanks to the inherent recursive ability, Kalman filter is able to run in real-time by taking advantage of measurement and estimation results.

3.3.2 Lane Tracking with Kalman Filter

Two kalman trackers are utilized respectively for right and left lane markers, with respect to the starting point $Pt_0(X_{Pt_0}, Y_{Pt_0})$ and ending point $Pt_1(X_{Pt_1}, Y_{Pt_1})$ (as depicted in

Figure 4). Notable in this case, the measurement noise (R_k) and process noise (Q_k) which result from lane detection can be deemed as Gaussianly distributed, which makes the lane tracking for the comprehensive module to be based on Gaussian stochastic process. Also because there is not input from external control in the proposed system, the control vector u_k and control matrix B_k in Equation 3.17 will not be taken into account.

Recalling Equation 3.17, where x_k is the state vector, F_k is the state transition matrix. We define the state vector as:

$$x_k = [X_{Pt_0} Y_{Pt_0} X_{Pt_1} Y_{Pt_1} X'_{Pt_0} Y'_{Pt_0} X'_{Pt_1} Y'_{Pt_1}]^T$$

where X' and Y' are the derivative form of X and Y.

Experimentally, yielding to the best tracking performance for our case, the state transition matrix can be defined as following:

$$F_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$u_k = 0$$

The coordinates of Pt_0 and Pt_1 are taken as measurement z_k for every instance:

$$z_k = [X_{Pt_0} Y_{Pt_0} X_{Pt_1} Y_{Pt_1}]^T$$

3.4 Lane Recognition

Roadway users (drivers, motorist and pedestrian) can read important informations from pavement markings, which can be divided into three categories: word-marking, pictogram-marking and line-marking. Word-markings are usually used to deliver a text message to

drivers, for example, the word BUS. To express messages for drivers, Pictogram-markings are well designed ideogram and present as understandable graphics, e.g. bicycle lanes or bus lanes. Line markings have very essential shapes which can be used to manage the traffic. Along road edges and between lanes, line markings are used to guide traffic, keep vehicles in line and avoid collisions. Marking colors (i.e., yellow or white) and forms (solid or dashed) deliver important messages together. For instance, Yellow lines indicates opposite directions of two lanes. Switching from one lane to adjacent lanes is prohibited if a solid yellow line exists between two lanes, while a dashed yellow line allows drivers to switch between lanes if they need to. On the other hand, white lines are usually painted on a multi-lane roadway, which separate vehicles moving in the same direction. Solid (sometimes in double) white line forbids switching between lanes, while lane switching is permitted to across a dashed white line. In the comprehensive module, TRM combines the detection and recognition of lane markings together, by making use of colour information. The following sections will elaborate the proposed lane recognition scheme.

3.4.1 Solid and Dashed lines

The values of LMP and $nLMP$ (Algorithm 4) are important measurement used to distinguish between solid and dashed lines. A solid line contains lane marking pixels as the majority of y-steps (which are marked as LMP in Algorithm 4, lines 15-17), while a dashed marking contains a moderate number of lane marking pixels and some blanks between dashes (referred as nLMP, lines 24-25 in Algorithm 4). The value of nLMP actually reflects a number of non-lane marking pixels.

However, in some extreme situations the recognition of nature of the marking can be challenging. For instance, solid markings on eroded surfaces can be recognized as dashed lane markings. Also, the shadow of cars, trees or other road-objects can inevitably affect the quality of detection, thus solid markings can be recognized as dashed markings. Moreover, if a detected lane marking does not contain enough lane marking pixels (LMP), it will not be selected as a lane marking. In order to overcome these problems, and to avoid a possible erroneous recognition, massive experiments have been conducted to determine

the threshold with respect to the values of nLMP and LMP, as described in lines 26-35 of Algorithm 4.

Experimentally we found that if a line contains very few lane marking pixels ($nLMP \geq 100$ or $LMP \leq 50$), in this case the line is deemed as a non-lane marking, as stated as lines 26-35 in Algorithm4. If $nLMP \leq 10$, which means a line contains a certain amount of lane marking pixels (as described in lines 32-33 of Algorithm 4), this line is considered as a solid marking. On the other hand, a lane marking can be recognized as dashed if $10 \leq nMLP \leq 100$ (as shown in lines 29-30 in Algorithm 4).

3.4.2 White and Yellow lines

A lane marking can only appear to be yellow or white, what is needed in colour recognition is to find a way of quantitatively defining yellow and white in real scenarios. It is common to know that, the yellow and white lines on the road do not appear to be strictly yellow and white. Hence the problem of this part lies mainly on distinguishing between yellow and white, with respect to RGB value. In real scenarios, yellow and white sometimes poorly contrast to each other, because of rough texture of road surface, lightening or other complex factors. Hence the boundary of yellow and white can be vague and ambiguous.

In fact, the RGB value of ideally yellow and white pixels are $(255, 255, 0)$ and $(255, 255, 255)$, respectively. From this fact we can see, the value of B channel dominantly determines the difference between yellow and white. The R and G value do not significantly influence the colour appearance when the pixel is either white or yellow. Further more, if the B value of a pixel is much less than the average of R and G value, it is most likely to be taken as a yellow pixel, otherwise, if the B value is much more than the average of R and G value, it can be grouped into white pixels.

Experimental results on massive video clips taken from Ottawa have shown the evidence that, white pixels should have B values more than 4/5 of the average of R and G values (as lines 18-20 of Algorithm 4); while yellow pixels have B values less than 4/5 of that average (as lines 21-23 of Algorithm 4). As shown in Figure 3.15, with the proposed scheme, we successfully distinguish solid and dashed lines, as well as yellow and white lines, by putting

different colour and text messages on them. "SY", "DY", "SW" and "DW" stand for "solid yellow", "dashed yellow", "solid white" and "dashed white", respectively.

Figure 3.15: Lane Recognition on Highway and Rural Area in Ottawa

3.5 Lane Departure Warning

Known as an essential part of Intelligent Transportation System (ITS), Lane Departure Warning (LDW) plays a vital role in the comprehensive module. The lane departure warning scheme is built up based on previous stages (TRM results which can be updated by lane tracking stage). Lane departure means a situation where a moving car departs from current lane or has the tendency to go across lane markings. As a result, the driver or monitoring system will see only one lane marking moving towards middle horizontally in front view. Based on this fact, a lane departure can be determined by checking the horizontal position of each lane marking, which is corresponding to the X-coordinates of top and bottom points of lane markings in image plane. The method of detecting the horizontal position of a lane marking can be described as Figure 3.17

Assuming we have lane markings with top point $Pt_0(x_{Pt_0}, y_{Pt_0})$ and bottom point $Pt_1(x_{Pt_1}, y_{Pt_1})$. In the pre-processing stage, we already get the ROI (as shown in Figure

Algorithm 4: TRM and Lane Marking Recognition

Pt_0: the top point(pt_0) of one line;
Pt_1: the bottom point(pt_1) of one line
AVG(RGB): get the average value of RGB for trapezoid;
$P(RGB)$: get pixel values of RGB for one point
x_{mid}: $x_{mid} = (x_{min} + x_{max})/2$ for every y step
LMP: Lane Marking Pixel;
$nLMP$: non-Lane Marking Pixel
W: numbers of WHITE points;
Y: numbers of YELLOW points
Event: select lane marking pixel and recognize lines
for $y = y_{Pt_0}$ *to* y_{Pt_1} **do**
    t = 20 + 21 $\times \frac{y - y_{Pt_0}}{y_{Pt_1} - y_{Pt_0}}$
    for x = *(*$(y - y_{Pt_0})\frac{x_{Pt_0} - x_{Pt_1}}{y_{Pt_0} - y_{Pt_1}} + x_{Pt_0} - t$*) to (*$(y - y_{Pt_0})\frac{x_{Pt_0} - x_{Pt_1}}{y_{Pt_0} - y_{Pt_1}} + x_{Pt_0} + t$*)* **do**
        Get the average color $P(RGB)$ of a 3×3 block centred at point (x, y)
        if *(*$P(R) > AVG(R)$*) and (*$P(G) > AVG(G)$*)* **then**
            Mark this y step as LMP
            $LMP = LMP + 1$
            if $P(B) > 0.8 \times \frac{P(R)+P(G)}{2}$ **then**
                Mark this point WHITE
                $W = W + 1$
            else
                Mark this point YELLOW
                $Y = Y + 1$
        else
            Mark this y step as nLMP and $nLMP = nLMP + 1$
if $LMP \leq 50$ **then**
    Mark this line Non-Lane-Marking
    return
if $10 \leq nLMP \leq 100$ **then**
    Mark this line DASH
else
    if $nLMP \leq 10$ **then**
        Mark this line SOLID
    if $nLMP \geq 100$ **then**
        Mark this line Non-Lane-Marking
if $Y > W$ **then**
    Mark this line YELLOW
else
    Mark this line WHITE
Use (x_{mid}, y_{Pt_0}) at first step of y which is marked as LMP to update P0(pt_0)
Use (x_{mid}, y_{Pt_1}) at last step of y which is marked as LMP to update P1(pt_1)

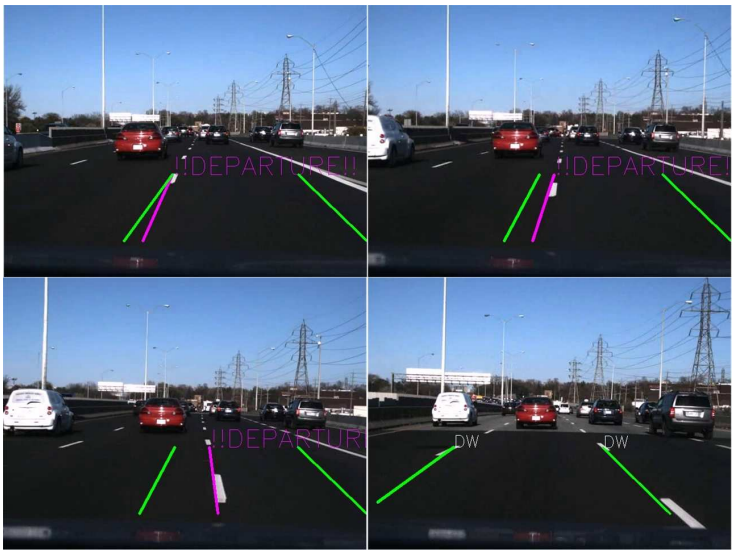


Figure 3.16: A process of Lane Departure Detection and Warning(in time sequence)

3.17) with its width(W), height(H) and the top-left point (m, n). Experimental results on the case of downtown and rural area of Ottawa have revealed that, if

$$(\frac{1}{5} * W + m) < P0_x < (\frac{4}{5} * W + m) \tag{3.26}$$

&&

$$(\frac{1}{5} * W + m) < P1_x < (\frac{4}{5} * W + m) \tag{3.27}$$

This lane marking with Pt_0 and Pt_1 can be taken as "moving towards middle". If lane markings only satisfy one of the restraints above, as two blues lines in Figure 3.17, that is the normal situation without lane departure.

As talked above, there is supposed be only one "moving towards middle" lane marking in a lane departure situation, which is either the left lane marking or the right lane marking. If the left and right lane markings are both "moving towards middle", that is mostly because the lane is getting narrow, or the ROI is not set to an adaptable size for real scenario. Hence two "moving towards middle" lane markings should be taken as a non-departure situation. Experimentally, as shown in Figure 3.16, the car is departing from one lane to another, starting from top left picture, then top right, then bottom left and at last driving

on a new lane in bottom right picture. Green lines indicate normal lane markings while purple lines indicate "moving to middle" lane markings. When a lane departure happens, a warning message is responsively written on image.

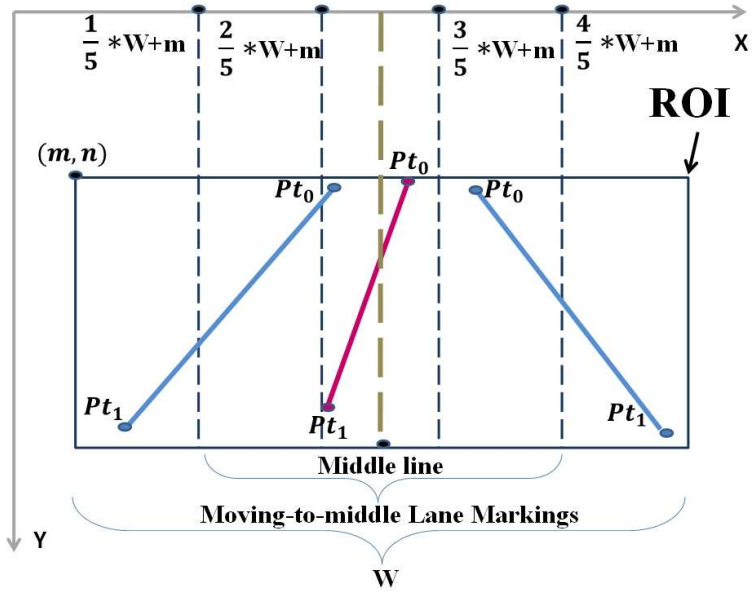


Figure 3.17: The proposed LDW method: Two blue lines within the ROI are normal lane markings which only have Pt_0 located between $\frac{1}{5} * W$ and $\frac{4}{5} * W$ for each line. The purple line in the middle is a "moving-to-middle" lane marking with both Pt_0 and Pt_1 between $\frac{1}{5} * W$ and $\frac{4}{5} * W$, which leads to a lane departure.

Chapter 4

Experimental Results

This chapter is organized in 6 sections, which covers the experimental platform, experiment overview and all the relevant performance evaluation. Emphasis will be laid on Section 4.3, 4.4, 4.5 and 4.6 for presenting the performance evaluation on the proposed simplified module, comprehensive module, recognition method and different edge segmentation methods.

4.1 Experimental Platform

Both comprehensive and simplified modules are implemented with OpenCV library and C++, under the environment of Windows, using Intel core $i3$ CPU and 4G RAM. The testbed used for experiment is shown as in Figure 4.1

4.1.1 Point Gray Flea®3 Camera System

The video sensing system that we use to collect video data is The Point Gray Flea®3 camera FL3-U3-13S2C-CS, which is equipped with $5.8mm - 58mm$ zoom lenses. We used this camera system because it offers a variety of CMOS image sensors in ultra-compact package. The camera allows some image processing features including gamma correction and color interpolation. Also, it provides other appealing functions such as on-camera

Figure 4.1: The Testbed used to test our system

power, temperature monitoring, in-field adaptable firmware, and a 32 megabytes frame buffer. Camera features are detailed as in Tables 4.1 and 4.2:

Table 4.1: Camera Parameters

Camera attribute	**Camera parameters**
Company	Point Grey
Model No.	FL3-U3-13S2C-CS
Image Sensor	Sony IMX035 CMOS
Image Sensor Size	Diagonal 1/3", $4.8mm \times 3.6mm (H \times V)$
Unit Size	3.63 μm
Maximum Resolution	1328×1048 at 120fps
Interface	USB 3.0
Image Buffer	32 MB frame buffer
Flash Memory	1MB

Most importantly, the camera system provides a software library (FlyCapture SDK) compatible with both Windows and Linux. Thanks to an interface to control all Point Grey digital vision devices, a synchronization between the cameras and the computer can be assured for the sake of real-time performance. Furthermore, we used a CS mount type which fits the Flea3 requirements. The two varifocal cameras with $5mm - 50mm$ focal length range provides the ability to adjust the FOV / zoom level of the camera.

Table 4.2: Lens Parameters

Lens Attribute	**Lens Parameters**
Company	Edmund Optics
Model No.	#55-256
Focal Length (Mm)	5.0 – 50.0
Maximum Camera Sensor Format	1/3”
Aperture (F/#)	F1.3 – 16C
FOV, For 1/3” Sensor	51.8 – 5.6
Working Distance (Mm)	800
Mount	Cs-Mount

By using the method proposed in [128], the cameras and lenses are calibrated correctly to work well with our system. The on-board camera is fixed at a height $1.3 meters$ above the ground and in the front of our experimental car. The frame rate of our videos is $25 fps$. Since the frame size has a grate impact on the total speed of the system, the image is resized to 640×480 after being captured from cameras, for real time speed.

4.1.2 Real-life Video Collection

For generality, we consider both day and night situations. Four videos were taken from highway and urban roads in Ottawa during daytime, and two videos from urban area of Ottawa at night. These videos represent common scenarios with different weathers, road surfaces, lightening condition, and traffic density, which people might encounter in real life. (Tables 4.3 and 4.4).

4.2 Experiment Overview

With different segmentation approaches, the difference of comprehensive and simplified module mainly exists in lane detection and tracking (LDT) performance. Therefore the

Table 4.3: Daytime Video Clips for testing

	Clip #1	Clip #2	Clip #3	Clip #4
Weather	Cloudy	Sunny	Sunny	Cloudy
Location	Highway	Urban	Highway	Urban
Traffic condition	Heavy	Light	Medium	From Light to Heavy
Road surface	Very Rough	Flat and Smooth	Flat and Smooth	From Smooth to Rough
Frame NO.	912	864	1080	960
Frame speed	24fps	24fps	24fps	24fps
Lane markings/frame	2	1	2	2
Line type	Dash White	Solid Yellow	Dash White and Solid White	Dash White and Solid Yellow
Number of frames containing departure	48	28	68	72

Table 4.4: Night-scene Video Clips for testing

	Clip #5	Clip #6
Location	Urban	Urban
Traffic condition	Heavy	Medium
Road surface	Normal Smooth	Normal Rough
Frame NO.	1200	1043
Frame speed	24fps	24fps
Lane markings/frame	2	2 or 1
Line type	Dash White	Dash White and Solid Yellow
Number of frames containing departure	63	57

evaluation on LDT performance has to be divided into two parts, for simplified and comprehensive module respectively.

Although the proposed lane marking recognition (LMR) and departure warning (LDW) scheme is primarily based on the lane detection results, it is still reasonable to say, correct detection does not guarantee correct recognition, especially for the case of a simplified LDT module without recognition (Section 4.3, as the system proposed in [78]). Thus it is better to separately evaluate the LDT, LMR and LDW performance. When calculating the DR (correct detection rate for lane detection), colours and shapes of lane markings as well as lane departure are not considered. The only thing which contributes to DR is the fitness of detected lane marking to ground truth.

By exploring the literature, the tabularly comparative evaluation is used as the conventional evaluation method for LDT performance. Ground truth is either hand-annotated on each frame or determined by visual inspection, as performed in [13], [106], [98] and [34]. Researchers usually qualitatively judge if the result fits well with ground truth for each frame or not.

In this case, tabularly comparative evaluation is employed for LDT performance of both modules, as well as the comparison of different edge detection methods (Section 4.5). Also as an essential part of the comprehensive module, the LMR and LDW performances

are also evaluated in this way (Section 4.6). Also, to comprehensively assess the goodness of LDT performance, two novel evaluation methods have been proposed aiming at the stability, accuracy and probability distribution of detection results. As a part of the LDT performance evaluation for comprehensive module, these two methods will be introduced in Section 4.4.2. The experiment stage will be organized as following:

- LDT Performance of Simplified Module (Section 4.3).
- LDT Performance of Comprehensive Module (Section 4.4).
- Edge Detection Performance (Section 4.5).
- LMR and LDW Performance (Section 4.6).

4.3 LDT Performance of Simplified Module

In this part, tabularly comparative experiments are performed to evaluate the efficiency and accuracy of simplified module. Figure 4.3 shows some frames with correct detection of lane markings. These pictures show that our proposed simplified module based on MSER segmentation and refinement performs well in different real scenarios. Apparently we can see from Figure 4.2, that the detection results of simplified module (Figure 4.2(i)) keeps desired details (lane marking areas) and removes some noises (cars, trees, curbs), while compared to results based on edge segmentation (Figure 4.2(c)) and MSER segmentation without refinement (Figure 4.2(f)).

The evaluation of each method is performed using some performance metrics. As introduced in Section 4.2, DR is one of the metrics that used for representing detection rate. Additionally, two metrics are used: False Positive rate (FPR) and False Negative rate (FNR). False Positive (FP) and False Negative (FN) considers the presence or absence of lane markings. FPR refers to the probability of falsely selecting a given object or contours such as vehicles, curbs of road, trees or electric poles, as a lane marking. On the other hand, FNR refers to the situation when a lane marking is falsely rejected by the detection method.

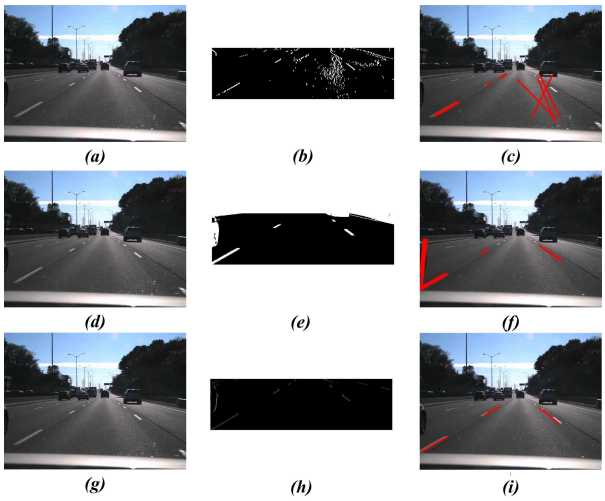


Figure 4.2: Comparison of Edge-based, MSER-based and the proposed simplified module. Images (a), (d) and (g) represent the same frame taken from Clip #1. (b), (e) and (h) are the results of edge segmentation, MSER segmentation and simplified module, respectively. Images (c), (f) and (i) are the results obtained after applying PPHT on (b), (e) and (h), respectively.

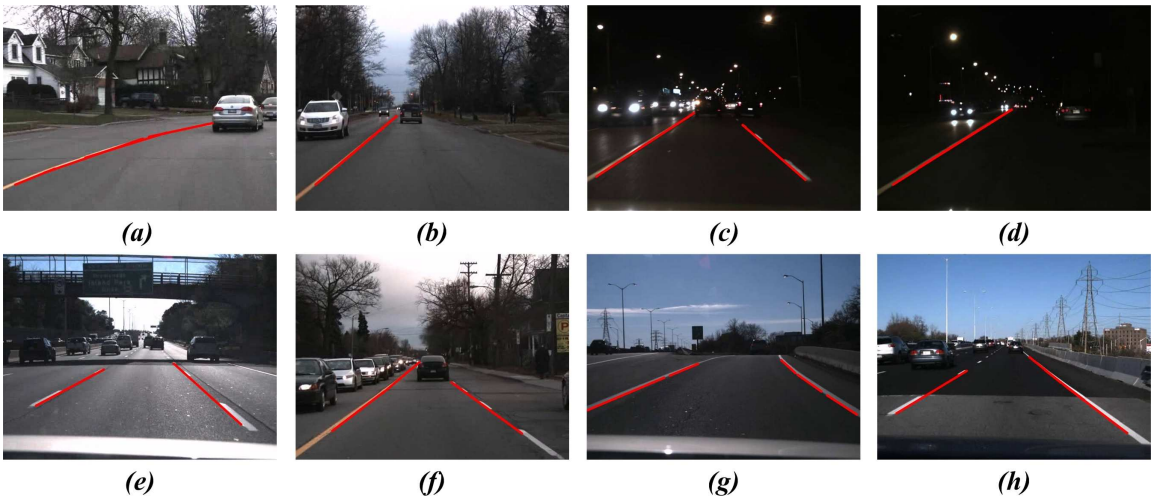


Figure 4.3: Correct detection in simplified module: (a) urban area: curvy lane marking occluded with vehicles; (b) urban area: single lane marking; (c) urban area: heavy traffic with strong lightening; (d) urban area: medium traffic; (e) urban area: medium traffic, strong sunlight, rough and shadowy road; (f) urban area: rough road with heavy traffic, cloudy weather; (g) highway: slope and curvy lane; (h) highway: different background brightness on road, heavy traffic.

4.3.1 Comparative Performance Evaluation (Simplified Module)

Three methods are implemented and included in Tables 4.6 and 4.5 for comparative evaluation, method (a), method (b) and the proposed simplified module referred as (c). The method (a) uses edge-segmentation based on Sobel algorithm (for the reason explained in Section 4.5), followed by PPHT to detect lane markings. Whereas the method (b) uses MSER segmentation without refinement, followed by PPHT. The reason behind this choice is to show the accuracy and efficiency of our simplified module (c) (which refines method (b)) against edge-based segmentation and MSER-based segmentation without refinement. Admitting the drawbacks of the proposed simplified module (method (c)), some images with detection failure have been presented in Figure 4.4, also serve as a demonstration about the FP and FN results.

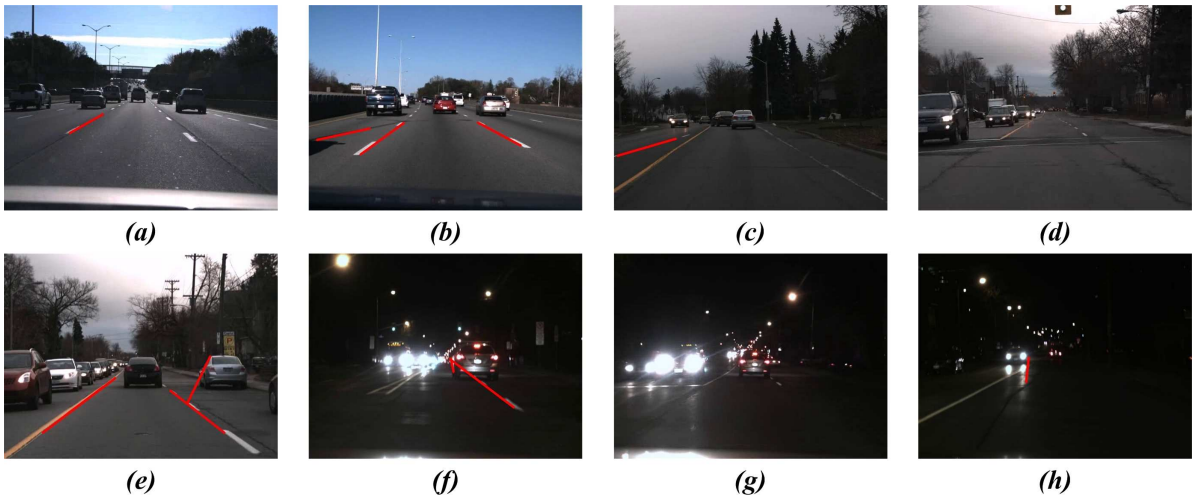


Figure 4.4: Examples of False Positive and False Negative results of simplified module: (b) and (e) show the false positive results as well as correct detection; (a) and (f) show the false negative results for one lane marking while the other one being correctly detected; (c), (d), (g) and (h) show the false negative results for both left and right lane markings as well as some false positive results.

Different experimental results are presented in Tables 4.6 and 4.5 for day and night time, respectively. We can see from Table 4.6 that MSER-based segmentation (b) performs better than edge-based segmentation (a) for the most of time, especially for Clip #4, which represents a common situation in urban area. In Clip #4, method (b) increases the detection rate from 77% of method (a) to 95.8%, which dramatically decreases the FNR

Table 4.5: Comparative performance evaluation for simplified module at night

	Clip #5			*Clip #6*		
	DR	**FPR**	**FNR**	**DR**	**FPR**	**FNR**
(a)	57.8%	1%	42.2%	64.1%	0	35.9%
(b)	74%	14.8%	26%	80.6%	9.7%	19.4%
(c)	77.6%	7.1%	22.4	83.3%	4.5%	16.7%

from 23% to 4.2%. Unfortunately, method (b) possesses some instability when it comes to other clips (having a FPR of 30.4% and FNR of 5.1%). In order to eliminate the instability of MSER segmentation, method (c) can be used to effectively refine the results of method (b). Method (c) increases the detection rate by nearly 10% for Clip #2 and Clip #3, compared to method (b). Further more, method (c) decreases FPR and FNR for all four video clips.

It is shown in Table 4.5 that, in spite of highest FNR and lowest DR, edge-based segmentation (method (a)) has much lower FPR than other two methods (only 1% and 0%). This is mainly because weak lightening at night makes the noisy edges less visible. Besides FPR, MSER segmentation performs significantly better than edge segmentation during night time, and method (c) refines the results of MSER segmentation. Also it is noticeable that, as a result of poor lightening in night time and unpredictable lightening interference of cars, lane detection system may perform worse than it does in daytime, which is reflected experimentally in Table 4.5 and Figures 4.4(f),(g) and (h). We find out that, during night time, the successfulness of LDT scheme in simplified module mainly depends on the traffic density and sometimes on lightening system on the road.

4.4 LDT Performance of Comprehensive Module

In this part, it is necessary to compare between different forms of Hough transform (HT), which is known as the most common detection algorithm of all the existing LDT systems. Standard Hough transform (SHT) is mostly used by researchers and has presented

Table 4.6: Comparative performance evaluation for simplified module in daytime

	Clip #1			*Clip #2*			*Clip #3*			*Clip #4*		
	DR	**FPR**	**FNR**	**DR**	**FPR**	**FNR**	**DR**	**FPR**	**FNR**	**DR**	**FPR**	**FNR**
(a)	72.7%	18.7%	27.3%	84.3%	0	15.7%	80.1%	7.7%	19.9%	77%	10.7%	23%
(b)	69.6%	14.2%	30.4%	89.2%	5.1%	10.8%	84.3%	5.5%	15.7%	95.8%	9.8%	4.2%
(c)	75.3%	7.4%	24.7%	97.9%	4.2%	2.1%	94.9%	0	5.1%	100%	3.6%	0

very good performance experimentally, as shown in [34] and [13]. Different from SHT, Probabilistic Hough Transform (PHT) rarely draws the attention (only used by Hota in [49]). As introduced previously, this comprehensive module uses an improvement of PHT which is called progressive probabilistic Hough transform (PPHT) as the core detection algorithm, followed by a tracking stage with Kalman filter (KF) and a novel refinement scheme composed of ATSL and TRM. Hence in this part, for the performance evaluation and comparison on LDT performance, three methods which need to be compared are organized as following:

Method (a): SHT + KF (as in [34] and [13])

Method (b): PHT + KF (different from [49], which uses PHT as well)

Method (c): PPHT + ATSL + TRM + KF (proposed comprehensive module)

Similar to the simplified module, two other methods are considered besides of the comprehensive module. The first method (referred as method (a)) uses Hough transform as the core method of lane marking detection. For instance, in [34], Cualain et al. present a method in the Hough transform domain for lane marking modelling based on subtractive clustering and Kalman filtering. Whereas the second method (referred as method (b)) uses the PHT algorithm as the core detection method. Taking the method proposed in [49] as an example, with the help of PHT, Hota et al. developed an agglomerative clustering algorithm based on the Y-intersects of each detected lines.

Both of methods (a) and (c) use Kalman filter for the tracking stage. Therefore different from [49] which uses a least square regression for the refinement of PHT, we implemented Kalman filter in method (b) for the sake of consistency. The reason behind this choice, is to show the performance of the comprehensive module against SHT and PHT. Our detection and tracking scheme is referred as method (c).

As mentioned in Section 4.2, the experiments for the LDT performance of comprehensive module are conducted in three different ways: the tabularly comparative evaluation and two proposed evaluation scheme, namely as point clustering comparison and pdf analysis, which will be introduced in Section 4.4.2 and 4.4.2, respectively.

In Figure 4.5, we show some images of correct lane detection using the comprehensive

Figure 4.5: Images of correct LDT results of the comprehensive module

module. In addition, we also consider the false positive and false negative results of the comprehensive module, which are defined the same way as in Section 4.3. As in Figure 4.6, examples of the LDT failures of comprehensive module are shown, as well as the LMR failure (which will be talked about in Section 4.6).

4.4.1 Comparative Performance Evaluation (Comprehensive Module)

For this part, same matrices are used as for the simplified module (Section 4.3). Also we have different experimental results (DR, FPR and FNR of each method) presented in Table 4.7 and Table 4.8, for day and night respectively. As seen from Table 4.7, the detection rate of method (a) is mostly lower than those of method (b) and our proposed method (c). The only exception is for Clip #2, which represents scenarios with unusually ideal conditions, i.e., smooth and flat road with ideal lighting conditions. Clip #2 only brings a slight advantage in the favor of method (a) (only 5.3% greater than method (b)), while in the other three scenarios, PHT (method (b) seems to have better performance than SHT (method (a)). This can be related to the evidence on PHT works better than SHT for Lane Detection.

Table 4.7: Comparative performance evaluation for comprehensive module in daytime

	Clip #1			*Clip #2*			*Clip #3*			*Clip #4*		
	DR	**FPR**	**FNR**	**DR**	**FPR**	**FNR**	**DR**	**FPR**	**FNR**	**DR**	**FPR**	**FNR**
(a)	93.9%	7.03%	6.1%	95.7%	0.1%	4.3%	97.0%	3.8%	3.0%	86.4%	7.2%	13.6%
(b)	85.4%	14.5%	14.6%	90.4%	0.4%	9.6%	99.7%	2.7%	0.3%	92.4%	8.4%	7.6%
(c)	89.3%	5.1%	10.7%	99.8%	0	0.2%	99.8%	2.1%	0.2%	97.9%	3.4%	2.1%

Figure 4.6: The left column shows false positive and false negative detection. The middle column shows false negative recognition for solid lines and false positive recognition for dashed lines (erroneously recognize solid as dashed line). The right column shows false negative detection and one erroneous recognition (recognize dashed line as solid).

For the other three scenarios, our proposed module seems to perform better than method (a) and method (b). Moreover, it can be seen from Table 4.7 that the use of TRM-ATSL (method (c)) significantly improves the detection rate compared to Hough transorm-KF without ATSL and TRM. The proposed comprehensive module with TRM-ATSL effectively improves the detection performance and decreases the false positive rate for all scenarios.

When it comes to night scenes, the proposed comprehensive module (c) reaches a detection rate of 98% for Clip #6, 7.6% higher than method (c) for Clip #5 (Table 4.8). This is due to the severe lightening interference at night brought by heavy traffic. Besides, Clip #6 contains solid lines and dashed lines, while Clip #5 contains only dashed lines. As a result of poor lightening, dashed lines are harder to be detected than solid lines. At night, lane detection rate is mainly affected by traffic density and the type of markings, and is also slightly affected by the roughness of the road.

Table 4.8: Comparative performance evaluation for comprehensive module at night

	Clip #5			*Clip #6*		
	DR	**FPR**	**FNR**	**DR**	**FPR**	**FNR**
(a)	84.7%	17.8%	15.3%	90%	7%	10%
(b)	86%	10%	14%	96.8%	6.2%	3.2%
(c)	90.4%	8%	9.6%	98%	2%	2%

4.4.2 New Methods of LDT Performance Evaluation

Apart from the conventional comparative evaluation, this thesis also proposed new experimental methods aiming to evaluate the LDT performance of the comprehensive module, which consists of two parts: 1) Point Clustering Comparison (Section 4.4.2) and 2) PDF analysis of Lane Detection (Section 4.4.2).

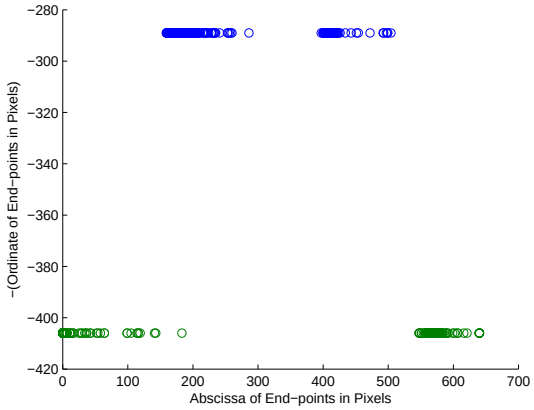
Point Clustering Comparison

This comparison needs be done based on the spatial distribution of end-points of both left and right markings. Ideally, we assume the LDT results and the testing environment are absolutely ideal (the camera is set to be stable and the road is perfectly smooth and flat). Based on this assumption, the detected left and right lane markings should almost keep still constantly without lane departure (which means driving without switching lanes or lateral offset). This yields, if we collect the ending points of left and right lane markings and draw the sparsity of these four points with MATLAB, there should be four point clusters which are extremely converged.

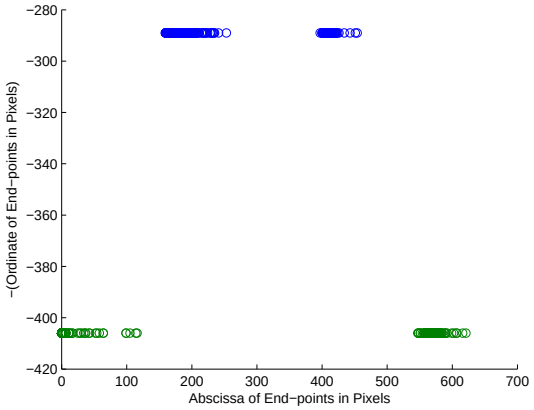
Furtherly Considering that the output of TRM and tracking stage is the starting and ending points of lane markings (Section 3.2.3 and 3.3.2), those 4 point-clusters drawn by MATLAB (starting and ending points of every detected lane markings) are expected to converge around 4 points. In this case, the three LDT approach introduced in Section 4.4, methods (a), (b) and (c) need to be performed on the same ideal clip. We collect the coordinates of all starting and ending points of detected lane markings. By taking the

abscissa as X coordinates and ordinates as Y coordinates of those points, those points are drawn with MATLAB.

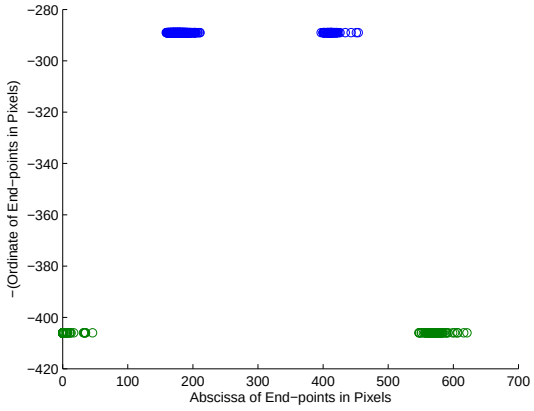
By comparing the sparsity of point clusters obtained with respect to the same video clip, we can tell which method has better convergence, as to say better stability. Because Clip #3 and Clip #4 represent common situations in highway and urban area respectively, for generality, we use fractions of both clips which do not contain the cases of lane departures and steep roads, and draw the point-cluster distribution of methods (a), (b) and (c).(as shown in Figure 4.7 and Figure 4.8)



(a) Method (a)

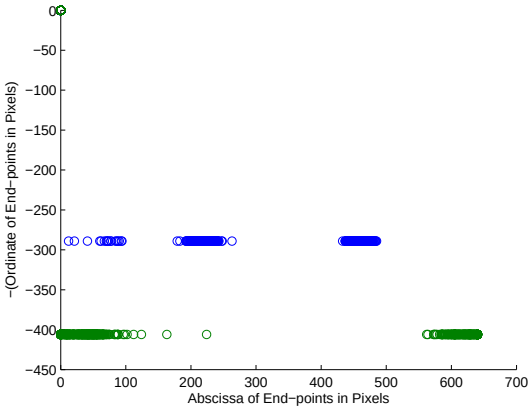


(b) Method (b)

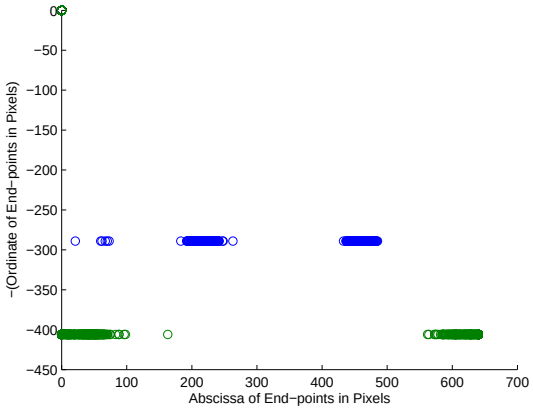


(c) Method (c)

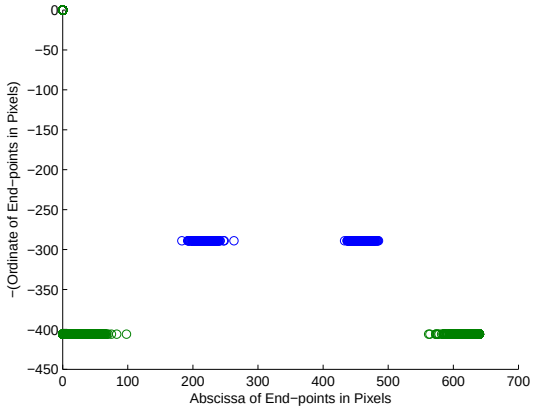
Figure 4.7: Cluster Comparison for Clip #3: We take abscissa as X coordinates and ordinate as Y coordinates of each detected point



(a) Method (a)



(b) Method (b)



(c) Method (c)

Figure 4.8: Cluster Comparison for Clip #4: We take abscissa as X coordinates and ordinate as Y coordinates of each detected point

As we can see from the clusters of Figures 4.7 and 4.8, the four detected end-points do not vary too much on ordinate. This is because we truncate every detected lane marking in the direction of Y-axis, with the height of region of interest in order to avoid having intersection of two lane markings. Also since the direction of y-axis in MATLAB is opposite to the y-axis in C++ (Opencv), we use (-ordinates) of detected points to indicate the accordant point clusters with C++ in the frames.

It is shown from Figures 4.7 and 4.8, that point clusters of method (b) converge better than method (a); method (c) obviously converge better than those of methods (b) and (a),

respectively. This proves that method (c) has the best stability, followed by (b) and (a), respectively, which indicates the advantage of proposed comprehensive module against the other two.

PDF Analysis of Lane Detection

Recalling the use of Kalman filter in Section 3.3.2, LDT system as proposed in the comprehensive module can be considered as a Gaussian stochastic process. Based on this fact, we can analyse the LDT performance from the perspective of stochastic model (as mentioned previously in Section 3.3.2), which involves the analysis of the Probability Density Function (PDF) of detection and tracking. The LDT system with the best similarity to Gaussian distribution can be considered as the best of all method candidates.

Referring to the spatial distribution of points in Figures 4.7 and Figure 4.8, position of starting and ending points do not vary too much on ordinate (in the direction of y-axis). Hence, the analysis of probability distribution can be only focused on the x-coordinates with respect to both ends of lane markings, which are detected by method (a), method (b) and method (c) (same methods defined as Section 4.4).

Likewise, similar to Point Clustering Comparison (Section 4.4.2), we use video clips without lane-departure, curve lanes, lane switching or ups and downs on road. Figure 4.9 shows the different experimental results on Clip #4, while Figure 4.10 shows the results on Clip #6. Figure 4.9 and Figure 4.10 represent two lane markings (with both ends of Pt_0 and Pt_1 as introduced in Section 3.3.2) on urban roads in daytime and one lane marking (with also Pt_0 and Pt_1 in Section 3.3.2) on urban roads at night, respectively. Ideally without any outliers, LDT systems should present smooth curves of strictly Gaussian distribution for each of four points (starting and ending points of right and left lane markings), which makes every curve only has four peaks (different positions of four points). However, as shown in Figure 4.9 and Figure 4.10, outliers sometimes result in more than four peaks in real scenarios, which can be described as Mix-Gaussian distribution.

Seeing from Figure 4.9 and Figure 4.10, high peaks always refer to the stablest positions of detected points, while some short peaks exist as unwanted outliers which inevitably

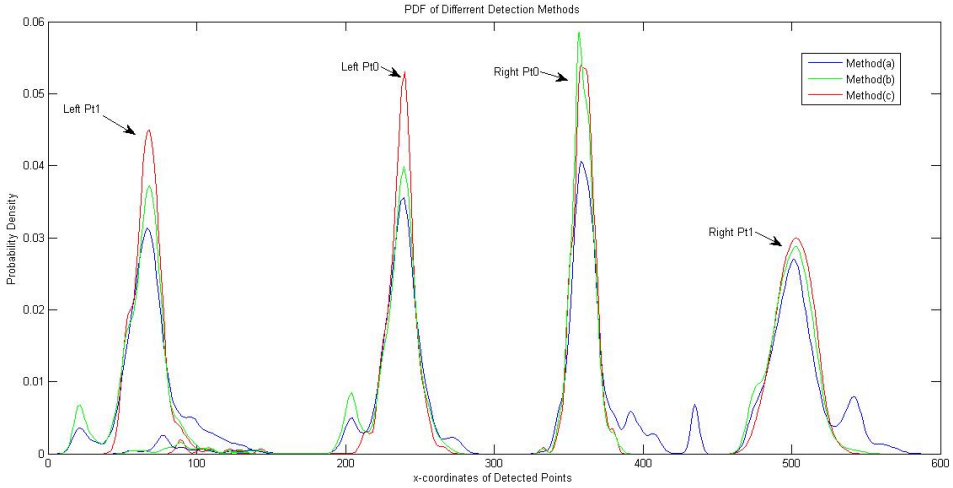


Figure 4.9: PDF of two-lane-marking detection results in daytime (urban roads): 1.) Method (c) has the best similarity to Gaussian distribution and highest peaks for all four positions (left Pt_1, Pt_0 and right Pt_1, Pt_0), hence it has the best accuracy and stability; 2.) Method (c) has the most smooth curves and removes severe outliers (short peaks) of Method (a) and Method (b); 3.) Method (b) relatively performs better than Method (a)

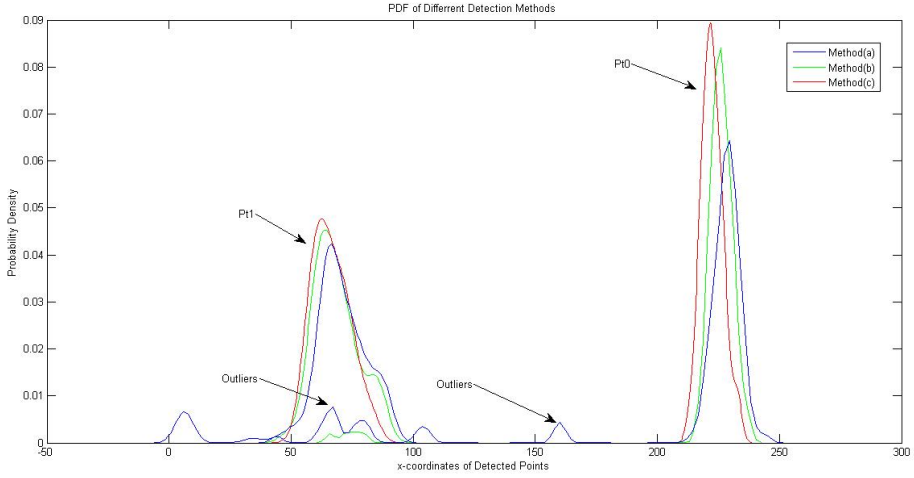


Figure 4.10: PDF of one-lane-marking detection results at night (urban roads): 1.) Method (c) has the best similarity to Gaussian distribution and highest peaks for both two positions (Pt_1 and Pt_0 of lane marking), hence it has the best accuracy and stability; 2.) Method (c) has the most smooth curves and removes severe outliers (short peaks) of Method (a) and Method (b); 3.) Method (b) relatively performs better than Method (a))

increases false positive detection. For the PDF curves, higher and narrower peaks means better stability while fewer outliers means better accuracy. Accordingly, some conclusions can be drawn from the pdf analysis of three different LDT systems (methods (a), (b) and (c)). As shown in Figure 4.9 and Figure 4.10, the pdf of method (c) is the most similar to Gaussian distribution. Hence method (c) has the best stability and accuracy of all, and method (b) with PHT has better performance than method (a) with SHT. This is in accordance with the results of point clustering comparison (Section 4.4.2).

Especially seeing from Figure 4.10, blue line (method (a)) has severe outliers (short peaks), while green line (method (b)) eliminates these outliers to some extent, but still has some bumps and not similar to Gaussian distribution. As a result of the proposed comprehensive module, the red line (method (c)) has the best convergence and remove outliers, appearing as extremely close to Gaussian distribution. This proves the advantage of the detection and tracking scheme in this thesis, as well as the rationality and necessity of using Kalman filter as the tracking algorithm for LDT system.

Nevertheless, in some complex scenarios (or non-linear dynamic systems), the detection and tracking results of lane markings may not comply with Gaussian distribution, sometimes featuring as a namely Mix-Gaussian distribution which has more than one peaks. For instance, Extended Kalman filter (EKF) is required for tracking stage of the LDT system in [112], which presents a non-Gaussian distribution. For that case, the PDF analysis need to be adjusted in favour of other probability distributions.

4.5 Edge Detection Performance

Edge detection is an essential part of the proposed comprehensive module, also known as one of the most popular segmentation method for computer vision systems. In attempt to choose the most suitable edge detection algorithm for our case, we compared the Sobel algorithm with other edge detectors such as Canny, Prewitt and Robert. The results of comparative evaluation and PDF analysis are shown as in Table 4.9 and Figure 4.5, respectively. Different edge segmentation algorithms are applied on 300 frames (part of

a video clip taken from Ottawa). After segmentation, we only use PPHT to detect lane markings, without tracking and any other refinement steps. There are 600 lane markings to be detected for these 300 frames.

Table 4.9: Comparison of different edge segmentation methods

	Processing time / frame	***Detected lane markings***	***TPR***	***FPR***
Robert	26 ms	582	66%	31%
Canny	94.5 ms	788	87%	44%
Prewitt	48.3 ms	812	90%	45%
Sobel	32.3 ms	609	92%	9.5%

Table 4.9 proves that Canny is not suitable for lane detection, with a False Positive Rate (FPR) of 44% and consumes much more time than other methods ($94.5ms$). Conversely, Sobel performs the best comprehensively of all four methods. Also by comparing the probability density function (PDF) of four algorithms, it is obvious to see from Figure 4.5 that, Canny does not have strong confidence for a center value (a converged and relatively higher peak than other values in the pdf curve); contrarily, Sobel presents the best stability (thinner and higher peaks) and accuracy (fewest short peaks, which are due to outliers) of all four methods. The results make it reasonable to use Sobel as the edge detector for comprehensive module, as well as the representation of edge segmentation in comparison with MSER segmentation for the experiment (as explained in Section 4.3).

4.6 LMR and LDW Performance

Similar to the comparative performance evaluation performed on LDT, we also use hand-label method to qualitatively judge the correct recognition of nature of lane markings (colour and shape) and the presence of lane departure, as well as false positive (FP) and false negative (FN) results. RR, FPR and FNR stand for Recognition Rate, False Positive Rate and False Negative Rate respectively. The definition of false positive (FP) and false negative (FN) results in LMR system has some extensions while still considering FP and

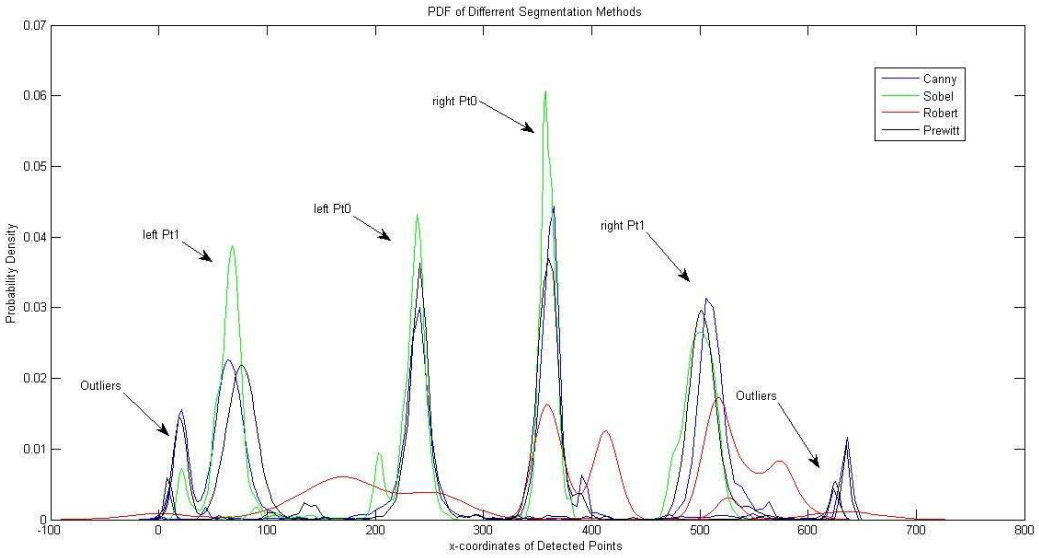


Figure 4.11: PDF Analysis of Different Segmentation Methods. The sobel algorithm (green line) converges best of all, so that has the best detection accuracy; and because of having fewest outliers, Sobel (green line) has the lowest false positive rate.

FN of LDT system: **The first type of FP and FN is defined same way as LDT, which are explained in Section 4.3.**

- FP refers to lines falsely selected as lane markings, which may result from a given object or contours such as vehicles, curbs of road, trees or electric poles;
- FN refers to the situation when a lane marking is falsely rejected by the system.

The second type of FP and FN is defined as follows:

- A solid line being incorrectly recognized as a dashed line gives one FP score to dash recognition and one FN score to solid recognition;
- For the situation where a dashed line being recognized as solid line, it gives one FP score to solid line and one FN score to dashed line.
- Similar to solid and dashed lines, white and yellow lines being erroneously recognized as each other also give FP and FN scores the same way above

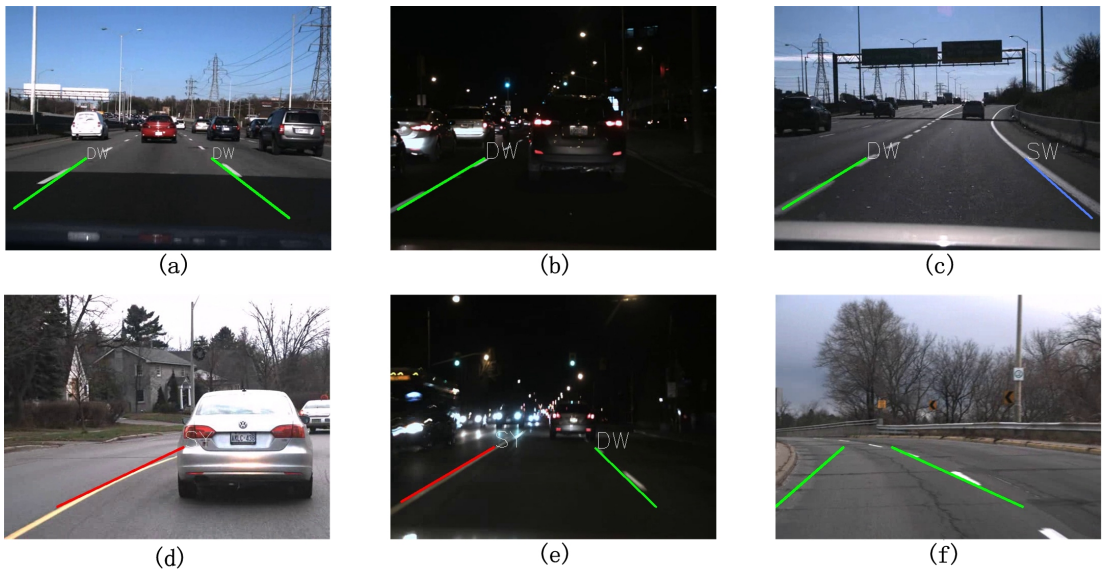


Figure 4.12: Correct LMR results in different scenarios: (a) strong sunlight, medium traffic and lane markings occluded with shadows in front (highway); (b) heavy traffic and strong lightening at night (urban area); (c) strong sunlight and smooth road surface (highway); (d) single lane marking occluded with vehicle in cloudy weather (urban area); (e) medium traffic and strong lightening at night (urban area); (f) slope and curvy road with rough surface (highway)

Figure 4.12 has shown the sample frames with correct recognition of lane markings. Figure 4.12(a) shows the scene where lane markings occluded with shadows, strong sunlight and normal traffic on highway; Figure 4.12(b)and Figure 4.12(e) shows the night-scene with normal and very heavy traffic in urban area, with strong lightening interference; Figure 4.12(c)shows the results of strong sunlight and smooth road surface on highway; Figure 4.12(d) and Figure 4.12(f) both show the cloudy scene, where we do not have enough lightening to distinguish the lane markings with road surface. Figure 4.12(d) has single lane markings with a small curvature occluded with vehicle in urban area and Figure 4.12(f) shows the result we having very curvy lane markings and very rough road surface on highway. These pictures show that our proposed LMR method performs well in real scenario. Table 4.10 and Table 4.11 show the daytime and nighttime results of LMR and LDW, in terms of the recognition rate (RR), false positive rate (FPR) and false negative rate (FNR). Same video clips as described in Section 4.1.2 are used for testing.

Table 4.10: Comparative evaluation on LMR and LDW performance (Daytime)

	Clip #1			*Clip #2*			*Clip #3*			*Clip #4*		
	RR	**FPR**	**FNR**	**RR**	**FPR**	**FNR**	**RR**	**FPR**	**FNR**	**RR**	**FPR**	**FNR**
Solid	N/A	N/A	N/A	99.8%	0	0.2%	97.0%	4.2%	3.0%	95.4%	8.4%	13.6%
Dashed	84.3%	7.14%	15.7%	N/A	N/A	N/A	94.7%	0.24%	5.3%	89.4%	3.4%	10.6%
Yellow	N/A	N/A	N/A	99.8%	0	0.2%	N/A	N/A	N/A	95.6	0	4.4%
White	89.3%	0	11.7%	N/A	N/A	N/A	97.8%	0	2.2%	100%	4.5%	0
Lane Departure	97.2%	12.4%	2.8%	100%	0	0	100%	0	0	100%	0	0

Table 4.11: Comparative evaluation on LMR and LDW performance (Night)

	Clip #5			*Clip #6*		
	RR	**FPR**	**FNR**	**RR**	**FPR**	**FNR**
Solid	N/A	N/A	N/A	95.3%	3.2%	4.7%
Dashed	89.7%	8%	10.3%	97%	4.7%	3%
Yellow	N/A	N/A	N/A	76.8%	12.7%	23.2%
White	84.4%	8%	15.6%	90.3%	33.7%	9.7%
Lane Departure	100%	0	0	100%	0	0

Also seeing from Table 4.10, the proposed Lane Recognition and Departure Warning method presents good performance in daytime. As we can see, recognition performs best in sunny and highway scenario (e.g. 97% and 94.7% for solid and dashed lines respectively in Clip #3, 99.8% in Clip #2), it is because highways are usually better constructed and have relatively less traffic density than urban roads. Especially when it comes to lane marking shapes, with the proposed method, solid lines are easier to be detected, which leads to relatively high recognition rate (RR) as well as false positive rate (FPR); while dashed lines are harder to be recognized because of less lane-pixels available. Erosion of the road surface or strong shadows may cause solid lines to be erroneously recognized as dashed; on the other hand, there also might be problems when driving very fast on a road with rough stains between dashes, which actually happens very often in real life. For different colours of lane markings, it can be found that, in some situations with poor lightening, yellow lines are easier to be recognized as white lines because of the weak contrast. This leads to the recognition of white line overwhelms yellow line in Clip #4 (with the RR of 100% and 95.6% respectively for white and yellow), which is of cloudy weather and rough road surface. In this case, thanks to weak sunlight and front lights of vehicles, lane markings (no matter yellow or white lines) can be clearly reflected on roads. Both of white and yellow lines are clearly contrasted to road surface while not to each other.

Similar to the problems encountered in Clip #4, Clip #5 and Clip #6 have night scenes with darkness and interference from front lights of cars, which usually leads to inaccurate recognition. Lane departure works well with most scenarios except Clip #1, which is with

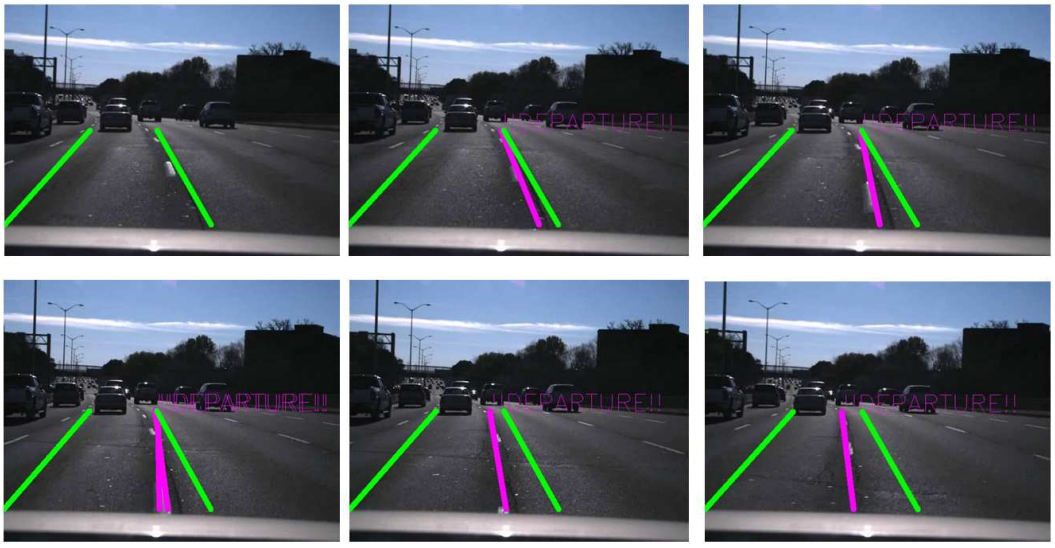


Figure 4.13: A sequence of images showing the process of lane departure detection and warning (urban area in Ottawa)

an uncommon situation. Lane departures can be accurately detected and warning messages are timely delivered as a real-time performance. Surprisingly, the proposed LDW system can also work in many challenging scenarios such as urban areas and strong sunlight, as shown in Figure 4.13.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we presented a real-time system which is composed of a comprehensive module and a simplified module. The comprehensive module consists of lane detection and tracking (LDT), lane departure warning (LDW) and lane marking recognition (LMR) systems; while the simplified module only focuses on LDT system. A lightweight version of Hough transform (PPHT) is applied for detection stage in the LDT systems of both comprehensive and simplified module, followed by a tracking stage implemented with Kalman filter. Experimental results have shown the evidence of both PPHT and Kalman filter improve the accuracy and robustness of the LDT systems.

Different from comprehensive module, instead of using segmentation methods based on edge information, the simplified module employs MSER algorithm for the preprocessing stage, followed by a proper refinement method. Thus the extracted ROIs after preprocessing stage can be well combined with PPHT and Kalman filter, which produces very good LDT results.

In the LDT system of comprehensive module, shape and texture information are used, which increases the detection rate. PPHT works as the core detection algorithm, followed by a novel refinement stage consisting of Angle Thresholding and Segment Linking (ATSL)

and Trapezoidal Refinement Method (TRM). TRM constructs one trapezoid for each left and right lane marking, thus improving the localization of real lane markings. TRM plays a significant role in the comprehensive module, as it feeds tracking, recognition and lane departure warning stages by corrected point information.

Based on TRM and on the concentration of pixels in each trapezoid, as an important contribution of the proposed system, the recognition of the color (yellow or white) and the shape (dashed or solid) of lane markings performs well in real scenario. Additionally, we have added a lane departure warning scheme to warn the driver when the vehicle is switching between lanes, by making use of both ends of each detected lane markings and splitting ROI.

Several experiments are conducted on highway and downtown area in Ottawa to test the real-time performance of our system. The detection rate of the simplified module averages 92.7% and exceeds 84.9% in poor conditions, while the detection rate of the comprehensive module averages 95.9% and exceeds 89.3% in poor conditions. The recognition rate of LMR system depends on the quality of lane paint and achieves an accuracy of 93.1%. However, lane markings are not easily recognized in severe conditions. Our future work includes mainly the improvement of the recognition of pavement marking in severe conditions, which will be introduced in Section 5.2.

5.2 Future Work

Some future works can be done in order for the improvement and extension of our work. For some challenging scenarios such as roads without lane markings, the proposed system still needs to be improved to allow for better performance. It can be done in terms of camera improvement. As mentioned in [42], interferometric wave radar can be used as a alternative data collector other than cameras. Additionally, thermal videos or other night vision cameras can be used to improve the detection rate at night.

As a part of intelligent transportation system (ITS), lane detection and tracking can be combined with traffic sign detection and road text detection. It can also be integrated

with wireless sensor networks ([37], [15], [19], [26], [1], [14], [86], [21], [23], [28], [18]) Also for the better prevention of potential lane departure crash, controllers and relevant control algorithms for a departure correction mechanism can be added to enable the LDT system have feedback control.

As an expansion of the LDW system, spacing requirements (as mentioned in [58]) for lane switching and Merging in Highway Systems can be computed based on the results of LDT results. Moreover, as proposed in [79], cascaded fuzzy inference system (CFIS) can be used to develop a car following and collision prevention scheme. Besides, tracking method of vehicle motion trajectories in [89] can also be employed in favour of the prevention of potential collision.

References

[1] *Algorithms and protocols for wireless sensor networks*, volume 62. John Wiley & Sons, 2008.

[2] A A Yla Jaaskiski and Nahum Kiryati. Automatic termination rules for probabilistic hough algorithms. In *PROCEEDINGS OF THE SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS*, volume 1, pages 121–121. PROCEEDINGS PUBLISHED BY VARIOUS PUBLISHERS, 1993.

[3] Osama Abumansoor and Azzedine Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *Vehicular Technology, IEEE Transactions on*, 61(1):275–285, 2012.

[4] Mohamed Aly. Real time detection of lane markers in urban streets. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 7–12. IEEE, 2008.

[5] Xiangjing An, Erke Shang, Jinze Song, Jian Li, and Hangen He. Real-time lane departure warning system based on a single fpga. *EURASIP Journal on Image and Video Processing*, 2013(1):1–18, 2013.

[6] N. Apostoloff and A Zelinsky. Robust vision based lane tracking using multiple cues and particle filtering. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 558–563, June 2003.

[7] Nicholas Apostoloff and Alexander Zelinsky. Vision in and out of vehicles: Integrated driver and road scene monitoring. *The International Journal of Robotics Research*, 23(4-5):513–538, 2004.

[8] AAM. Assidiq, O.O. Khalifa, R. Islam, and S. Khan. Real time lane detection for autonomous vehicles. In *International Conference on Computer and Communication Engineering, 2008. ICCCE 2008.*, pages 82–88, May 2008.

[9] Athanasios Bamis, Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[10] Mario Bellino, Yuri Lopez De Meneses, Peter Ryser, and Jacques Jacot. Lane detection algorithm for an onboard camera. In *European Workshop on Photonics in the Automobile*, pages 102–111. International Society for Optics and Photonics, 2005.

[11] James R Bergen and Haim Shvaytser (Schweitzer). A probabilistic algorithm for computing hough transforms. *Journal of Algorithms*, 12(4):639 – 656, 1991.

[12] M. Bertozzi and A. Broggi. Real-time lane and obstacle detection on the gold system. In *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, pages 213–218, Sep 1996.

[13] Amol Borkar, Monson Hayes, and Mark T Smith. Robust lane detection and tracking with ransac and kalman filter. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3261–3264. IEEE, 2009.

[14] A Boukerche. *Handbook of algorithms for wireless networking and mobile computing*. CRC Press, 2005.

[15] Alledine Boukerche, HAB Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Secure localization algorithms for wireless sensor networks. *Communications Magazine, IEEE*, 46(4):96–101, 2008.

[16] Azzedine Boukerche and Mirela Sechi M Annoni Notare. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing*, 62(9):1476–1490, 2002.

[17] Azzedine Boukerche and Luciano Bononi. Simulation and modeling of wireless, mobile, and ad hoc networks. *Mobile ad hoc networking*, pages 373–409, 2004.

[18] Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. *Computer communications*, 29(4):477–489, 2006.

[19] Azzedine Boukerche, Xuzhen Cheng, and Joseph Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[20] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Analysis of a randomized congestion control scheme with dsdv routing in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61(7):967–995, 2001.

[21] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Swimnet: a scalable parallel simulation testbed for wireless and mobile networks. *Wireless Networks*, 7(5):467–486, 2001.

[22] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 618–624. IEEE, 2004.

[23] Azzedine Boukerche and Xin Fei. A voronoi approach for coverage protocols in wireless sensor networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 5190–5194. IEEE, 2007.

[24] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. An efficient synchronization scheme of multimedia streams in wireless and mobile systems. *Parallel and Distributed Systems, IEEE Transactions on*, 13(9):911–923, 2002.

[25] Azzedine Boukerche, Kathia Regina Lemos Jucá, João Bosco Sobral, and Mirela Sechi Moretti Annoni Notare. An artificial immune based intrusion detection model

for computer and telecommunication systems. *Parallel Computing*, 30(5):629–646, 2004.

[26] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communications*, 14(6):6–12, 2007.

[27] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[28] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 560–567. IEEE, 2005.

[29] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *Selected Areas in Communications, IEEE Journal on*, 27(4):387–399, 2009.

[30] Chao-Jung Chen, Bing-Fei Wu, Wen-Hsin Lin, Chih-Chun Kao, and Yi-Han Chen. Mobile lane departure warning systems. In *Consumer Electronics, 2009. ISCE'09. IEEE 13th International Symposium on*, pages 90–93. IEEE, 2009.

[31] Y. Chien. Pattern classification and scene analysis. *IEEE Transactions on Automatic Control*, 19(4):462–463, Aug 1974.

[32] Kuo-Yu Chiu and Sheng-Fuu Lin. Lane detection using color-based segmentation. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 706–711, June 2005.

[33] Kyuhyoung Choi, Kyungwon Min, Sungchul Lee, Wonki Park, Yongduek Seo, and Yousik Hong. Lane tracking in hough space using kalman filter. 2010.

[34] DO Cualain, C Hughes, M Glavin, and E Jones. Automotive standards-grade lane departure warning system. *IET Intelligent Transport Systems*, 6(1):44–57, 2012.

[35] R. Danescu and S. Nedevschi. Probabilistic lane tracking in difficult road scenarios using stereovision. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2):272–282, June 2009.

[36] C. D'Cruz and Ju Jia Zou. Lane detection for driver assistance and intelligent vehicle applications. In *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, pages 1291–1296, Oct 2007.

[37] Horacio ABF de Oliveira, Azzedine Boukerche, E Freire Nakamura, and Antonio Alfredo Ferreira Loureiro. An efficient directed localization recursion protocol for wireless sensor networks. *Computers, IEEE Transactions on*, 58(5):677–691, 2009.

[38] Jiayong Deng and Youngjoon Han. A real-time system of lane detection and tracking based on optimized ransac b-spline fitting. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, pages 157–164. ACM, 2013.

[39] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, 68(3):321–335, 2008.

[40] B Fardi, U Scheunert, H Cramer, and G Wanielik. A new approach for lane departure identification. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 100–105. IEEE, 2003.

[41] B. Fardi and G. Wanielik. Hough transformation based approach for road border detection in infrared images. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 549–554, June 2004.

[42] D. Felguera-Martin, J.-T. Gonzalez-Partida, P. Almorox-Gonzalez, and M. Burgos-Garcia. Vehicular traffic surveillance and road lane detection using radar interferometry. *Vehicular Technology, IEEE Transactions on*, 61(3):959–970, March 2012.

[43] You Feng, Wang Rong-ben, and Zhang Rong-hui. Research on road recognition algorithm based on structure environment for its. In *Computing, Communication,*

Control, and Management, 2008. CCCM '08. ISECS International Colloquium on, volume 1, pages 84–87, Aug 2008.

[44] S.G. Foda and AK. Dawoud. Highway lane boundary determination for autonomous navigation. In *Communications, Computers and signal Processing, 2001. PACRIM. 2001 IEEE Pacific Rim Conference on*, volume 2, pages 698–702 vol.2, 2001.

[45] M. Foedisch, R. Madhavan, and C. Schlenoff. Symbolic road perception-based autonomous driving in urban environments. In *Applied Imagery and Pattern Recognition Workshop, 2006. AIPR 2006. 35th IEEE*, pages 12–12, Oct 2006.

[46] Ján Gamec and Daniel Urdzík. Detection of driving space. *Acta Electrotechnica et Informatica Vol*, 8(4):60–63, 2008.

[47] David Hanwell and Majid Mirmehdi. Detection of lane departure on high-speed roads. In *ICPRAM (2)*, pages 529–536, 2012.

[48] Yinghua He, Hong Wang, and Bo Zhang. Color-based road detection in urban traffic scenes. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):309–318, Dec 2004.

[49] Rudra N. Hota, Shahanaz Syed, Subhadip Bandyopadhyay, and P. Radha Krishna. A simple and efficient lane detection using clustering and weighted regression. In *COMAD'09*, pages –1–1, 2009.

[50] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[51] Pei-Yung Hsiao, Chun-Wei Yeh, Shih-Shinh Huang, and Li-Chen Fu. A portable vision-based real-time lane departure warning system: Day and night. *Vehicular Technology, IEEE Transactions on*, 58(4):2089–2094, May 2009.

[52] Shih-Shinh Huang, Chung-Jen Chen, Pei-Yung Hsiao, and Li-Chen Fu. On-board vision system for lane recognition and front-vehicle detection to enhance driver's

awareness. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, pages 2456–2461 Vol.3, April 2004.

[53] K. Ishikawa, K. Kobayashi, and K. Watanabe. A lane detection method for intelligent ground vehicle competition. In *SICE 2003 Annual Conference*, volume 1, pages 1086–1089 Vol.1, Aug 2003.

[54] Liu Jia, Li Zheying, and Chen TingTing. Study on road recognition algorithm. In *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pages 2539–2541, May 2007.

[55] C.R. Jung and C.R. Kelber. A robust linear-parabolic model for lane following. In *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*, pages 72–79, Oct 2004.

[56] C.R. Jung and C.R. Kelber. An improved linear-parabolic model for lane following and curve detection. In *Computer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on*, pages 131–138, Oct 2005.

[57] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[58] A. Kanaris, E.B. Kosmatopoulos, and P.A. Loannou. Strategies and spacing requirements for lane changing and merging in automated highway systems. *Vehicular Technology, IEEE Transactions on*, 50(6):1568–1581, Nov 2001.

[59] Dong Jung Kang, Jang Won Choi, and In-So Kweon. Finding and tracking road lanes using ldquo;line-snakes rdquo;. In *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, pages 189–194, Sep 1996.

[60] Zu Kim. Realtime lane tracking of curved local road. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 1149–1155. IEEE, 2006.

[61] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 24(4):303 – 316, 1991.

[62] Suhong Ko, Seongchan Gim, Ce Pan, Jongman Kim, and Kihyun Pyun. Road lane departure warning using optimal path finding of the dynamic programming. In *SICE-ICASE, 2006. International Joint Conference*, pages 2919–2923, Oct 2006.

[63] Sachio Kobayashi, Tomoyoshi Aoki, Takuma Nakamori, and Naoki Mori. Vehicle and lane mark recognition apparatus, US Patent ep 1909230b1, 09 2008.

[64] Chris Kreucher and Sridhar Lakshmanan. Lana: a lane extraction algorithm that uses frequency domain features. *Robotics and Automation, IEEE Transactions on*, 15(2):343–350, 1999.

[65] Chris Kreucher, Sridhar Lakshmanan, and Karl Kluge. A driver warning system based on the lois lane detection algorithm. In *Proceedings of IEEE International Conference on Intelligent Vehicles*, pages 17–22. Stuttgart, Germany, 1998.

[66] Robert Laganiere. Compositing a bird's eye view mosaic. *image*, 10:3, 2000.

[67] Joon Woong Lee. A machine vision system for lane-departure detection. *Computer vision and image understanding*, 86(1):52–78, 2002.

[68] Guo Lei, Wang Jianqiang, and Li Keqiang. Lane keeping system based on thasv-ii platform. In *Vehicular Electronics and Safety, 2006. ICVES 2006. IEEE International Conference on*, pages 305–308, Dec 2006.

[69] Qing Li, Nanning Zheng, and Hong Cheng. Lane boundary detection using an adaptive randomized hough transform. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 5, pages 4084–4088 Vol.5, June 2004.

[70] Qing Li, Nanning Zheng, and Hong Cheng. Springrobot: a prototype autonomous vehicle and its algorithms for lane detection. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):300–308, Dec 2004.

[71] Qingquan Li, Long Chen, Ming Li, Shih-Lung Shaw, and A. Nuchter. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in

challenging road scenarios. *Vehicular Technology, IEEE Transactions on*, 63(2):540–555, Feb 2014.

[72] Xiangyang Li, Xiangzhong Fang, Ci Wang, and Wei Zhang. Lane detection and tracking using a parallel-snake approach. *Journal of Intelligent and Robotic Systems*, pages 1–13, 2014.

[73] P. Liatsis, J.Y. Goulermas, and P. Katsande. A novel lane support framework for vision-based vehicle guidance. In *Industrial Technology, 2003 IEEE International Conference on*, volume 2, pages 936–941 Vol.2, Dec 2003.

[74] Haiping Lin, Suhong Ko, Wang Shi, Yeongim Kim, and Hyongsuk Kim. Lane departure identification on highway with searching the region of interest on hough space. In *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pages 1088–1091. IEEE, 2007.

[75] C. Lipski, B. Scholz, K. Berger, C. Linz, T. Stich, and M. Magnor. A fast and robust approach to lane marking detection and lane tracking. In *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*, pages 57–60, March 2008.

[76] Jing-Fu Liu, Yi-Feng Su, Ming-Kuan Ko, and Pen-Ning Yu. Development of a vision-based driver assistance system with lane departure warning and forward collision warning functions. In *Digital Image Computing: Techniques and Applications (DICTA), 2008*, pages 480–485. IEEE, 2008.

[77] Weina Lu, Haifang Wang, and Qingzhu Wang. A synchronous detection of the road boundary and lane marking for intelligent vehicles. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, volume 1, pages 741–745, July 2007.

[78] Abdelhamid Mammeri, Azzedine Boukerche, and Guangqian Lu. Lane detection and tracking system based on the mser algorithm, hough transform and kalman filter. In *Proceedings of the 17th ACM International Conference on Modeling, Analysis and*

Simulation of Wireless and Mobile Systems, MSWiM '14, pages 259–266, New York, NY, USA, 2014. ACM.

[79] J. Mar and Hung-Ta Lin. The car-following and lane-changing collision prevention system based on the cascaded fuzzy inference system. *Vehicular Technology, IEEE Transactions on*, 54(3):910–924, May 2005.

[80] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using the progressive probabilistic hough transform. *Computer Vision and Image Understanding*, 78(1):119 – 137, 2000.

[81] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[82] Joel C McCall and Mohan M Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *Intelligent Transportation Systems, IEEE Transactions on*, 7(1):20–37, 2006.

[83] Xiaodong Miao, Shunming Li, and Huan Shen. On-board lane detection system for intelligent vehicle based on monocular vision. *International journal on smart sensing and intelligent systems*, 5(4):957–972, 2012.

[84] S. Nedevschi, R. Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, and C. Pocol. 3d lane detection system based on stereovision. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 161–166, Oct 2004.

[85] Marcos Nieto, Jon Arrspide Laborda, and Luis Salgado. Road environment modeling using robust perspective analysis and recursive bayesian segmentation. *Machine Vision and Applications*, 22(6):927–945, 2011.

[86] Horacio ABF Oliveira, Azzedine Boukerche, Eduardo F Nakamura, and Antonio AF Loureiro. Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm. *Performance Evaluation*, 66(3):209–222, 2009.

[87] W. Phueakjeen, N. Jindapetch, L. Kuburat, and N. Suvanvorn. A study of the edge detection for road lane. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*, pages 995–998, May 2011.

[88] D. Pomerleau. Ralph: rapidly adapting lateral position handler. In *Intelligent Vehicles '95 Symposium., Proceedings of the*, pages 506–511, Sep 1995.

[89] Jianqiang Ren, Yangzhou Chen, Le Xin, and Jianjun Shi. Lane detection in video-based intelligent transportation monitoring via fast extracting and clustering of vehicle motion trajectories. *Mathematical Problems in Engineering*, 01:1–12, 2014.

[90] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 2129–2133. IEEE, 2008.

[91] A Routray and K.B. Mohanty. A fast edge detection algorithm for road boundary extraction under non-uniform light condition. In *Information Technology, (ICIT 2007). 10th International Conference on*, pages 38–40, Dec 2007.

[92] Jiang Ruyi, Klette Reinhard, Vaudrey Tobi, and Wang Shigang. Lane detection and tracking using a new lane model and distance transform. *Machine Vision and Applications*, 22(4):721–737, 2011.

[93] Federal Highway Administration (FHWA) Roadway Departure Safety. United states department of transportation, washington, d.c. Accessed on, may 15, 2014., Website: http://safety.fhwa.dot.gov/roadway_dept/., 2013.

[94] F Samadzadegan, A Sarafraz, and M Tabibi. Automatic lane detection in image sequences for vision-based navigation purposes. *ISPRS Image Engineering and Vision Metrology*, 2006.

[95] Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. A predictive energy-efficient technique to support object-tracking sensor networks. *Vehicular Technology, IEEE Transactions on*, 60(2):656–663, 2011.

[96] R.K. Satzoda and M.M. Trivedi. Vision-based lane analysis: Exploration of issues and approaches for embedded realization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 604–609, June 2013.

[97] R.K. Satzoda and M.M. Trivedi. Drive analysis using vehicle dynamics and vision-based lane semantics. *Intelligent Transportation Systems, IEEE Transactions on*, PP(99):1–10, 2014.

[98] R.K. Satzoda and M.M. Trivedi. On performance evaluation metrics for lane estimation. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2625–2630, Aug 2014.

[99] A Saudi, J. Teo, Mohd Hanafi Ahmad Hijazi, and J. Sulaiman. Fast lane detection with randomized hough transform. In *Information Technology, 2008. ITSim 2008. International Symposium on*, volume 4, pages 1–5, Aug 2008.

[100] D. Schreiber, B. Alefs, and M. Clabian. Single camera lane detection and tracking. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 302–307, Sept 2005.

[101] D. Schreiber, B. Alefs, and M. Clabian. Single camera lane detection and tracking. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 302–307, Sept 2005.

[102] Stephan Sehestedt, Sarath Kodagoda, Alen Alempijevic, and Gamini Dissanayake. Efficient lane detection and tracking in urban environments. In *EMCR*, volume 01, September 2007.

[103] D. Shaked, O. Yaron, and N. Kiryati. Deriving stopping rules for the probabilistic hough transform by sequential analysis. In *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer Vision amp; Image Processing., Proceedings of the 12th IAPR International. Conference on*, volume 2, pages 229–234 vol.2, Oct 1994.

[104] Yu SHEN, Jianwu DANG, Enen REN, and Tao LEI. A multi-structure elements based lane recognition algorithm. *Przeglkad Elektrotechniczny*, 89:206–210, 2013.

[105] ASM Shihavuddin, K Ahmed, MS Munir, and Khandakar Rashed Ahmed. Road boundary detection by a remote vehicle using radon transform for path map generation of an unknown area. *International Journal of Computer Science and Network Security*, 8(8):64–69, 2008.

[106] S. Sivaraman and M.M. Trivedi. Integrated lane and vehicle detection, localization, and tracking: A synergistic approach. *Intelligent Transportation Systems, IEEE Transactions on*, 14(2):906–917, June 2013.

[107] N. Soquet, D. Aubert, and N. Hautiere. Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 160–165, June 2007.

[108] Miguel Angel Sotelo, Francisco Javier Rodriguez, Luis Magdalena, Luis Miguel Bergasa, and Luciano Boquete. A color vision-based lane tracking system for autonomous driving on unmarked roads. *Autonomous Robots*, 16(1):95–116, 2004.

[109] Hao Sun, Cheng Wang, and N. El-Sheimy. Automatic traffic lane detection for mobile mapping systems. In *Multi-Platform/Multi-Sensor Remote Sensing and Mapping (M2RSM), 2011 International Workshop on*, pages 1–5, Jan 2011.

[110] Tsung-Ying Sun, Shang-Jeng Tsai, and V. Chan. Hsi color model based lane-marking detection. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 1168–1172, Sept 2006.

[111] Akihiro Suzuki, Nobuhiko Yasui, Nobuyuki Nakano, and Mamoru Kaneko. Lane recognition system for guiding of autonomous vehicle. In *Intelligent Vehicles' 92 Symposium., Proceedings of the*, pages 196–201. IEEE, 1992.

[112] Min Tian, Fuqiang Liu, and Zhencheng Hu. Single camera 3d lane detection and tracking based on ekf for urban intelligent vehicle. In *Vehicular Electronics and Safety, 2006. ICVES 2006. IEEE International Conference on*, pages 413–418, Dec 2006.

[113] Quoc-Bao Truong and Byung-Ryong Lee. New lane detection algorithm for autonomous vehicles using computer vision. In *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pages 1208–1213, Oct 2008.

[114] Quoc-Bao Truong, Byung-Ryong Lee, Nam Geon Heo, Young Jim Yum, and Jong Gook Kim. Lane boundaries detection algorithm using vector lane concept. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 2319–2325, Dec 2008.

[115] M. Tsogas, A. Polychronopoulos, and A. Amditis. Using digital maps to enhance lane keeping support systems. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 148–153, June 2007.

[116] Masafumi Tsuji, Ryota Shirato, Hiroyuki Furusho, and Kiyoshi Akutagawa. Estimation of road configuration and vehicle attitude by lane detection for a lane-keeping system. *Navigation*, 2014:10–21, 2001.

[117] Vincent Voisin, Manuel Avila, Bruno Emile, Stephane Begot, and Jean-Christophe Bardet. Road markings detection and tracking using hough transform and kalman filter. In *Advanced Concepts for Intelligent Vision Systems*, pages 76–83. Springer, 2005.

[118] Hong Wang and Qiang Chen. Real-time lane detection in various conditions and night cases. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 1226–1231. IEEE, 2006.

[119] Jung-Ming Wang, Yun-Chung Chung, Shyang-Lih Chang, and Sei-Wang Chen. Lane marks detection using steerable filters. In *Proc. of 16th IPPR Conf. on Computer Vision, Graphics and Image Processing*, pages 858–865, 2003.

[120] Yanqing Wang, Deyun Chen, and Chaoxia Shi. Vision-based road detection by adaptive region segmentation and edge constraint. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 1, pages 342–346, Dec 2008.

[121] Yifei Wang, Naim Dahnoun, and Alin Achim. A novel system for robust lane detection and tracking. *Signal Processing*, 92(2):319–334, 2012.

[122] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4):269–280, 2004.

[123] Qinghua Wen, Zehong Yang, Yixu Song, and Peifa Jia. Road boundary detection in complex urban environment based on low-resolution vision. In *11th Joint International Conference on Information Sciences*. Atlantis Press, 2008.

[124] Bing-Fei Wu, Chuan-Tsai Lin, and Yen-Lin Chen. Dynamic calibration and occlusion handling algorithms for lane tracking. *Industrial Electronics, IEEE Transactions on*, 56(5):1757–1773, May 2009.

[125] Sibel Yenikaya, Gökhan Yenikaya, and Ekrem Düven. Keeping the vehicle on the road: A survey on on-road lane detection systems. *ACM Comput. Surv.*, 46(1):2:1–2:43, July 2013.

[126] A Yla-Jaaski and N. Kiryati. Adaptive termination of voting in the probabilistic circular hough transform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(9):911–915, Sep 1994.

[127] Bing Yu, Weigong Zhang, and Yingfeng Cai. A lane departure warning system based on machine vision. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 1, pages 197–201, Dec 2008.

[128] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000.

[129] Banggui Zheng, Bingxiang Tian, Jianmin Duan, and Dezhi Gao. Automatic detection technique of preceding lane and vehicle. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 1370–1375, Sept 2008.

[130] Shengyan Zhou, Yanhua Jiang, Junqiang Xi, Jianwei Gong, Guangming Xiong, and Huiyan Chen. A novel lane detection based on geometrical model and gabor filter. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 59–64, June 2010.