A Novel Road Marking Detection and Recognition Technique using a Camera-based Advanced Driver Assistance System

by

Zongzhi Tang

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in
Systems Science

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

Abstract

Advanced Driver Assistance System (ADAS) was widely learned nowadays. As crucial parts of ADAS, lane markings detection, as well as other objects detection, have become more popular than before. However, most methods implemented in such areas cannot perfectly balance the performance of accuracy versus efficiency, and the mainstream methods (e.g. Machine Learning) suffer from several limitations which can hardly break the wall between partial autonomous and fully autonomous driving.

This thesis proposed a real-time lane marking detection framework for ADAS, which included 4-extreme points set descriptor and a rule-based cascade classifier. By analyzing the behavior of lane markings on the road surface, a characteristic of markings was discovered, i.e., standard markings can sustain their shape in the perpendicular plane of the driving direction. By employing this feature, a 4-extreme points set descriptor was applied to describe the shape of each marking first. Specifically, after processing Maximally Stable Extremal Region (MSER) and Hough transforms on a 2-D image, several contours of interest are obtained. A bounding box, with borders parallel to the image coordinate, intersected with each contour at 4 points in the edge, which was named 4-extreme points set. Afterward, to verify consistency of each contour and standard marking, some rules abstracted from construction manual are employed such as Area Filter, Colour Filter, Relative Location Filter, Convex Filter, etc.

To reduce the errors caused by changes in driving direction, an enhanced module was then introduced. By tracking the vanishing point as well as other key points of the road net, a method for 3-D reconstruction, with respect to the optical axis between vanishing point and camera center, is possible. The principle of such algorithm was exhibited, and a description about how to obtain the depth information from this model was also provided. Among all of these processes, a key-point based classification method is the main contribution of this paper because of its function in eliminating the deformation of the object caused by inverse perspective mapping.

Several experiments were conducted in highway and urban roads in Ottawa. The detection rate of the markings by the proposed algorithm reached an average accuracy rate of 96.77% while F_1 Score (harmonic mean of precision and recall) also attained a rate of 90.57%. In summary, the proposed method exhibited a state-of-the-art performance and represents a significant advancement of understanding.

Acknowledgements

First and foremost, I would like to thank my parents for allowing me to realize my own potential. All the support they have provided me over the years was the greatest gift anyone has ever given me.

Next, I need to thank my supervisor, Dr. Azzedine Boukerche. With his patience and excellent expertise, he has 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 honor 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 research.

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. Guangqian Lu, who inspired me and shared a lot of knowledge on C++ and OpenCV with me. Thanks to Mingchang Zhao for helping me acknowledge the latest frame of OpenCV. Thanks to Yuhan Lee for helping me render many fine graphs. Thanks to Gary Cui for helping and proofreading of my manuscripts. Thanks to Run Ye, Peng Sun, Shichao Guan, Tao Zhang, Qiao Li and all those who contributed to the possibility to complete my master thesis.

Table of Contents

List of Tables

List of Figures

Nomenclature

F_1	Harmonic Mean of Precision and Sensitivity
ACC	Accuracy
ACF	Autocorrelation Function
ADAS	Advanced Driving Assistant System
ANNs	Artificial Neural Networks
AR	Autocorrelation
ARIMA	Autoregressive Integrated Moving Average Model
AT	Angle Coding+Topology Coding
AV	Autonomous Vehicle
BING	Binarized Normed Gradient
BW-HOG	Histogram of Oriented Gradients in Grayscale Image
CCH	Chain Code Histogram
CDNN	Convolutional Deep Neural Network
CFSS	Combination of Simple Shape Descriptors
CNNs	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
ELM	Extreme Learning Machines
FAST	Features From Accelerated Segment Test
FN	False Negatives
FP	False Positives
FPS	Frames per Second

GPS	Global Positioning System
GTSRB	German Traffic Sign Recognition Benchmark
HOG	Histogram of Oriented Gradients
HOV	High Occupancy Vehicle
HSI	Hue Saturation and Intensity Color Space
HSV	Hue Saturation Value
HT	Hough Transform
ICP	Iterated Closest Point
IPT	Inverse Perspective Transform
ITS	Intelligent Transportation System
LANA	Lane-Finding in Another Domain
LDT	Lane Detection and Tracking
LiDAR	Light Detection And Ranging
LMS	Laser Measurement System
LSD	Line Segment Detector
MA	Moving Average
MBR	Minimum Bounding Rectangle
MFT	Multi-resolution Fourier Transform
MLS	Mobile Laser Scanning
MLT	Median Local Thresholding
MRA	Automated Repainting Module
MSER	Maximally Stable Extremal Region
NNS	Nearest Neighbour Search
OBD	On-board Diagnostics
OCR	Optical Character Recognition
OS	Operating System
PACF	Partial Autocorrelation Function

PCA	principal Component Analysis
PCR	Principal Component Regression
PGH	Pairwise Geometric Histogram
PLSR	Partial Least Squares Regression
POI	Point of Interest
PPHT	Progressive Probabilistic Hough Transform
PPV	Precision or Positive Predictive Value
RANSAC	Random Sample Consensus
RAT	Refined Angle Coding+Topology Coding
RBF	Radial Based Function
RGB	Red Green Blue
ROC	Receiver Operating Characteristic Curve
ROI	Region of Interest
ROMA	Road Markings Analysis System
SAD	Sum of Absolute Difference
SAE	Society of Automotive Engineers
SCS	Street Coordinate System
SIFT	Scale-invariant Feature Transform
Snake	Active Contour Model
SPC	Specificity or True Negative Rate
SSD	Sum of Squared Differences
SVM	Support Vector Machines
TN	True Negatives
TP	True Positives
TPR	Sensitivity or True Positive Rate or Recall
VANETs	Vehicle Ad-Hoc Networks

Chapter 1

Introduction

Recently, autonomous vehicles(AV) have become a favoured research field. Both the automotive industry and academic interests are enthusiastic about the process of industrializing the production of autonomous vehicles. Since these technologies were designed to reduce the fatalities caused by accidents on the road, there have been a few tragedies caused by the autonomous vehicles itself. However, the general trend and direction towards forward progress that modem science and technology has applied to our daily lives is overwhelming. In this context, the only solution for autonomous vehicles with regards to the general public is to continue studying related technologies in depth thoroughly and steadfastly.

1.1 Background and Problems

According to the SAE internationals J3016 report [1], which was adopted by ONTARIO REGULATION 306/15 [19] in Canada, there are six levels of autonomous vehicles development. As shown in Table 1.1 , it covers a range from no automation (Level 0) to full automation (Level 5). So far in 2016, most of the major automobile manufacturers have already achieved Level 1 or Level 2 (e.g. Tesla Model S), which includes technologies like adaptive forward lighting, parking steering assistant, blind spot monitoring, lane keeping assist, automatic emergency braking, etc. Levels 1 and 2 automation technologies perform specific driving tasks and provide warnings in some situations, but the driver retains overall

Table 1.1: Levels for autonomous vehicles development [1]

Level 5	Full Automation
Level 4	High Automation
Level 3	Conditional Automation
Level 2	Partial Automation
Level 1	Driver Assistance
Level 0	No Automation

responsibility for monitoring the driving environment. In Levels 3, 4, and 5, automation crosses a critical threshold. At Level 3, Conditional Automation, the vehicle system takes full-time control in specific modes only, such as highway driving, dedicated autonomous vehicles lanes, or unobstructed city streets. This means from Level 3, the system should be designed to responsible for monitoring the driving environment.

AVs are a breed of advanced robots. Technologies like the Internet of Things, Big Data, and machine learning can be merged together under this topic. The capabilities of both the sensors and the data processing by the control units are continually growing, and highly advanced software is used to analyze this information in fractions of a second. In the future, passenger cars and commercial vehicles will have a complete image of the surroundings in real time.

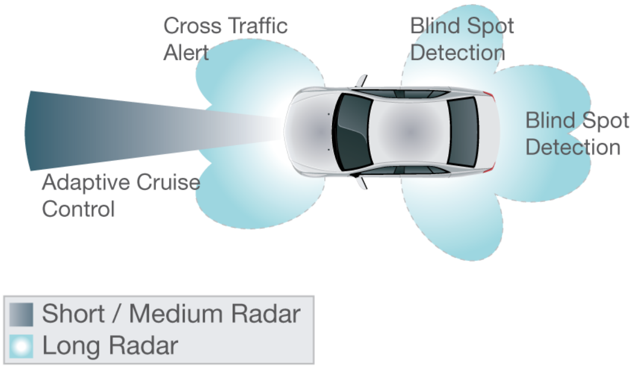


Figure 1.1: ADAS by Radars [3]

Radar sensors that are usually located in the front and rear of the vehicle can detect other vehicles and obstacles. The rear sensor detects traffic approaching from behind and vehicles that are overtaking. The traffic in front is monitored by long range radar. The short-range radar surveys the vehicle's immediate surroundings.

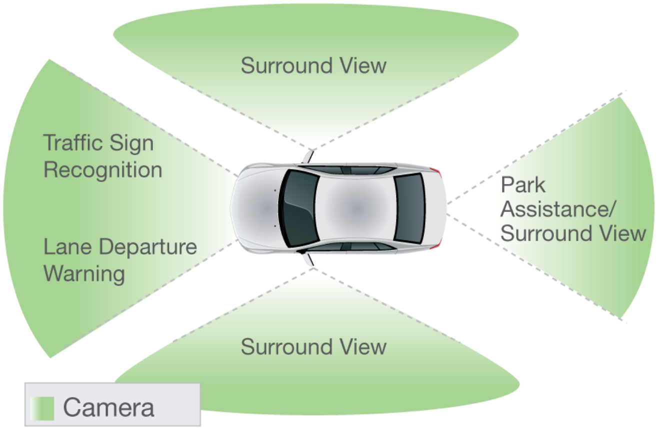


Figure 1.2: ADAS by Cameras [3]

Cameras are used, for instance, to recognize lane markings, traffic signs, traffic lights, and other road users.

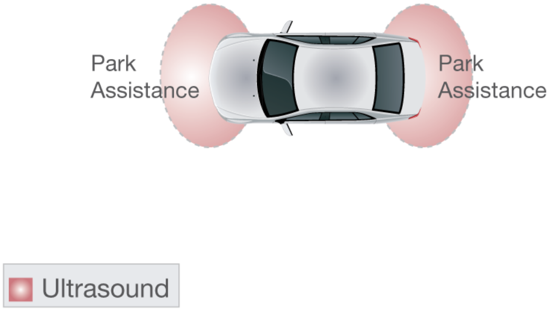
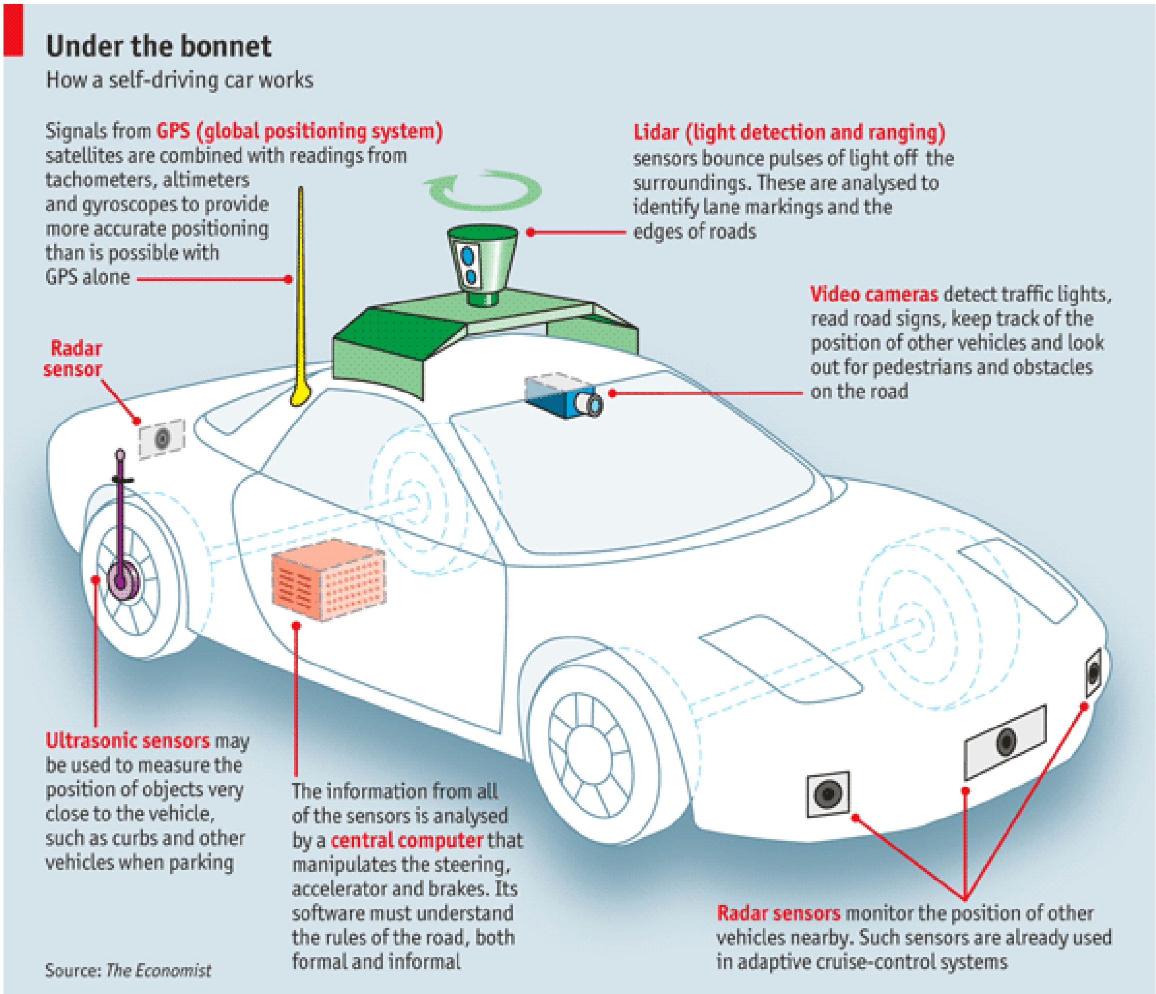


Figure 1.3: ADAS by Ultrasound Sensors [3]

Ultrasound sensors have been installed in vehicles from the beginning of the Nineties [20], to help drivers maneuver into parking spaces. Since then, their range of functions has increased markedly. They can measure parking spaces while the vehicle is in motion, and detect vehicles driving in an adjacent lane.

In the past, radar, cameras, and ultrasound sensors were used for separate functions, but now all the relevant data can be linked intelligently and simultaneously by sensor fusion. This is what makes automated driving possible in the first place.



Under the bonnet

How a self-driving car works

Signals from **GPS (global positioning system)** satellites are combined with readings from tachometers, altimeters and gyroscopes to provide more accurate positioning than is possible with GPS alone

Lidar (light detection and ranging) sensors bounce pulses of light off the surroundings. These are analysed to identify lane markings and the edges of roads

Video cameras detect traffic lights, read road signs, keep track of the position of other vehicles and look out for pedestrians and obstacles on the road

Radar sensor

Ultrasonic sensors may be used to measure the position of objects very close to the vehicle, such as curbs and other vehicles when parking

The information from all of the sensors is analysed by a **central computer** that manipulates the steering, accelerator and brakes. Its software must understand the rules of the road, both formal and informal

Radar sensors monitor the position of other vehicles nearby. Such sensors are already used in adaptive cruise-control systems

Source: *The Economist*

Figure 1.4: Autonomous vehicle integrated with sensors [4]

Better sensors are crucial for the success of fully autonomous vehicles. Advanced image sensors could reduce the dependence of autonomous vehicles on costly 3D LiDAR systems. Better image sensors could reduce the number of lasers within the rotating LiDAR systems. Googles current LiDAR sensors currently contain 64 lasers. However, it is unlikely that fully autonomous vehicles operating in urban contexts will be able to operate without any LiDAR sensors within the next few years.

There are two types of manufactures who are entering the autonomous vehicles market. On one hand, traditional automakers like Mercedes-Benz or General Motors choose to add driver assistance features to their existing fleets gradually, until eventually, their cars achieve high or full automation (Level 4 and above). Meanwhile high-tech companies, like Google or Uber, aim to design, test and sell fully automated vehicles from the very beginning.

As is shown in Table 1.2 , players in this specific market are demonstrating a commitment to release their level 4 autonomous car around 2020. A great deal of technical progress has been occurred as a result of the catfish effect produced by industry leaders such as Google Auto and Tesla Motors. The establishment of a production chain stretching through the automaker, Tier 1 supplier, on-board diagnostics (OBD) and embedded OS vendor, and semiconductor supplier, as well as the involvement of various government departments at all levels has produced an environment ready for the development of a new generation of autonomous vehicles. Although proposed solutions are diverse for each company or alliance, a common denominator can still be found amongst all solutions. For instance, an independent environment awareness system including LiDAR, camera, and millimeter wave radar is a basic necessity. A high definition three-dimensional map is a helpful supplement. Also, Vehicle to Vehicle communication, Vehicle to Infrastructure communication, and even Vehicle to Pedestrian or Vehicle to Cyclists communication have been introduced to improve accuracy and expand for highly scalable ITS (intelligent transportation system)-inspired applications.

Table 1.2: Autonomous Car Developing Schedule

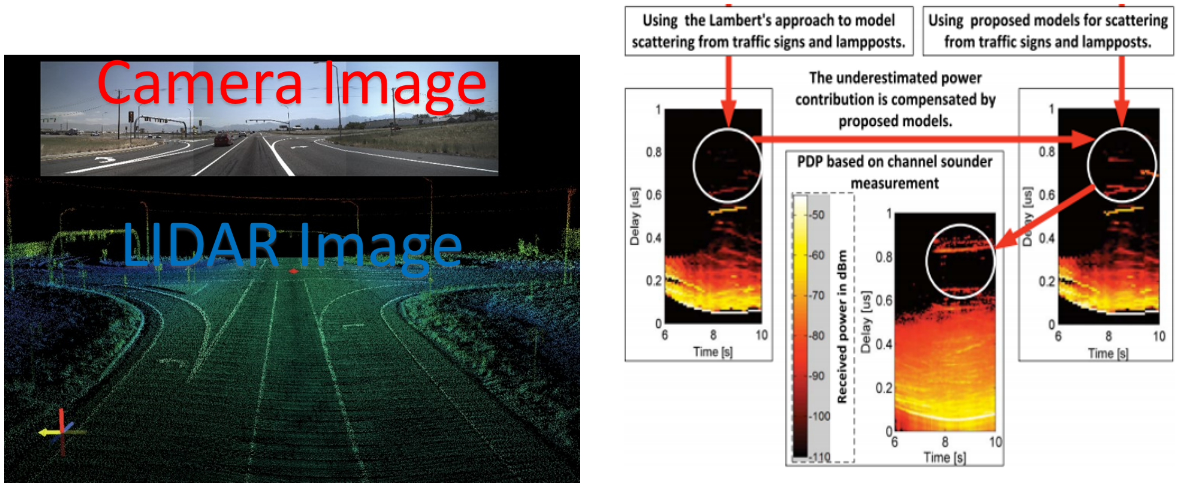
Country/ Region	Factory	Tech Solution	Academic Partner	Industry Partner
U.S.	GM	Camera+LiDAR	Carnegie Mellon University	Cruise Automation, Magna, Lyft, Mobileye
U.S.	Ford	Camera+LiDAR	University of Michigan, MIT, Stanford University, Aachen University	Ford Smart Mobility, Delphi, SAIPS, State Farm
U.S./ GERMAN	Daimler AG	Camera+LiDAR+MAP	Stanford University, Ulm University, Karlsruhe Institute of Technology	HERE MAP, Bosch, Car2go
GERMAN	Volkswagen Group	Monocular +MAP	Volkswagen Electronics Research Laboratory, Stanford University, MIT, UC Berkeley	HERE MAP,Mobileye, Delphi, SDS, NVIDIA
GERMAN	BMW	Binocular Vision System +MAP		HERE MAP, BAIDU, Intel, Mobileye
SWEDEN	VOLVO	Camera +Radar	Chalmers University of Technology	Microsoft, Uber
ITALY	F.I.A.T			
FRANCE	PSA Peugeot Citron	Camera +Radar	MINES ParisTech	TRW
FRANCE/ JAPAN	Renault-Nissan Alliance	Camera + Radar+ Ultrasonic sensor	Future Lab mobility research center, NASAs Ames Research Center	Mobileye
JAPAN	Toyota	LiDAR+ Camera + Radar+ Rotary encoder	Nagoya University, Stanford University, MIT, University of Michigan, Toyota Central R&D Labs	Delphi, Microsoft
JAPAN	Honda	LiDAR+ Camera + Radar	University of Maryland Honda Research Institute	
JAPAN	MITSUBISHI	LiDAR	Information Technology R& D Center, Mitsubishi Electric	
JAPAN	MAZDA			
JAPAN	SUBARU	Binocular Vision System +Radar+ MAP		Fuji Heavy Industries
KOREA	Hyundai	LiDAR+ Camera + Radar		Delphi, Cisco
U.S.	TESLA	Radar+Inferred Camera		Mobileye, NVIDIA
U.S.	GOOGLE	LiDAR+MAP	Stanford university	
U.S.	APPLE			
U.S.	UBER	LiDAR+ Camera + Radar	Carnegie Mellon University, University of Arizona	BING MAP
CHINA	FAW Group Corporation	Binocular Vision System	National University of Defense Technology	
CHINA	BAIDU	LiDAR+ MAP	Baidu Institute of Deep Learning	BMW, NVIDIA

Table 1.3: Autonomous Car Developing Schedule 2

Factory	Reference	Partial Automation Release Year	Full Automation Release Year
GM	[21] [22]	2015	2020
Ford	[23] [24]		2021
Daimler AG	[25] [26]	2015	2025
Volkswagen Group	[27] [28] [29] [30]		2017
BMW	[31]		2020
VOLVO	[32] [33]		2017
F.I.A.T			
PSA Peugeot Citron	[34] [35] [36]	2020	2021
Renault-Nissan Alliance	[37] [38] [39]	2018	2020
Toyota	[40]		2020
Honda	[41] [42]		2020
MITSUBISHI	[43] [44]		2020
MAZDA			
SUBARU	[45]	2020	
Hyundai	[46] [47] [48]	2020	2030
TESLA	[49]	2015	2020
GOOGLE	[50]		2020
APPLE			
UBER	[51]		2030
FAW Group Corporation	[52]		
BAIDU	[53]		2018

1.2 Motivation and Contribution

Human error when changing lanes on the road results in thousands of deaths every year [54]. Due to the divergent directions that autonomous vehicles technology has developed, multiple choices are provided by the market for sensors mounted on the vehicle. As is shown in Table 1.2, LiDAR, Radar and Camera are supposed to be the core components of ADAS by different manufactories respectively. However, camera was used by all of them since camera image can provides the highest resolution among all [55]. The most critical high-cost component among all is the LiDAR, a specialized remote sensor. According to Berman's [56] research, LiDARs have dropped in price by a factor of 10 since their 2007 introduction. Although the current price of $8,000 for a small-scale LiDAR will drop even more as autonomous vehicles proliferate, its price is still too high to afford for working families. As a result, if features of road lanes can be extracted based on the images captured by camera sensor independently without the use of LiDAR, the adoption rate for autonomous vehicles can be dramatically increased. However, the existing technologies and state-of-the-practice in this field suffer from some major drawbacks, including difficulty differentiating colours and shapes, especially in complex situations or low-light scenarios.



Camera and LiDAR Image

Radar Image [55]

Figure 1.5: Comparison of Camera, LiDAR, Radar Image

In response to such significant problems, a large amount of research regarding combined multimedia mobile application and vehicle network has been produced like ([57], [58], [59],

[60], [61], [62], [63], [64], [65], [66], [67], [68]). One simulation projects that "cooperative adaptive cruise control (CACC)" (that uses vehicle-to-vehicle communication) deployed at 10 percent, 50 percent, and 90 percent market penetration levels will increase lanes effective capacities by around 1 percent, 21 percent, and 80 percent, respectively [69].

As a significant factor of the self-driving system, "driver situational awareness" requires a capability of real-time processing and reacting to the objects appearing on the road captured by the sensor and the on-vehicle computer system. During this process, the emerging consensus is that LiDAR is good at depth information extraction (3-D moving objects) while cameras are much more efficient in pattern or light recognition (2-D stable objects).

In this thesis, we discuss a new method of road marking recognition using cameras with considerable low computation cost. Moreover, we have analyzed the feasibility of replacement of LiDAR sensor, and providing a more extensible solution of the self-driving system. Especially considering the fact that, comparing with LiDAR, the camera has competitive advantages not only in cost but also relies less on traditional mechanical structures. Specifically, my main contribution include:

- Introduction of MSER in line detection process;
- Development of a new method to scan the contours produced by MSER and collect exactly the edge points of non-ordered points set;
- Development of a new method to judge the shape of road lines;
- Development of a new method to judge the color of road lines dynamically despite illumination effect;
- Development of a new method to extract the feature of patterns on road surface like arrows and diamond;
- Analysis of the feasibility of using monocular vision system to obtain depth information without a high computation cost in the scenario of roads

1.3 Thesis Outline

The contents of this thesis are organized in the following chapters:

In Chapter 2, a literature review of ADAS, particularly the research method of lane marking detection in recent decades was cited following a common process including pre-processing, object detection, object recognition, and tracking. As the focus of proposed work is about innovation of descriptor and classifier in image processing, these two subjects constitute a large proportion of this chapter.

In Chapter 3, an MSER-based high-efficiency integrated solution was denoted. The main idea of the proposed method includes a creative descriptor of the 4 extreme points set and a rule-based cascaded classifier. Part of our previous work in line detection and recognition, has been included in the frame of the whole system. A vanishing-point-tracking based environment sensing module has been discussed to eliminate the error caused by instability of ego vehicle's direction and provide a direction of the road network to calibrate the whole process in the end. A Kalman filter was utilized for tracking key points in this stage.

Chapter 4 presents the methodology and results of experiments conducted using a platform including a laptop and Digital Video Recorder mounted on the vehicle. Ten clips of real life videos (including 32452 frames) have been collected around the City of Ottawa. The program built with VC++ and QT was run on the laptop and gave feedback to the drivers in real time. After each road test, both video records and logs were stored. Then a standard discriminant rules including True Positive Rate (TPR), Specificity (SPC), F-score, etc. was implemented on these experiment data to compare with other researchers' work.

In Chapter 5, a summary of the proposed algorithm as well as an analysis of advantages and disadvantages of this experiment was listed. In the meantime, some possible improvements and methods of enhancement were discussed in the future work.

Chapter 2

Related Work

To explore a better method for road marking detection, it is necessary to review the research in this field since the very beginning. After a comparison and analysis of the content among 81 research papers, it became clear that most studies followed four steps: pre-processing, detection, recognition and tracking. As is introduced by Gonzalez [70] since 1970s, pre-processing methods, such as intensity transformations and spatial filtering as well as frequency filtering, are widely used for the purpose of initial image enhancement. After that, detection (also known as segmentation) is applied to the appropriate extracted factor. Next, a region extracted in the previous stage would be representing by its external characteristics or internal characteristics using the corresponding feature descriptor. Object recognition techniques including quantitative descriptor matching and symbolic information comparison are then implemented. Finally, tracking of objects is utilized to improve the processing results in a clip of continuous video.

2.1 Preprocessing

In this stage, several powerful classic algorithms were applied by previous researchers. Among these algorithms, Region of Interest (ROI), image smoothing, binarization, and vanishing point extraction as well as inverse perspective transform were chosen as the subject of this review because they are widely used by the references in this field.

2.1.1 Region of Interest (ROI)

The extraction of region of interest is an important task in the preprocessing stage, aimed at reducing the computational cost due to the processing time. It is often unnecessary to process the entire image. Computation should be focused on regions which contain important information. Most of the traditional methods are in the frequency domain. As a result, ROIs are often defined as a belt area in the horizontal direction in [71]. (Shown in Fig. 2.1) Since spatial domain analysis are popular now with the advancement of computer processing power, researchers more frequently use a rectangular or triangular area instead.

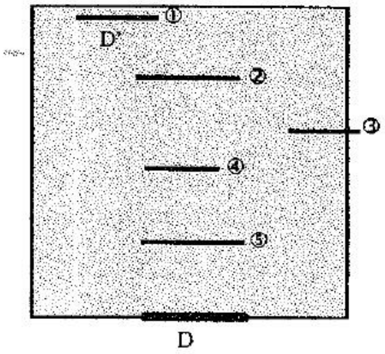
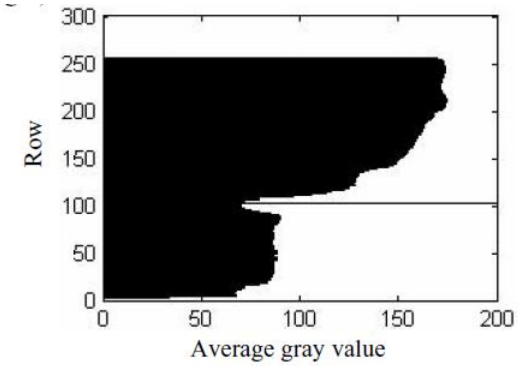


Figure 2.1: Belt Region of ROI by [5]

By utilizing the latter methods, more detail can be obtained with a holistic perspective. 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. For instance, Lu [6] used a threshold of average gray value in rows to find the skyline. (Shown in Fig. 2.2) Otherwise, fixed ROI close to the bottom area are also applied frequently like [7] [72]. (Shown in Fig. 2.3)

2.1.2 Image Smoothing

In signal processing, a filter is a device or process that removes some unwanted component or feature from a signal. Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. Most



Horizontal projection

(a) **(b)**

The division of the road surface and background

Figure 2.2: ROI under skyline in [6]

Figure 2.3: ROI defined in [7]

often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise. However, filters do not exclusively act in the frequency domain; time domain filters also exist, especially in the field of image processing.

Gaussian filter and median filter are two main methods in image processing. Both of them are often desirable to perform some kind of noise reduction on an image or signal. Amongst the two methods, gaussian filter is a low-band pass filter functioning in frequency domain while median filter is replaces each entry with the median of neighboring entries and works in time domain.

Table 2.1: Original 3×3 Image

1	1	1
1	2	1
1	1	1

Table 2.2: Weight of 3×3 Image

$w_{(1,1)}$	$w_{(1,2)}$	$w_{(1,3)}$
$w_{(2,1)}$	$w_{(2,2)}$	$w_{(2,3)}$
$w_{(3,1)}$	$w_{(3,2)}$	$w_{(3,3)}$

Since the process of weight calculating varies between different filters , the value of middle point $x_{(2,2)}$ can be calculate by the same linear function with respect of the weight of its neighbour.

$$V(x, y) = \sum_{i=1}^{3} \sum_{j=1}^{3} x_{(i,j)} w_{(i,j)} \tag{2.1}$$

Gaussian filter

Gaussian filter, also known as Gaussian smoothing operator in the field of image processing, are used to blur images and removed unwanted noise pixel. Usually these kind of pixels are described by convolution. Although it is a kind of low pass frequency filter, the performance is more due to frequency response from noise.

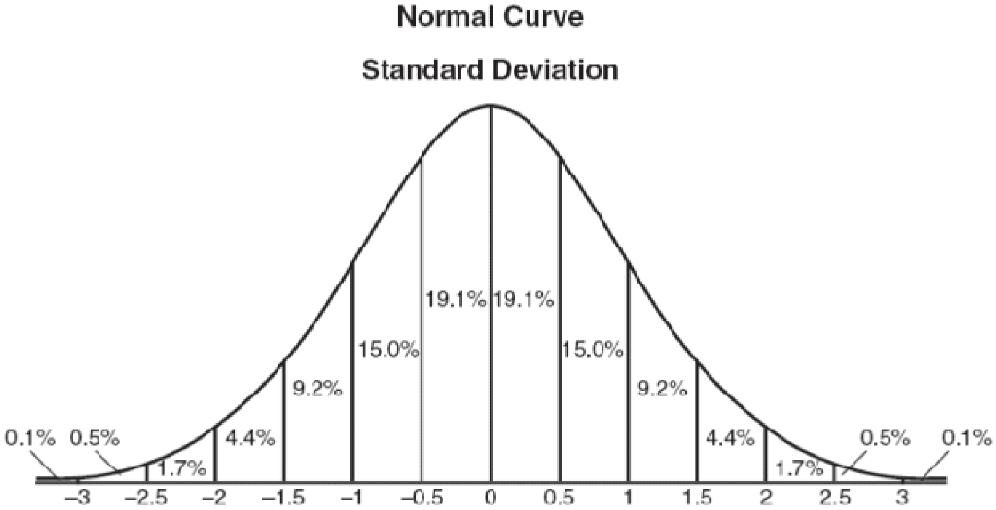


Figure 2.4: Normal Distribution

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.2}$$

Assume that $\sigma = 1.5$, then a new weight table can be obtained as:

Median filter

In some scenarios, nonlinear digital filtering techniques like median filter are applied. Non-Gaussian noise cannot be removed but can be reduced by Gaussian filter. These kinds of

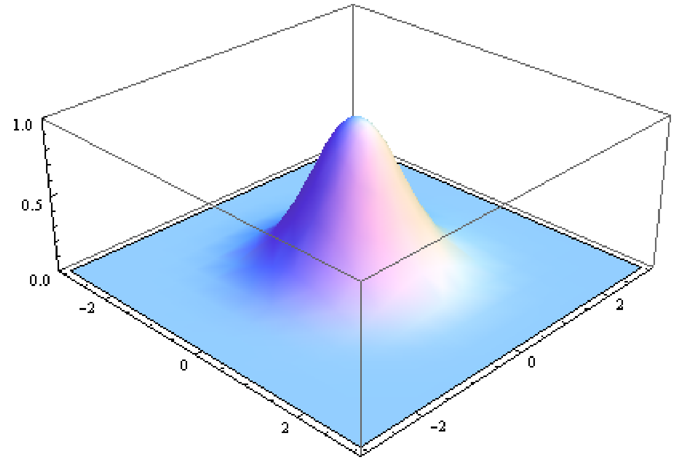


Figure 2.5: 2D Normal Distribution

Table 2.3: Weight Table by Gaussian Kernal

0.095	0.118	0.095
0.118	0.148	0.118
0.095	0.118	0.095

noises can be easily handled by median filter.

$$f(x,y) = med\{f(1,1), f(1,2) \ldots f(n,n)\}, n \in [1,N] \tag{2.3}$$

$med\{f(1,1), f(1,2) \ldots f(3,3)\} = 1$, then a new weight table can be obtained as:

Table 2.4: Weight Table by Median Kernal

0.111	0.111	0.111
0.111	0.111	0.111
0.111	0.111	0.111

Morphological filter

Top-hat filter and eroding/dilating are also known as basic morphological filters for preprocessing. Top hat filter is often used to correct uneven illumination. Eroding and dilating is often used to eliminate missing or joining disparate elements in an image.

Let $f : E \mapsto R$ be a grayscale image, mapping points from an Euclidean space or discrete grid E (such as R^2 or Z^2) into the real line. Let $b(x)$ be a grayscale structuring element.

Then, the top-hat transform of f is given by:

$$T_w(f) = f - f \circ b, \tag{2.4}$$

where $\circ$ denotes the opening operation.

The bottom-hat transform of f is given by:

$$T_b(f) = f \bullet b - f, \tag{2.5}$$

where $\bullet$ is the closing operation.

By applying Top-hat filter, the contrast of the original image can be obviously enhanced and the background can also be equalized even if the illumination of the original image is uneven.

2.1.3 Binarization

Otsu [73] presents a non-parametric and unsupervised method of automatic threshold selection for picture segmentation. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth and the first-order cumulative moments of the gray-level histogram. It is straightforward to extend the method to multi-threshold problems. Several experimental results are also presented to support the validity of the method. Said method was widely applied by Guan *et al.* [74] and Ding *et al.* [75].

2.1.4 Vanishing Point Extraction

Vanishing point detection is used to determine the region of interest in many papers, Magee *et al.*[76] describes a computationally inexpensive algorithm for the determination of vanishing points once line segments in an image have been determined. The approach is particularly attractive since it has no computationally degenerate cases and the only operations necessary are vector cross products and arc tangents. The need to know the

Algorithm 1: Otsu's method [73]

```
Input: A grayscale image with 256 gray-levels
Output: A threshold t
/* Weights $\omega_{0,1}$ are the probabilities of the two classes separated by a
   threshold t, $\sigma^2_{0,1}$ are variances of these two classes.            */
Compute histogram and probabilities of each intensity level
Set up initial $\omega_i(0)$ and $\mu_i(0)$
for all possible thresholds $t = 1 \ldots$ maximum intensity do
    Update $\omega_i$ and $\mu_i$ ;
    Compute $\sigma^2_b(t)$ ;
Desired threshold corresponds to the maximum $\sigma^2_b(t)$
```

distance to the focal plane is also eliminated thus avoiding tedious calibration procedures. Seo *et al.* [9] determine the ROI of a rectangle area based on vanishing point's position. As shown in 2.7, the green circle(circle in the middle) containing the "x" mark represents the detected vanishing point. The yellow horizontal line(the one above the rectangle) is the identified horizon line and the white vertical line indicates the horizontal center of the horizon line. The green line(reference middle line) between the vanishing point and the center of the image bottom is used to approximate the instantaneous driving direction of roadway. This kind of method was also applied by Kheyrollahi *et al.* [77], and Schreiber *et al.* [14].

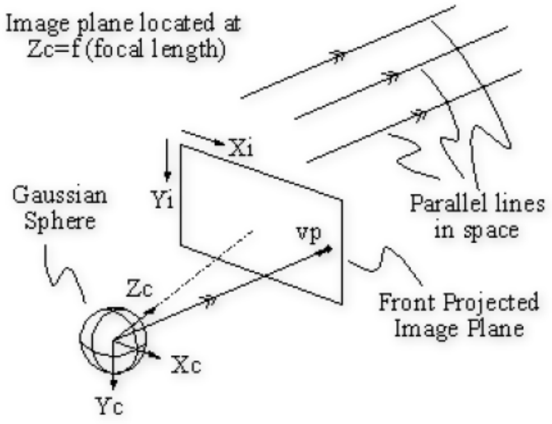


Figure 2.6: Gaussian Sphere Mapping [8]

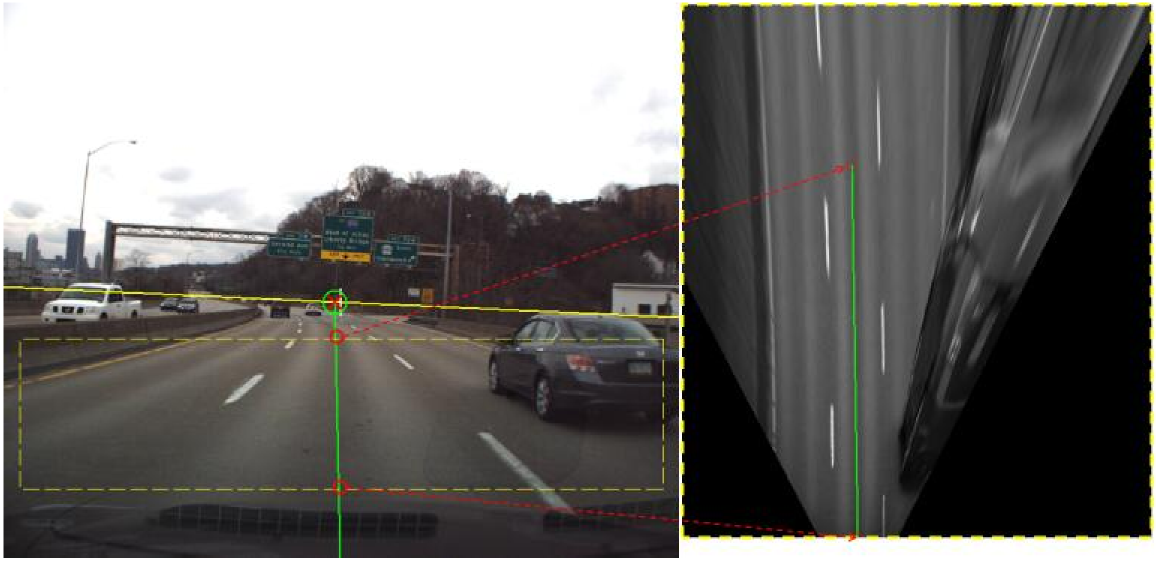


Figure 2.7: Vanishing Point Extraction Method by[9]

2.1.5 Inverse Perspective Transform

Inverse perspective transform (also known as IPM) is another tool used by researchers in order to gain a top-view (birds-eye view) image since many classifiers are adaptive to rotating, scaling, and translation but not distortion. In order to achieve this, multiple techniques are introduced. Among all these techniques, vanishing point is a factor of significant importance.

The width and shape of lane markings are often changed as a result of perspective effect. Baglidassarian *et al.* [78] presents a model with assumptions about object structures in the 3D scene and in the 2D image. It concerns the road structure and its geometry and contains several models: one of the road, one of the vehicle, one of the camera and a model of the arrows. For the model of the arrows, the features of invariant moments are described up to the 4th order. They also pointed that the inverse perspective projection is the weak point in the processing chain. Almost all of the latest work in the field utilized IPM as the initial processing method . For instance, Laganire [79] describes a method that allows the composition of an overhead view mosaic of a worksite from the views available. This global view of the site will help a teleoperator to adequately specify the actions to be performed in the realization of a given task. In his paper, P is a point in 3D space, $P = [X, Y, Z, 1]^T$

in world coordinate. Then, the corresponding point of P in image, p, can be represented as $p = [x, y, 1]^T$ in the image coordinate. There is also a relationship between these two points:

$$p = MP \tag{2.6}$$

$$M = C \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{2.7}$$

$$C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

Where R is a rotation matrix and T is a translation vector. C represents the intrinsic parameters of the camera, and f_x and f_y correspond to the focal length respectively measured in horizontal and vertical pixel units. The position $(c_x .\ c_y)$ is the principal point where the optical axis pierces the image plane.

IPM technology was used in almost all notable references as the first part of the algorithm including [12], [77], [80], [81], [82], [83], [10], [13], [84], [85], [86], [87], [88], [89], [90], [14], [9].

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, contour-based methods and position-based methods. Veit *et al.* [91] proposes a systematic approach to evaluate algorithms for extracting road marking features from images. They summarized existing road feature extractors and classified them with respect to the kind of feature they rely on:

- Geometric selection: segments;
- Geometric and photometric selection: pair of positive and negative gradients, response to a convolution filter at a given scale, ridges at a given scale;
- Photometric selection: light pixels, edges.

In this section, both geometric and photometric are take into consideration.

2.2.1 Feature Extraction Based on Edges

Sobel

The Sobel operator is used in image processing for edge detection since it creates an image emphasizing edges. Sobel operator uses two 3×3 kernels to calculate approximations of derivatives of the original image A, a matrix in horizontal G_x and one in vertical G_y respectively.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times A \tag{2.9}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times A \tag{2.10}$$

where $*$ denotes the 2-dimensional signal processing convolution operation. Through this method, both the "right"-direction and the "down"-direction are enhanced. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude following the formula $G = \sqrt{G_x^2 + G_y^2}$.

Mizuno *et al.* [92] notes that the dynamic range of a conventional TV camera is insufficient to properly represent colour and brightness in outdoor situations: for example, lane markings in the images are easily saturated in direct sunshine or darkened in the shade. Their research presents an improvement in robustness of the vision system for lane marking detection using the wide dynamic range vision sensor. Two long horizontal windows are set up in an acquired image. From images in these windows, horizontal edges are obtained by the Sobel operator. Candidates of the lane boundaries are obtained by thresholding the edge images. A pair of lines which satisfy that two lines are parallel with each other and the distance between them is suitable are extracted from the lines which classified every candidate into two windows.

Ding *et al.* [75] introduces their theory of algorithm results of different image processing techniques in lane marking and line recognition, which including edge detection and lane extraction using Sobel operator.

Canny

In addition to simple operators, the Canny operator was introduced by John Canny in 1986 [93]. As one of the best edge detectors, it successfully uses a non-maximum suppression and double threshold to modify the gradient magnitude matrix obtained by Roberts operator or Sobel operator. For non-maximum suppression, there are two steps. Firstly, the edge strength of the current pixel is compared with the edge strength of the pixel in the positive and negative gradient directions. Then, if the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction, the value will be preserved. Otherwise, the value will be suppressed.

For double threshold technology, a low threshold value T_l and a high threshold value T_h are set. A two threshold rule is then applied:

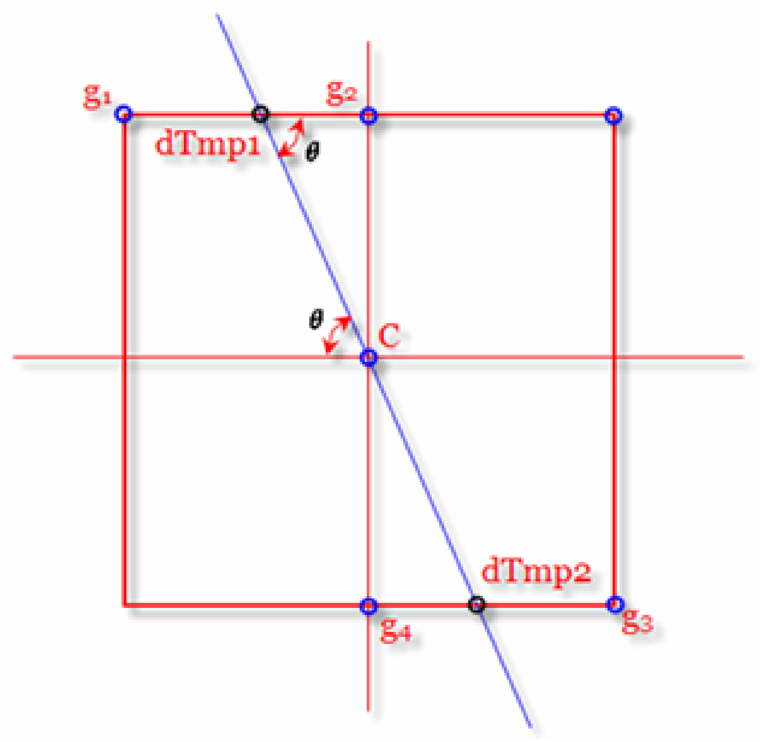


Figure 2.8: Non-maximum suppression

- If the strength of a pixel is smaller than T_l, it will be suppressed;
- If the strength of a pixel is larger than T_h, it will be marked as a strong edge pixel;
- If the strength of a pixel is between T_h and T_l, and at least one pixel of the eight surrounding pixels is larger than T_h, then the pixel will be marked as an edge pixel. Otherwise it will be suppressed.

Chang *et al.* [94] proposes a lane marking detection system which applies the Canny edge detector to investigate boundaries. Li *et al.* [81] combines local adaptive threshold and Canny edge detection for road marking extraction on urban streets , which outputs a binary image by separating desirable foreground objects from the background.

Hough transform

The Hough transform is widely used in image processing field. Taking a straight line as an example, the formula of a line in x-y coordinates is expressed as:

$$y = kx + b; \tag{2.11}$$

For each point (x_i, y_i)on the line, $y_i = kx_i + b$; In the meanwhile, in a $k - b$ coordinate system, this point can be expressed as a line $b = -x_i k + y_i$. Similarity, for points $(x_i, y_i) \neq (x_j, y_j) \neq (x_l, y_l)$,

$$\begin{aligned} k &= \frac{x_j - x_i}{y_j - y_i} = \frac{x_l - x_i}{y_l - y_i}, \\ b &= y_i - kx_i = y_j - kx_j = y_l - kx_l, \end{aligned} \tag{2.12}$$

Which means all the lines in $k - b$ coordinate related to a point belong to a line in $x - y$ coordinate though one point.

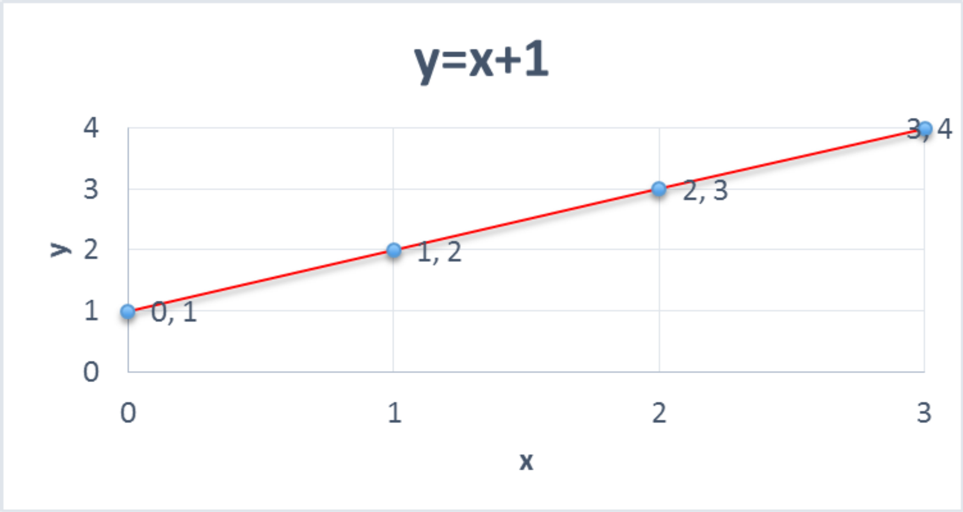


Figure 2.9: y=x+1 in X-Y coordinate

This method is the most commonly used edge extractor for line detection; There are two

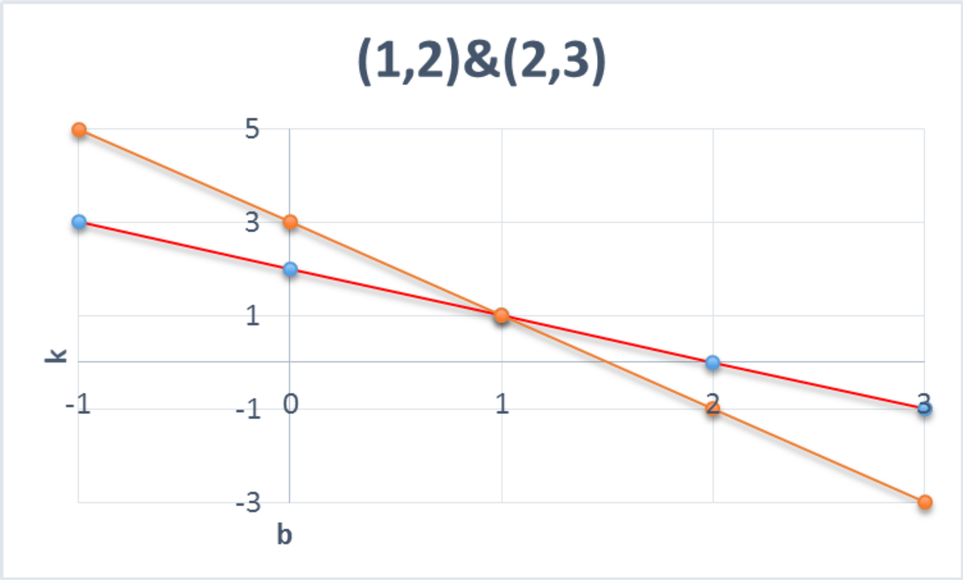


Figure 2.10: y=x+1 in k-b coordinate

names which commonly appear in previous works for this method: Radon Transform and Hough Transform. Ginkel *et al.* [95] gives a short introduction to the Radon and Hough transforms and how they relate to each other. They pointed out that the Hough transform is a discretization of the Radon transform and the later formalism has advantages over the former. (Shown in Fig. 2.11)

2.2.2 Feature Extraction Based on Contours

Segmentation

Brightness is a useful tool for analyzing the objects in the frame. Burrow *et al.* [96] focus on image segmentation in their research. They present five algorithms including variance method, moment preserving thresholding, minimum error thresholding, relaxation thresholding as global thresholding, and sub-image thresholding as local thresholding method, to segment road markings captured on digital images. These techniques seek to compute a gray-scale threshold value of pixels that distinguish any road markings present from their background. The associated computational processes have been presented, together with the results of a sensitivity analysis that was carried out to identify the optimum technique.

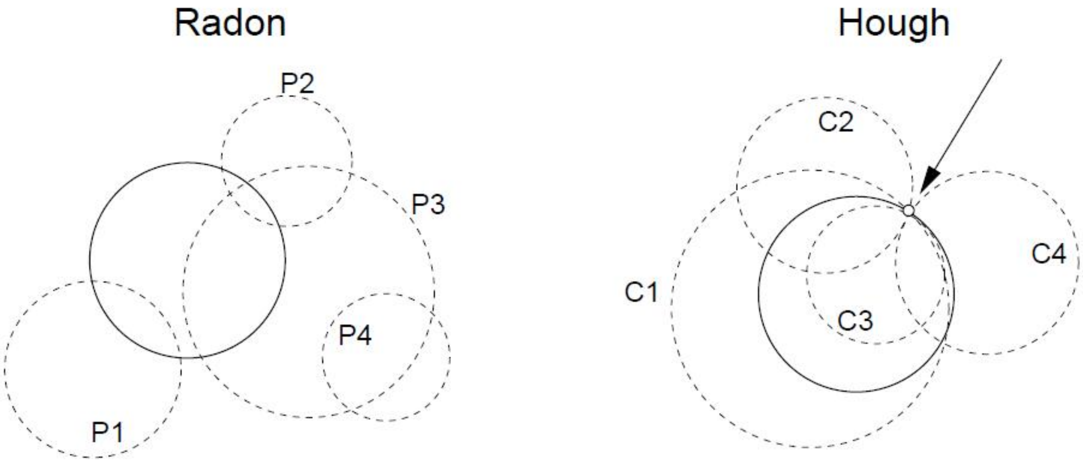


Figure 2.11: Difference between Radon Transform(left) and Hough Transform(right), former has integrating the intensity values along each of the candidate curves while later only considering individual points

It was found by them that the sub-image thresholding procedure performed satisfactorily on all images considered.

Lu *et al.* [6] showed that the dividing line between the road surface and background can be found through the brightness feature of the image. The average gray value of each row is calculated in their algorithm, so that the horizontal projection is formed. The row of the first minimum is the dividing line. Their research described a method designed to recognize the road boundary and lane marking. Firstly, the region of interest is dynamically partitioned from the image. Secondly, the feature pixels of road boundary are extracted by using the gradient vectors of pixels. Finally, the feature extraction of lane marking is finished by self-adapted threshold segmenting.

Ishino *et al.* [97] proposed a method of extracting road markings using aerial images and digital maps, and recognizing crosswalks and traffic lane lines. They also extract road markings using brightness values in their first step.

Pollard *et al.* [98] proposed an approach to improve the quality of lane marking extraction. They found the best extraction algorithms are all based on local thresholding, where lane feature extraction is based on the assumption that lane pixels form bright regions surrounded by darker regions. Consequently, their extractions purpose is to elaborate a way to detect these bright regions with a gradient of intensity higher than a threshold T_G with respect to the background.

Li *et al.* [81] described an integrated approach for image segmentation which combined local adaptive threshold and Canny edge detection and output a binary image by separating desirable foreground objects from the background.

He *et al.* [87] mentioned that they first transformed an original image into an orthoimage (through IPM) and binarized it with Median Local Thresholding (MLT) algorithm.

Snake

Active contour model, also known as Snake, is an algorithm for delineating an object outline from a 2D image. For instance, an image can be seen as a set of n points $\mathbf{v}_i$ where $i = 0 \ldots n-1$. To find the edge accurately, the internal elastic energy term $E_{internal}$ and the external edge-based energy term $E_{external}$ have been used to build an object function which defines the goodness of fitting as the same as minimization of $E_{internal}$ and $E_{external}$. Such object function can be represented as:

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))ds = \int_0^1 (E_{internal}(v(s)) + E_{image}(v(s)) + E_{con}(v(s)))ds \quad (2.13)$$

The purpose of the internal energy term is to control the deformations made to the snake, and the purpose of the external energy term is to control the fitting of the contour onto the image. The external energy is usually a combination of the forces due to the image itself E_{image} and the constraint forces introduced by the user E_{con}. The energy function of the snake is the sum of its external energy and internal energy.

Given an initial guess for a snake, the energy function of the snake is iteratively minimized. Gradient descent minimization is usually used to minimize snake energy. Each iteration takes one step in the negative gradient of the point with controlled step size γ to find local minima. This gradient-descent minimization can be implemented as:

$$\bar{v_i} \leftarrow \bar{v_i} + F_{snake}(v_i) \tag{2.14}$$

Where $F_{snake}(v_i)$ is defined by the negative of the gradient of the energy field.

$$F_{snake}(\bar{v_i}) = -\nabla E_{snake}(\bar{v_i}) = -\left(w_{internal}\nabla E_{internal}(\bar{v_i}) + w_{external}\nabla E_{external}(\bar{v_i})\right) \tag{2.15}$$

This model was used by Kass *et al.* [99] in their experiments. Wu *et al.* [10] used the snake algorithm to detect the positions of traffic signs on the IPM images captured by a stereo camera system. In their case, the contour was the road marking and the contrast in illumination between the markings and the road was sufficient for the algorithm to work reliably.

Connected Component Analysis

Connected component analysis is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. It is used in computer vision to detect connected regions in binary digital images, although color images and data with higher dimensionality can also be processed. Connectivity is determined by the medium; image graphs, for example, can be 4-connected or 8-connected.

Vacek *et al.* [100] successfully use the rules in Germany, where road markings follow strict regulations. A width of 12 cm is used for small markings and signals normal lane boundaries whereas 25 cm wide markings are used to indicate turning lanes or emergency lanes. For each lane, a region within the image is determined in the location where an arrow is expected. A segmentation of this region takes place in order to extract the arrow in the second step. The bounding box of the extracted arrow is then used for template matching with known arrows. The size of the region corresponds to the size of the biggest arrow plus a tolerance to cope with noise from the lane detection. Connected component analysis is applied in order to extract the largest component, which is the candidate for arrow estimation.

Chira *et al.* [85] describes a LabVIEW based system for detection, measurement and classification of painted objects. The system is able to calibrate monocular video sources, define the features of the painted objects that they look for, and detect them in real time using geometric pattern matching and edge detection. The segmentation is based on intensity levels and connected components analysis. This method was also used by Foucher*et al.* [86] in order to detect and recognize repetitive markings (such as crosswalks) as well as single patterns (such as arrows).

MSER

Maximally stable extremal regions (MSER) are used to detect blobs in images. The advantages of MSER includes a better performance in extracting objects completely while maintain the details like edges. Wu *et al.* [101] presents a system for detecting and recognizing road markings from video input obtained from an in-car camera. They use MSER features and perform the template matching in an efficient manner so that their system can detect multiple road markings in a single image. They detect MSERs on the rectified road scene images. Almost all regions containing road markings are detected, in addition to some regions in other parts of the scene. They extract a set of feature points from the regions of interest computed as explained above. To enable real-time computation without the use of GPUs, they use the FAST corner detector proposed by Rosten et al. Furthermore, Wu *et al.* [10] propose a method towards reliable visual localization using traffic signs painted on the road such as arrows, pedestrian crossings, and speed limits by a stereo camera system. These road markings are relatively easily detected since they are designed to be highly conspicuous. At runtime, inverse perspective rectification, MSER detection, FAST corner detection, and HOG descriptor computation are performed on each test image.

A. Mammeri, A. Boukerche and Z. Tang [102] also utilized MSER in pre-processing/detection stage. After analyzing the advantages of MSER and the requirements of line marking detection in the proposed scenario, an experiment was conducted to shown the performance of line tracking using a simpler framework.

2.2.3 Feature Extraction Based on Position

There are two types of image based on position: stereo images, which are usually obtained by the LIDAR/binocular camera system, and panoramic image which are usually obtained by a panoramic camera or a multi- camera system.

Stereo Position

Sebsadji *et al.* [103] evaluate different lane extraction algorithms for stereo images. Their second contribution is to propose stereo versions of extractors. The goal of stereo selection is to reduce the number of false alarms which may be fairly high in an urban environment. Finally, they compare the performances of the single-image and stereo algorithms. Concerning the extra stereo selection, the receiver operating characteristic curve (ROC curve) show that the stereo selection reduces the number of false alarms compared to the corresponding single-image algorithm, but it also slightly reduces the number of actual objects detected. Missed pixels by stereo selection are very likely due to variations of road plane (road slope, pitch angle...).

Guan *et al.* [74] presents an automated approach to detection and extraction of road markings (road line only) from mobile laser scanning (MLS) point clouds by taking advantages of multiple data features. To improve computational efficiency, the raw MLS point cloud data are first converted to geo-referenced images, based on elevation, intensity and point density, using inverse distance weighted interpolation. Afterwards, three filters are designed to extract road markings step-by-step: (1) the elevation filter is used to generate an elevation mask to remove high objects from the geo-referenced intensity image, (2) the point density filter is implemented to extract road surfaces in the geo-referenced intensity image, (3) the filtered geo-referenced intensity image is processed by thresholding and point density to obtain road markings, followed by a Canny detector and Hough transform used to extract straight-lines of road markings. Two RIEGL VMX-450 datasets demonstrated that the proposed multi-feature road marking extraction method performs well in road marking extraction from a large volume of mobile laser scanning data.

Wu *et al.* [10] propose a method which provides an absolute global localization if the road markings have been surveyed beforehand, and relative positioning information otherwise. They demonstrate using experiments and with ground truth data that their method provides accurate lane-level visual localization under various lighting conditions and using various types of road markings. During runtime, the positions of these corners are then detected using the snake algorithm on the IPM images.

Cui *et al.* [104] propose an accurate and real-time positioning method for intelligent road vehicles in urban environments. The proposed method uses a robust lane marking detection algorithm, as well as an efficient shape registration algorithm between the detected lane markings and a pre-existing GPS based road shape, to improve the robustness and accuracy of global localization of a road vehicle. They exploit both the technologies of visual localization based on lane marking detection and the wide availability of Global Positioning System (GPS) based localization. By formulating the positioning problem in a relative sense, they can estimate the vehicle localization in real-time and bound its absolute error in centimeter-level by a cross validation scheme. The validation scheme integrates the vision based lane marking detection with the shape registration, and improves the performance of the overall localization system. The GPS localization can be refined by using lane marking detection when the GPS suffers from frequent satellite signal masking or blockage, while lane marking detection is validated and complemented by the pre-existing GPS based road shape when it does not work well in adverse weather conditions or with poor lane signature.

Hernandez *et al.* [105] proposes a method to detect road lane markings for safe autonomous navigation purpose. It focuses on unconventional methods of identifying lane markings on a road surface through Laser Measurement System (LMS).

Panoramic Position

While for the latter, Li *et al.* [90] propose an algorithm to detect and recognize road lane markings from panoramic images. The dataset that they used to evaluate the performance of their algorithm is captured by a camera mounted on the roof of a driving vehicle, and a panoramic image is captured every 5 meters under very good weather conditions. A panoramic image contains a road scene with a resolution of 4800 ×2400 pixels. They use 910 highway panoramic images under various lighting conditions, and manually annotated all road markings that are completely visible on the top view images of the 910 highway panoramic images. The road markings in their dataset are not damaged or faded, and the images are taken during the day-time in fair weather conditions. A similar method is also applied by Hazelhoffa *et al.* [106] in a conceptual system for the automated creation of such combined databases and investigates the benefit of this combination for the specific case of pedestrian crossings.

Besides, Seo *et al.* [9] utilize the driving direction of a roadway to filter out false positives in a principled way. They mentioned longitudinal lane-markings delineate the driving direction of a road and the orientations of any true, longitudinal lane-markings appearing on input images should be aligned with this direction. To approximate the driving direction of a road, they detect the vanishing point on a horizon line and draw a line to link the image coordinates of the detected vanishing point to those of the center of the image bottom. They then filter out any lane-marking blobs if their orientations are not aligned with that of the approximated driving direction.

2.3 Recognition Methods

For further refinement of the detection result, descriptors and classifiers are used together to get the result of an integrated algorithm.

2.3.1 Feature Description

Li *et al.* [5] present a survey of recent progress and advances in visual feature detection. They describe the relations among edges, corners and blobs from the psychological view and classify the algorithms in detecting edges, corners and blobs into different categories and provide detailed descriptions for representative recent algorithms in each category. In this section, a similar framework was built to classify the research about feature extraction method, however their "Interest Point" in Fig. 2.12 was combining with "Corner" into point based descriptor. Region based descriptor in this chapter only contains the method focus on blobs itself.
Range [107] compared 5 popular irregular shape matching techniques, including Chain Code Histogram(CCH), Pairwise Geometric Histogram(PGH), Combination of Simple Shape Descriptors(CFSS) and Hausdoff distance and Hu-Moment, to obtain Table 2.5, which is a good supplement to [5].

Point based Descriptor

Kendall [108] gives a definition of "shape" for a set of k points in m dimensions. The first task is to identify the shape spaces in which such objects naturally live, and then to examine the probability structures induced on such a shape space by corresponding structures in R^m. They also outline their work on "size-and-shape", on shapes of sets of points in Riemannian spaces, and on shape-theoretic aspects of random Delaunay tessellations.

Maier *et al.* [109] creates a point list in street coordinate system (SCS), which represents the re-projected contour list of the arrow candidate. For that purpose, high precise camera calibration is needed, enriched by a video-based ego motion estimation in order to

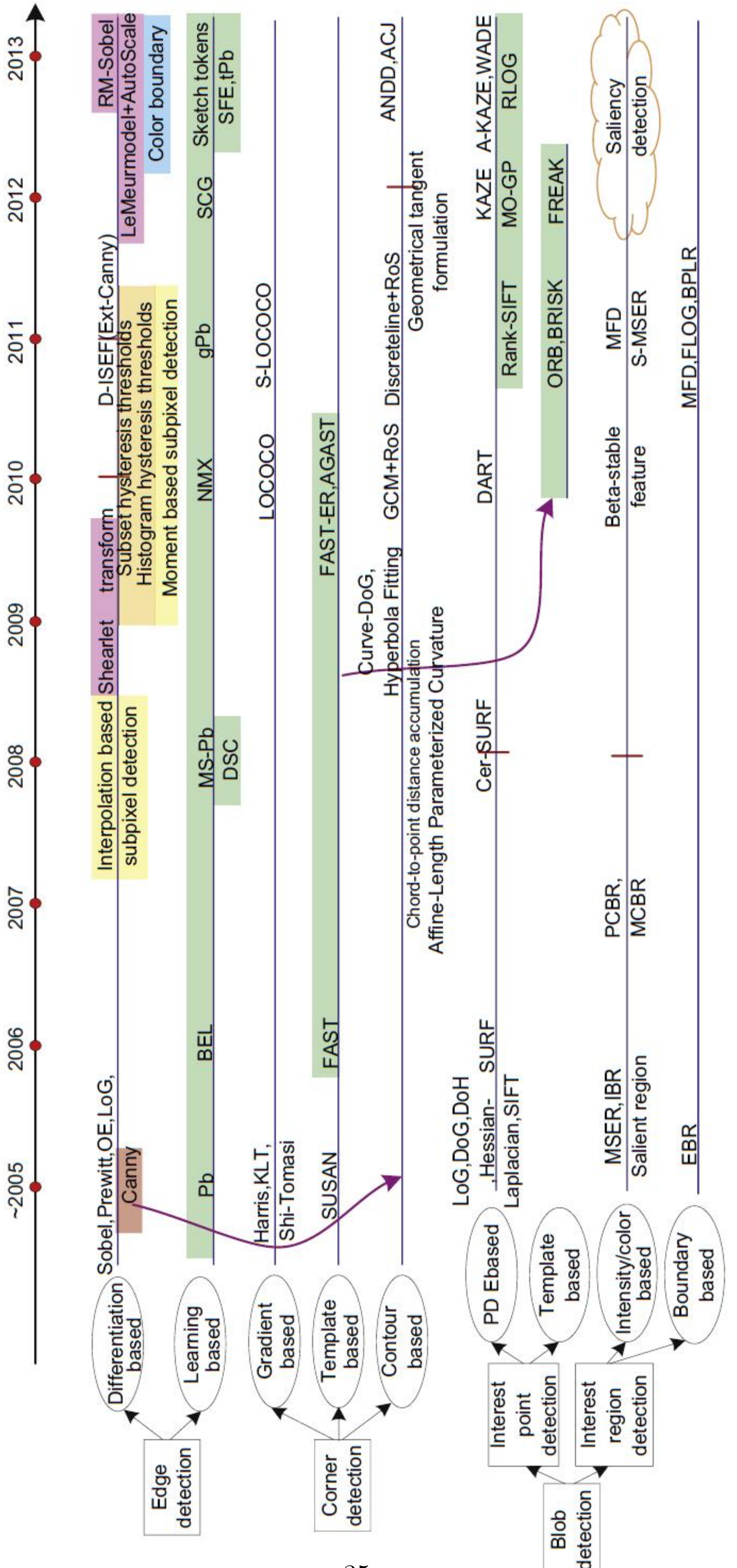


Figure 2.12: Recent representative advances in visual feature detection by [5]

Table 2.5: Comparison of Shape Matching Algorithms

	Method	**CLASS**	**Translation**	**Scaling**	**Rotation**	**Distortion**	**Efficiency**	**Condition**
Shape encoding	CCH	Contour Based	Good	Fair	$\frac{\pi}{2}$	Weak	Very very fast	Shapes with Obvious difference
	PGH	Contour Based	Good	Good	2π	Weak	Fast	Polygonal Objects, partial closure allowed
	CFSS	Contour Based	Good	Good	2π	Weak	Very fast	Similar to PGH, perform a little better
Non-shape encoding	Hausdoff	Contour Based	Good	Good	$>\frac{\pi}{2}$	Weak	Slow	Enable to match partial coincident objects
	HU Moment	Region Based	Good	Good	2π	Weak	Normal	More precise

compensate for short term movements of the vehicle such as pitching and rolling.

Wu *et al.* [10] applied FAST corner detection with markings detected by stereo cameras. (Shown in Fig. 2.13)

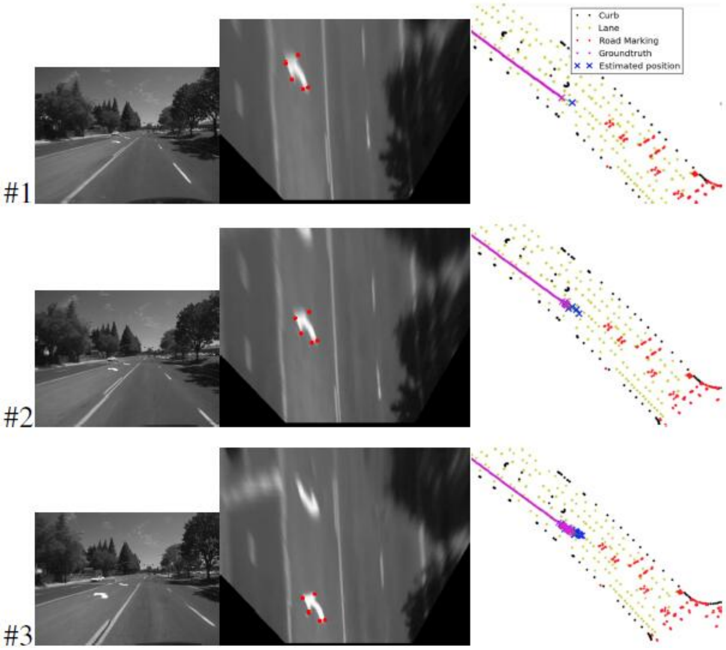


Figure 2.13: The input image, the inverse perspective mapped image, and the vehicle position shown against the surveyed map data in [10]

Edge based Descriptor

Baglidassarian *et al.* [78] presents a model concerning the road structure and its geometry. They use the algorithm of Wagner-Fisher which measures the likeness of two chains obtained from marking detected. They also mention in their paper that the inverse perspective projection is the weak point in the processing chain.

Frank [71] tried to solve one of the main tasks of road network managers, repainting used/deteriorating road markings. As introduced by him, the MRA project (which stands for automated repainting module) was started in France in order to increase the productivity and quality and to improve the work conditions of the agents working on the repainting process. It consists of the development of Marodeur's automation. The result obtained by them was a polygonal decomposition of the forms whose analysis identifies the kind of road marking. Once the road markings are extracted from the image, their method consists of analysis of a square shaped window on the road plane. This analysis starts from the bottom of the image up to a line 30 meters ahead.

Charbonnier *et al.* [110] present an image analysis method based on a segmentation followed by a polygonal shape overlapping for road marking recognition.

Ieng *et al.* [111] use a lane marking features extractor followed by the robust fitting algorithm to estimate the lane marking shape as a single analytical curve. Their lane marking features extractor is based on the fact that lane marking widths are in a small range of possible values on a road. This implies a geometric constraint on the observed lane marking widths from a camera onboard a vehicle. Their lane marking features extractor uses this property to select pairs of edge points corresponding with a high probability to a section of lane marking. The extracted features are then grouped to estimate the parameters of the analytical curve model of those lane markings.

Rebut *et al.* [112] propose a method where rectilinear marking and arrow extraction is carried out by mathematical morphology in order to take into account geometric characteristics (size for example). They also use Fourier descriptors to describe the boundaries of an object. In their case, approximated by polygonal curves, Fourier descriptors are used to represent the shape in frequency domain, with general information in lower frequencies and details in higher frequencies.

Ishino *et al.* [97] propose a method of extracting road markings using aerial images and digital maps, and recognizing crosswalks and traffic lane lines. They extract crosswalks and traffic lane lines using edge directions and shape features.

Gioi *et al.* [113] propose a linear-time line segment detector which requires no parameter tuning. In contrast to classic edge detectors, their method defines a line segment as an image region, the line-support region, namely a straight region whose points share roughly the same image gradient angle. Such line segments are roughly oriented along the average level-line direction.

Chira *et al.* [85] describes a LabVIEW based system for detection of painted objects. Their system detects objects in real time using edge detection. To detect the continuous lane markings, they use LabVIEWs Find Edge function which finds straight edges (lines) within a region in an image. This function includes parameters like search direction, edge polarity, kernel size, minimum edge strength, interpolation type.

Suchitra *et al.* [2] propose a method to robustly identify and classify arrow markings in road images. In their method, simple and unique signatures are first derived for the various arrow types, based on signed edge maps and decomposing the arrows into smaller parts. The signed edge maps are processed using Hough Transform (HT), and the resulting Hough spaces are analyzed systematically, using a set of simple rules. Edge pairs are sorted by Hough transform in the left side and right side separately, and then classified by rules (list in table 2.6).

Table 2.6: Classifier using Edges pair [2]

Arrow Part	**Right Slant**	**Left Slant**
Main Arrow Tail	1 in E_l	1 in E_r
Forward Arrow Head	1 in E_l	1 in E_r
Left Arrow Head	1 in E_l	1 in E_l
Right Arrow Head	1 in E_r	1 in E_r

Li *et al.* [81] describes a method for road marking extraction on urban streets acquired by a camera mounted on a moving vehicle. An integrated approach for image segmentation is presented by them which combines local adaptive threshold and canny edge detection, and outputs a binary image by separating desirable foreground objects from the background.

Chang *et al.* [94] propose a lane-mark detection system which applies the Canny edge detector to investigate boundaries. In order to remove the noise edges, they divide the boundary image into sub-images to calculate local edge-orientation of each block and remove the edge with abnormal orientation.

Ding *et al.* [75] introduce their theory of algorithmic results of different image processing techniques in lane marking line recognition, including binarization, edge detection and lane extraction (road line only). They compare various combinations of algorithms to detect lane marking lines in several conditions and then propose an ideal detection algorithm which uses the progressive threshold method to binarize, the contour extraction method, lane edge tracing, and the curve fitting method.

Guan *et al.* [74] presents an automated approach to detection and extraction of road markings from mobile laser scanning (MLS) point clouds. Three filters are designed to extract road markings step-by-step:

- (1) The elevation filter is used to generate an elevation mask to remove high objects from the geo-referenced intensity image,
- (2) The point density filter is implemented to extract road surfaces in the geo-referenced intensity image,
- (3) The filtered geo-referenced intensity image is processed by thresholding and point density to obtain road markings, followed by a Canny detector and Hough transform used to extract straight-lines of road markings.

He *et al.* [87] propose a special local junction feature (L-junction) to describe each road marking as a junction string, where each different deviation is dispersed into a different junction. For each potential arrow road marking, they represent its boundaries as a sequence of L-junctions and further encode these L-junctions counter-clockwise w.r.t. their angles and mutual position relations. L-junction is a newly defined corner feature, with three properties: one location and two branches. The location is the intersection point of two LSD lines, the two branches follow along the boundaries around the location and show the corners geometrical feature.

Contour based Descriptor

Zhang *et al.* [11] proposes a fast parallel algorithm for extracting the skeleton of a picture consisting of removing all the contour points of the picture except those points that belong to the skeleton. In order to preserve the connectivity of the skeleton, they divide each iteration into two sub-iterations that remove the boundary and corner points respectively.

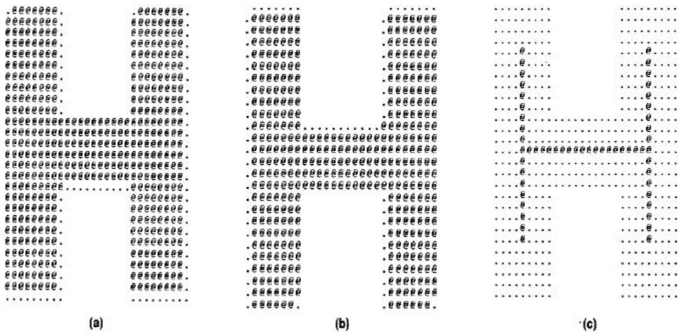


Figure 2.14: Results of thinning "H" in [11].

Baglidassarian *et al.* [78] presents a model with assumptions about object structures in the 3D scene and in the 2D image. It concerns the road structure and its geometry. For models of the arrows, the features of invariant moments they use are described up to the 4th order. These moments are calculated from the Hu moments [114], invariant to scale, translation and rotation. The centroid normalized moments mcn_i are obtained from centroid moments mc_i.

The method used here was invented by Hu [114] in 1962, which determines the orthogonal invariant moments using the theory of algebraic moment. For a 2D continuous function $f(x, y)$ the moment (sometimes called "raw moment") of order (p + q) is defined as

$$M_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y) dxdy, for p, q = 0, 1, 2, ...n \tag{2.16}$$

Adapting this to a scalar (grayscale) image with pixel intensities $I(x, y)$, raw image moments $M_{i,j}$ are calculated by

$$M_{i,j} = \sum_x \sum_y x^i y^j I(x,y) \tag{2.17}$$

Hu states that if $f(x, y)$ is piecewise continuous and has nonzero values only in a finite part of the x-y plane, moments of all orders exist, and the moment sequence (M_{pq}) is uniquely determined by $f(x, y)$. Conversely, $(M_{p,q})$ uniquely determines $f(x, y)$. In practice, the image is summarized with functions of a few lower order moments. Central moments are defined as

$$\mu_{p,q} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y) dxdy \tag{2.18}$$

where $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$ are the components of the centroid.
If $f(x, y)$ is a digital image, then the previous equation becomes

$$\mu_{p,q} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{2.19}$$

It can be shown that:

$$\mu_{p,q} = \sum_m^p \sum_n^q \binom{p}{m}\binom{q}{n} (-\bar{x})^{p-m} (-\bar{y})^{q-n} M_{mn} \tag{2.20}$$

The covariance matrix of the image I(x,y) is

$$cov\,[I(x,y)] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} \tag{2.21}$$

Invariants η_{ij} with respect to both translation and scale can be constructed from central moments by dividing through a properly scaled zero-th central moment:

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{\left(1+\frac{i+j}{2}\right)}} \tag{2.22}$$

where $i + j \geq 2$. Note that translational invariance directly follows by only using central moments.

As shown in the work of Hu, moment invariants with respect to translation, scale, and rotation can be constructed:

$$\begin{aligned}
I_1 =& \eta_{20} + \eta_{02}; \\
I_2 =& (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2; \\
I_3 =& (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2; \\
I_4 =& (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2; \\
I_5 =& (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
& + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]; \\
I_6 =& (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}); \\
I_7 =& (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
& - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2];
\end{aligned} \tag{2.23}$$

These are well-known as Hu moment invariants.

The first one, I_1, is analogous to the moment of inertia around the image's centroid, where the pixels' intensities are analogous to physical density. The last one, I_7, is skew invariant, which enables it to distinguish mirror images of otherwise identical images.

Charbonnier *et al.* [110] presents an image analysis method for road marking recognition. A high-level process allows the grouping of isolated quadrangles into marking lines that can be analyzed to discard non-linear markings and to determine parameters (like standard dashes modulation or line width).

Li *et al.* [12] proposes a shape-based road marker detection and recognition method. They obtain the road image in plane coordinate system through inverse perspective transform and then abstract 8-neighbor chain codes of the close regions from the plane image. Finally, they chose Bhattacharyya distance as the measure of divergence, which is defined as

$$J_B = -ln \int [p(x|\omega_1)p(x|\omega_2)]^{\frac{1}{2}} dx \tag{2.24}$$

Assume a multivariate normal distribution for features of each class. Thus the Bhattacharyya distance J_B may be written as:

$$J_B = \frac{1}{8}(\mu_2 - \mu_1)^T [\frac{\Sigma_1 + \Sigma_2}{2}]^{-1}(\mu_2 - \mu_1) + \frac{1}{2} ln \frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1|\,|\Sigma_2|}} \tag{2.25}$$

Wang *et al.* [84] apply an arrow marking detection and recognition algorithm based on improved Haar wavelet descriptors. An improved Haar wavelet feature extraction approach is utilized by them to describe the feature of arrow markings. In order to guarantee generalization performance, the F-score method is used for feature reduction in their research.

Kheyrollahi *et al.* [77] extract a set of connected contours by the backtracking approach then used Douglas-Peuker to derive a closed polygon shape to represent the contour in a simplified layer.

Wu *et al.* [101] computes the histogram of oriented gradients (HOG) descriptor for each point of interest (POI). The HOG descriptor consists of a 128 dimensional feature vector computed using certain optimal scales and orientations of the image patch around a POI. Considering the speed requirement, they extract the HOG feature vectors at 3 fixed scales and 1 fixed orientation. For each scale, a 128 dimensional feature is extracted for each POI. By concatenating the features extracted at different scales, they obtain the final 384-dimensional feature vector for each POI. For all the template images, the ROI containing the road signs is obtained from the labeled ground truth. After rectification, POIs are detected within the labeled ROIs of all template images. Then feature vectors are extracted for all the POIs. All of the feature vectors and the coordinates of the corresponding POIs are stored as the template pool.

Following, Wu *et al.* [10] propose a method towards reliable visual localization using traffic signs painted on the road such as arrows, pedestrian crossings, and speed limits by a stereo camera system in 2013. MSER detection, FAST corner detection, and HOG descriptor computation are performed on each test image. Their method automatically recognizes road markings and uses features detected within them to compute the location of the vehicle.

Guan *et al.* [115] uses Otsu's method which suggests the gray value of 60 as the optimal threshold to separate the foreground (road markings) from the background. After that they use a horizontally linear shaped structuring element to dilate and erode the road markings. The linear structure with length l and direction h is denoted by $K_{line\theta}(l)$. The direction θ is determined by the vehicle trajectory data, that is, the direction in which the vehicle is moving. A range of length l from 3 to 11 is used to determine $K_{line\theta}(l)$ for the road markings.

Li *et al.* [90] extracts the distance between the center and the boundary of each considered potential road marking, at regular angular steps then segment them, into a feature vector.

Liu *et al.* [88] adopts Haar-like features to quickly eliminate non-marking candidate regions. In the fine recognition step, the BW-HOG feature is designed to recognize the types of markings.

Suhr *et al.* [116] proposes a method which efficiently restricts the search area based on the lane detection results. The detector generates multiple candidates using a top-hat filter and projection histogram, and then classifies their types using a histogram of oriented gradient (HOG).

Hazelhoffa *et al.* [106] used the histogram of oriented gradients (HOG) technique and extended the original approach with the use of color gradient information, in order to exploit the color transitions, both within the signs and between the sign and surroundings in their research.

Chen *et al.* [117] used the binarized normed gradient (BING) method for road marking detection in their experiments.

Poggenhans *et al.* [89] presents an approach to detect, classify and approximate a great variety of road markings using a stereoscopic camera system which is independent of orientation, position or the exact shape. They use a histogram of the marking width as the main part of the feature vector for line-shaped markings, and Optical Character Recognition (OCR) for characters.

Position based Descriptor

Hara *et al.* [118] presents a method to detect road markings. The road area on the top-view image is estimated approximately using the flood-fill algorithm. In the algorithm, starting points called "seeds" $SD_{(x,y)}$ are given by GPS data, and then individual pixels around the seed $NearSD_{(x,y)}$ are selected for comparison. Pixels that are determined to have similar characteristics to seed, are set as a new seed and compared with pixels around the new seed. These are recursively processed, expanding the area of similar pixels and continuing until similar pixels cannot be found. For this road area detection, the system assigns seeds once in 50 points, firstly tracking the trajectory. These seeds from the trajectory are always placed on the road area and the road area is delineated by searching the similar gray pixels around the seeds.

Paula *et al.* [119] presents a method for detection and recognition of road lane markings using an uncalibrated onboard camera. Lane boundaries are detected based on a linear parabolic model and then represented by a simple model related to the pavement, which is then explored to estimate pixels related to lane markings.

Seo *et al.* [9] applies intensity thresholding, keeping only pixels where the intensity values are greater than a predefined threshold (e.g., 10). A connected-component group is adopted to identify a set of lane-marking pixel blobs. For each lane-marking pixel blob, the eigenvalues and eigenvectors of the pixel coordinates dispersion matrix are computed to fit a line segment to the blob. The eigenvector, e_1, associated with the largest eigenvalue is used to represent the orientation of a line segment and its length.

2.3.2 Classifier

Delgado *et al.* [120] evaluates 179 classifiers arising from 17 families (including discriminant analysis, Bayesian, neural networks, support vector machines, decision trees, rule-based classifiers, boosting, bagging, stacking, random forests and other ensembles, generalized linear models, nearest neighbors, partial least squares and principal component regression, logistic and multinomial regression, multiple adaptive regression splines and other methods), implemented in Weka, R (with and without the caret package), C and MATLAB, including all the relevant classifiers available today. To make the comparison clear, the classifier used in road markings filed so far were organized following the same criteria in this section.

Generalized Linear Models

Generalized linear models were formulated as a way of unifying various statistical models, including linear regression, logistic regression and Poisson regression. The least squares method for maximum likelihood estimation remains popular and is the default method on many statistical tools.

Baglidassarian *et al.* [78] compares a road marking's chain representation (algebraic moment) with that of the prototypes to recognize an object. They use the algorithm of Wagner-Fisher which measures the likeness of two chains.

Frank [71] used the width of the road marking shapes to distinguish their class. For instance, shapes that are two small are ignored, shapes with varying width may be arrows; the only shapes they are focus on are linear road markings like continuous lines and dashed lines which have nearly constant widths.

Charbonnier *et al.* [110] analyzes the grouping of isolated quadrangles to discard nonlinear markings and to determine parameters (like standard dashes modulation or line width).

Lu *et al.* [6] applies the least-squares fit technique to match lane model with the feature pixels. The processing course is optimized by calling the lane detecting and tracking module circularly.

Vacek *et al.* [100] successfully use the rules in German. Road markings follow strict regulations in Germany. A width of 12 cm is used for small markings and signals normal lane boundaries whereas 25 cm wide markings which are used to indicate turning lanes or emergency lanes. The bounding box is scaled to the size of the templates and the sum-of-squared-differences (SSD) between template and extracted region is used for determining the correct type of arrow. A maximum of three lanes are visible at the same time at these intersections. They were processed off-line and thus the processing time was not evaluated.

Chira *et al.* [85] uses LabVIEW functions of geometric pattern matching, edge detection and calibration information combined with custom C++ modules, and produced a robust, real-timer interactive system. Two different types of geometric pattern matching technique were used to detect the arrows and the lane markings as their geometry is different. For the arrows they used feature based geometric pattern matching and for the lane markings they used edge based geometric pattern matching. They found that the number of correctly detected and classified objects is highly dependent on the quality of the road markings. If the road mark is worn out it won't be detected at all or it will be erroneously classified as road marking. The same happens when the object is entering or exiting the image frame, as it is not completely visible. Furthermore, the reliability of the results depends also on the template images used for pattern matching and on the features selected for them.

Hara *et al.* [118] uses shadow detection and run-length compensation for robust recognition. Direction contributively features with run-length compensation were also applied to detect the faded road marking. Run-length compensation is a method which is stable to extract the feature from the faded road marking. Hence, by estimating the deterioration degree in the observation domain, a run-length value which is measured in non-faded state road markings approximately derives from the result of estimation. Rotating the marking templates to the same angle as the obtained road direction, template matching is performed with the crossing caution template, the 30km/h speed limit template, and the intensity image of HSI color model. The sum of absolute differences (SAD) based calculating method is selected for template matching.

Chang *et al.* [94] proposes the edge-pair scanning method to verify the edges which belong to lane-marks by using the relationship of adjacent edges of lane-marks and the width between these two edges. After eliminating the noise edges, they apply Hough Transform to fit the lane-marks as straight line models.

He *et al.* [87] proposes a weighted edit distance strategy and assigns different deviations with different weights to measure the similarity between the detected junction string and the ground truth junction string. Their framework is robust enough to detect deviations in arrow road marking but is sensitive to non-arrow road markings' deviations. Another deep learning framework (Boosting+Convolutional Deep Neural Network (CDNN)) is also implemented for comparison. In sum(Shown in Fig. 2.15), their experimental results shows the framework of Angle Coding+Topology Coding (AT) and Refined Angle Coding+Topology Coding (RAT) can maximally keep the high detection rate, while CDNN is superior in eroded arrow road markings' detection.

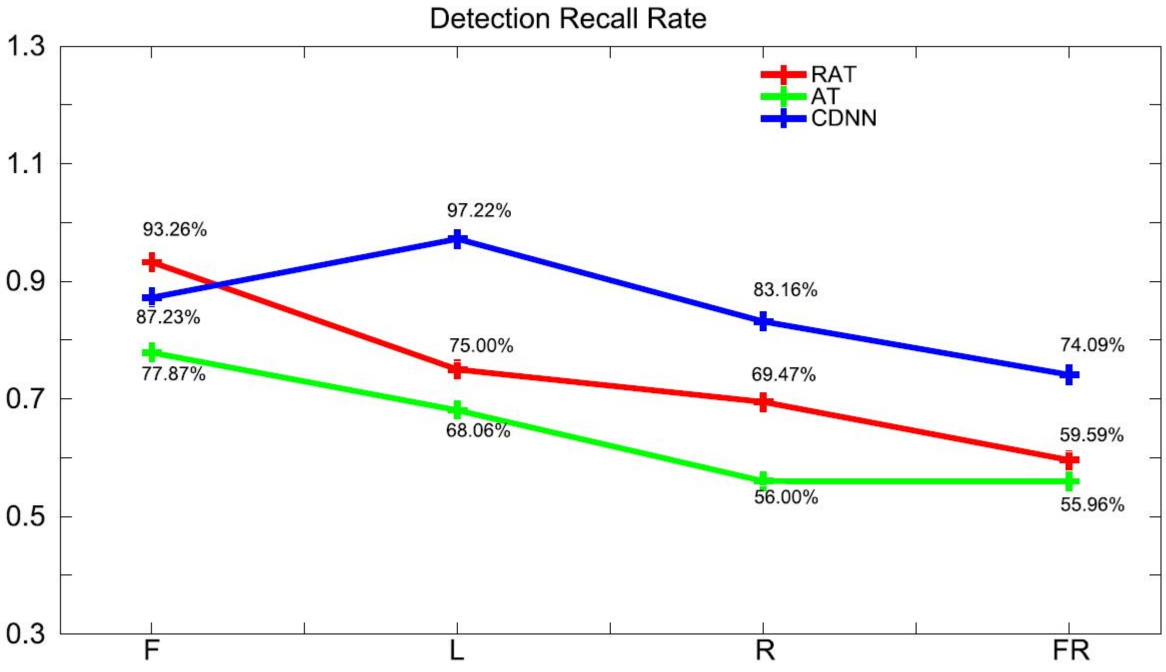


Figure 2.15: The overall detection recall rate for four types of arrow road markings.

Hernandez *et al.* [105] proposes a distance clustering analysis in order to determine the surface course. They also implement the Random Sample Consensus (RANSAC) line fitting method for removing the noise points around the road lane area. Finally, automatic peak detection was used in their work to perform lane marking detection on road surfaces.

Rule based Method

Rule-based systems are the domain-specific expert systems which utilize rules to make deductions or choices. As a set of methods widely used in artificial intelligence applications and research, they store and manipulate knowledge to interpret information in a useful way.

Enkelmann *et al.* [121] present a ROMA-System (Road Markings Analysis). In which, chosen segment s_2 has to pass the following tests: it must be parallel to segment s_1; and the gray value gradient of s_2 must point in the direction opposite to the gray value gradient of segment s_l. From those pairs of parallel line segments, the width and length of a rectangle in the scene is calculated using calibration information of the camera with respect to the planar road surface. This rectangle will be accepted as a road marking if its length is greater than its width. Once the list of single road markings is determined, a subsequent grouping of single road markings is performed to extract marking lines. It is assumed that the single road markings belonging to a marking line have similar width and length values.

Chira *et al.* [85] use LabVIEWs Find Edge function to detect the continuous lane. They set line fit options, by applying Hough Edge Rake, to detect the straight edges among the number of lines.

Li *et al.* [81] describes a method using geometrical analysis of the contours to compute the summary characteristics and match them to templates. A bounding box is introduced to summarize their characteristics, and this box can bound contours with the minimal rectangle. A filter approach is then performed by judging the center point, width and height of the bounding box, which can remove the undesirable candidates. Then candidate contours are compared with the template by means of template matching.

Bayesian Approaches

The Bayesian approach is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available. It derives the posterior probability as a consequence of two antecedents, a prior probability and a "likelihood function" derived from a statistical model for the observed data.

Kreucher *et al.* [122] introduces a Bayesian algorithm called LANA (lane-finding in another domain) for detecting lane markers in images acquired from a forward-looking vehicle-mounted camera. The method is based on a set of frequency domain features. A simple one-dimensional edge detection is followed by a least median squares technique for determining the curvature and orientation of the road. Lanes tracking from frame-to-frame are utilized in an existing model of the road geometry. It follows:

- A given image is broken up into 8*8 pixel blocks.
- For each block, a frequency-domain based feature vector is computed.

This feature vector reflects the amount of "diagonally dominant edge energy" that is contained in that 8*8 block. At any given node, the image's intensity profile is used to determine whether or not an edge is present at that node. This is accomplished by using a multi-resolution Fourier transform (MFT). Their paper finds the frequency domain to be a convenient vehicle to discriminate between edges that are diagonally dominant and those that are randomly oriented.

Nearest Neighbour Methods

Nearest neighbour methods, also known as proximity search or Nearest neighbour search (NNS), are an optimization problem for finding closest (or most similar) points. There are numerous variants of the NNS problem and the two most well-known are the k-nearest neighbour search and the ε-approximate nearest neighbour search.

Rebut *et al.* [112] carries out arrow recognition by computing the similarity between objects and the arrow database. To compare an object and a model characterized by their Fourier descriptors, respectively, a Euclidian distance between the two coefficients vectors is computed as a similarity measurement. In addition, they use a KNN classifier to improve the performance.

Foucher *et al.* [86] recognizes the markings based on the comparison with a single pattern or with repetitive rectangular patterns. The recognition of crosswalks and arrows are processed independently. The candidate image, which corresponds to the bounding box of the resulting connected component and the model image, is normalized to the size 120*120. The models are learned with their rotated position within $(-15^{\circ}, +15^{\circ})$ related to major axis in the image plane. The similarity criterion firstly considers the binary Hamming distance between the model image and the candidate image. If the Euclidean distance is lower than a threshold σ_h, the connected component is recognized with the type of closest model. Otherwise, a second similarity criterion is computed between the histograms of the closest model and the histograms of the candidate.

Liu *et al.* [82] used template matching, specifically the Cosine correlation coefficient for recognition of markings. The connected domain analysis makes the method suitable for situations even there are no lane lines.

Partial least squares and principal component regression

Partial least squares regression (PLS regression) is a statistical method that finds a linear regression model by projecting the predicted variables and the observable variables to a new space. Principal component regression (PCR) is a regression analysis technique that considers regressing the outcome on a set of covariates based on a standard linear regression model, but uses principal component analysis (PCA) for estimating the unknown regression coefficients in the model. PLSR and PCR are both methods to model a response variable when there are a large number of predictor variables, and those predictors are highly correlated or even collinear.

Maier *et al.* [109] presents a general geometric approach using curve-based prototype fitting. Connected components are extracted within the ROI using segmentation threshold which gained by N-level-fitting techniques. In order to compare a candidate of extracted contour, with the particular prototypes for classification, they use a generalization of the Iterated Closest Point (ICP) algorithm. Their approach encodes the prototypes as a curve which allows a fast computation of point to curve distances and which is composed of a preferably low number of segments. The reconstructed contour point list of the extracted connected component is passed to the prototype fitting algorithm after some plausibility checks. As the prototype fitting minimizes the least squares error but not the Hausdorff distance, the latter can even increase during the iterations. Nevertheless, the arrow pointing ahead is also the best fitting arrow according to the Hausdorff metric, which turns the overall classification to the expected result.

Discriminant Analysis

Discriminant function analysis is a statistical analysis to predict a categorical dependent variable by one or more continuous or binary independent variables. It is used when groups are known a priori.

Li *et al.* [12] selects moment features and an improved minimal-error-rate classifier to recognize different lane markers and other road markers. The lengths and slopes of lane markers are also rapidly calculated using the moment features. They choose five kinds of frequently appearing road markers as the objects of detection and recognition, namely solid line, dash line, merge line, forward arrow, and forward-rightward arrow. Over 5000 samples of the five kinds of road marker are chosne in several image sequences taken in daytime, sunny or cloudy weather. Half of the sample set was used to train the classifier and the other half was used to test it. Dozens of non-marker sample were also collected to test the classifier. Fig. 2.16 shows the univariate probability distribution of the selected features.

Wu *et al.* [101] computes their matching algorithm in two steps. First, they find putative matching pairs of POIs based on their feature vectors, and secondly, they refine the result through a structural matching algorithm that matches the 2D geometry of the POIs within the road marking. For a subset of the matching edges, the POIs involved in the test image and the template form two different shapes. Therefore, a matching cost between the two shapes is calculated to decide if they have the same geometric structure. The videos of the road scenes were recorded at 800x600 resolution. They collected 29 videos, which contain a total of 28614 frames. Among those, 22 images are selected as the templates, which contain ten different types of road signs including speed limit signs "35" and "40", "left turn", "right turn" and "forward arrows", and "bike", "stop", "ped", "Xing" and "rail" signs. Their true positive rate is about 90.1% and false positive rate is about 0.9%, indicating that false positive detections occur only very rarely.

Ensemble Method

Ensemble methods use multiple algorithms to obtain better predictive performance than could be obtained from any of the constituent algorithms.

Paula *et al.* [119] applied a cascade of binary classifiers to distinguish five types of markings: dashed, dashed-solid, solid-dashed, single-solid and double-solid.

He *et al.* [87] proposes a weighted edit distance strategy and assigns different deviations with different weights to detect arrow road markings. They collected three freeway datasets: clean/dirty arrow road marking images (300 images respectively), a video dataset (arrow road marking and non-road marking images (670 images)). Specifically, a Weighted

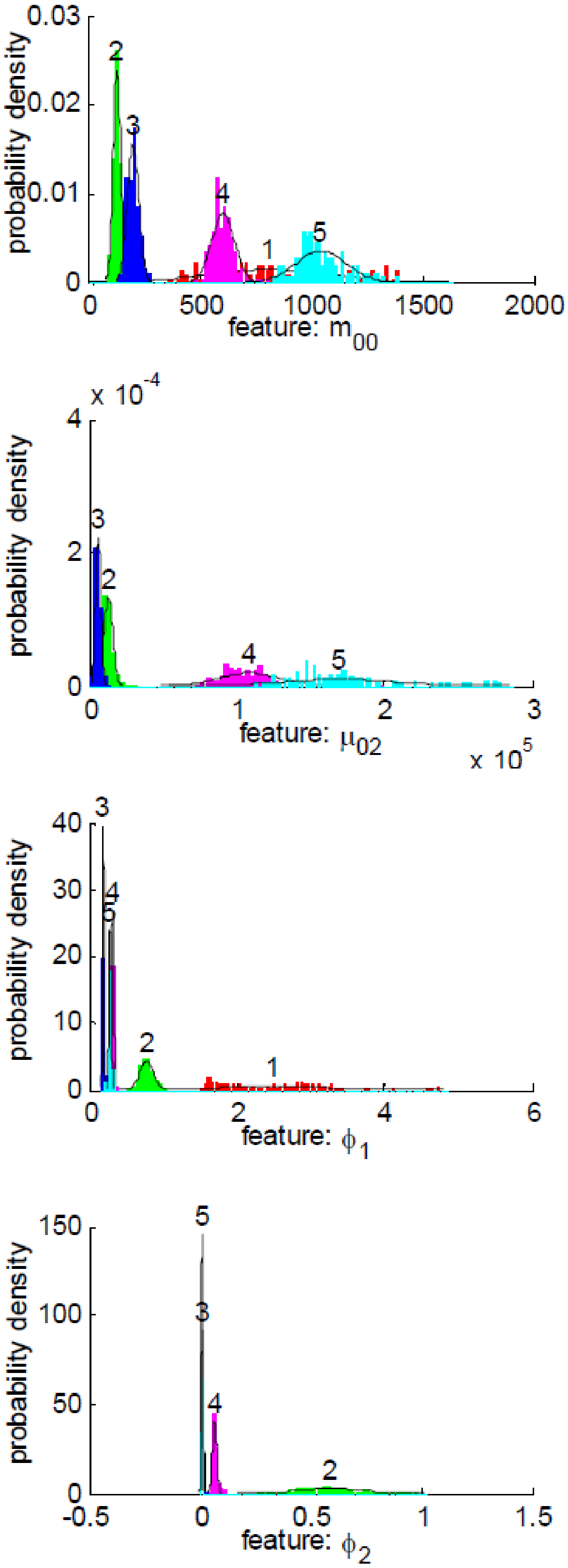


Figure 2.16: Univariate probability distribution of selected features. [12]

Edit Distance strategy is used to measure the similarity between a coding string and six ground truth string templates. The match which shares the shortest edit distance (also must be within a predefined threshold) is chosen as the recognition result. An arrow road marking's deformation, abrasion and other negative impacts are transferred into L-junction's angle difference and position difference but the framework is robust enough to tolerate them; as a L-junction within a range is encoded as the same code, the weighted edit distance strategy can also handle this problem by assigning different impacts with different penalty weight.

Support Vector Machine

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible.

Noda *et al.* [13] generates numerous learning images with the appearance variation in their research in order to recognize road markings in real environments accurately. They note that there are many factors that cause appearance changes of a road marking in an in-vehicle camera image, such as changes of the camera position and camera posture, or internal property of the camera. Fig. 2.17 shows examples of such appearance changes. Therefore, they classify these factors into twelve groups and constructed a generation model for each group.

- Shape deformation
- Resolution degradation
- Optical blur
- Motion blur
- Clipping error

Figure 2.17: Road markings with appearance changes by [13]

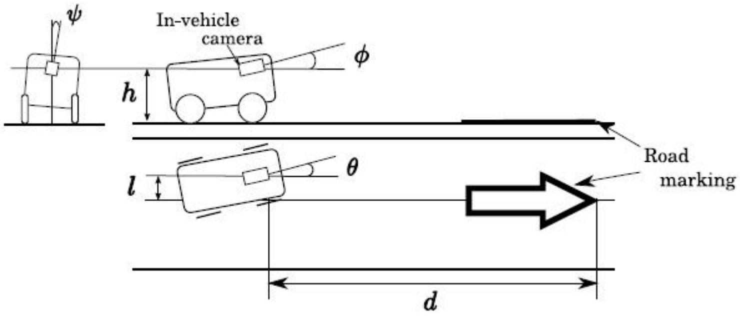
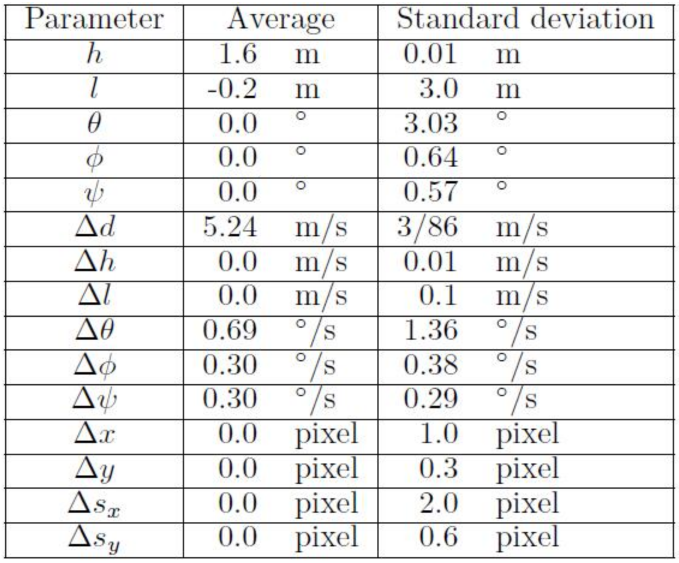


Figure 2.18: Geometrical parameters used by [13]

The generation models are used to generate road marking images for learning. As introduced by them, in Japan, the shape and size of road markings are standardized, so they prepared an image of the standard road marking image as the original image for generation. Then parameters are utilized to configure the system. (Shown in Fig. 2.18)

Following is an overview of their generation process. First, a road-plane image which includes the original road marking image was captured. Then, optical blur and motion blur are added to the image. Next, the image is projected to the road-plane. Finally, road marking images are clipped from the road-plane image with various clipping errors. (Shown in Fig. 2.19) Their experiment achieved an overall recognition rates of 95% among 1000 frames including markings.



Parameter	Average		Standard deviation	
h	1.6	m	0.01	m
l	-0.2	m	3.0	m
θ	0.0	°	3.03	°
ϕ	0.0	°	0.64	°
ψ	0.0	°	0.57	°
Δd	5.24	m/s	3/86	m/s
Δh	0.0	m/s	0.01	m/s
Δl	0.0	m/s	0.1	m/s
$\Delta \theta$	0.69	°/s	1.36	°/s
$\Delta \phi$	0.30	°/s	0.38	°/s
$\Delta \psi$	0.30	°/s	0.29	°/s
Δx	0.0	pixel	1.0	pixel
Δy	0.0	pixel	0.3	pixel
Δs_x	0.0	pixel	2.0	pixel
Δs_y	0.0	pixel	0.6	pixel

Figure 2.19: Generation parameters (normal distribution) in [13]

Wang *et al.* [84] applies an arrow markings detection and recognition algorithm based on multi-class SVM. In the phase of training in their paper, the Euclidean distance of every two classes from k class training samples is calculated, with small distances indicating two classes are more similar.

Qin *et al.* [83] use SVM for the classification, with classifiers trained independently for each type of markings. Geometric features are extracted by them from each contour. The selected features (Hu Moments, weights, Bounding Rectangle and approximate polygon) are all rotation invariant, and generally carry enough geometric information for the contour classification purposes. Their experiment result shows the classification performance of each classifier under 5-fold cross validation.

Paula *et al.* [119] describes choices for the four binary classifiers C_1, C_2, C_3 and C_4, as well as the feature vectors used for each classifier. For the classifier C_1, SVM is chosen and controlled by parameter $C > 0$. The experimental results indicate that the Radial-Based-Function (RBF) kernel presents the best results, which is defined by:

$$K_\gamma(x, y) = e^{-\gamma\|x-y\|^2} \tag{2.26}$$

where $\gamma > 0$ is the width parameter. They experimentally evaluated different values for both C and with the holdout cross-validation randomly selecting a fraction $g < 1$ for training model, and a fraction $1 - g$ for validation. They achieved cross validation accuracy as 100.0% and the estimated parameters were: $C = 1$ and $\gamma = 4$.

Schreiber *et al.* [14] detected symbols on roads (e.g. arrows, speed limits, bus lanes and other pictograms) using TESSERACT, an Optical Character Recognition (OCR) System trained by SVM. The obtained information can be used on one side to improve localization, on the other side to provide further information for planning or generation of planning maps. To prepare the OCR system, a font with the desired symbols was generated. The font consists in the case of all letters, symbols and arrows allowed on German roads, which is defined. To increase detection rate, two further fonts with the same characters slightly slanted are generated. Fig. 2.20 shows all fonts used for training. A second set of training data is generated with only symbols and arrows and the character "I". This allows a better classification if a single symbol is detected on the road, since usually letters and numbers occur in groups. With the character "I", a better differentiation between segments of dashed lines and straight arrows is possible.

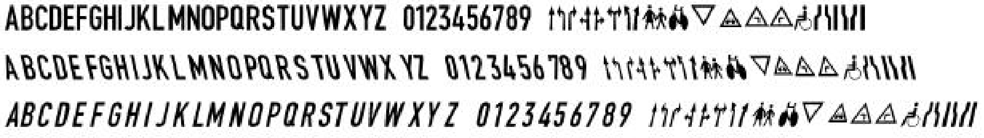


Figure 2.20: Fonts used for training of TESSERACT in [14]

Furthermore, markers for the watershed based segmentation are computed as sets of three markers (road/marking/road). A marker set consists of three pixel positions u_l, u_m, u_r in one image row v, where u_l is left of a road marking, u_m directly on the marking and u_r right of the marking. The following conditions must be fulfilled:

$$m_min < g(u_m, v) < m_max \tag{2.27}$$

$$r_min < g(u_l, v) < r_max \tag{2.28}$$

$$r_min < g(u_r, v) < r_max \tag{2.29}$$

$$g(u_m, v)g(u_l, v) > \delta g_min \tag{2.30}$$

$$g(u_m, v)g(u_r, v) > \delta g_min \tag{2.31}$$

$$G_med(u_l, v) < G_0 \tag{2.32}$$

$$G_med(u_r, v) < G_0 \tag{2.33}$$

In total they tested 398 symbols painted on the 104 km road, including 364 arrows, 12 speed limits and 9 ”BUS” symbols. With both, monoscopic and stereoscopic system about 80% of the symbols are detected.

Li *et al.* [90] classifies each segment using SVM. After translation, scale and rotation are normalized, the features are then extracted by calculating the distance d_i from the center to the boundary of each segment at certain angles. Angles from 0^o to 360^o are

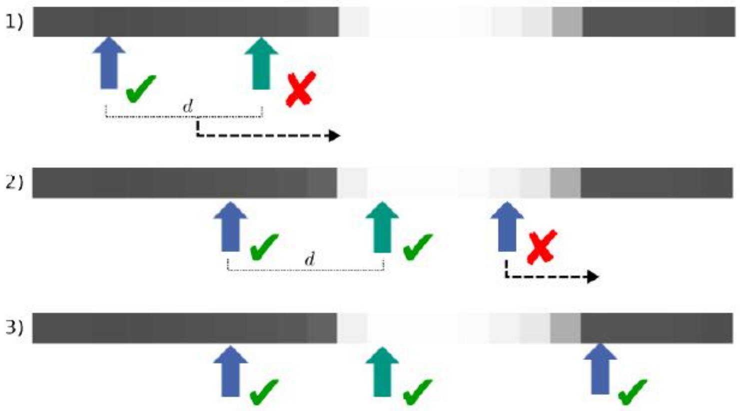


Figure 2.21: Each line is checked from left to right if a position is found where all conditions are fulfilled. By [14]

chosen with a step size of 30^o. The value of this step size was chosen empirically and based on numerous experiments. Each segment can then be described by the vector v = (d_1; d_2; :::; $d_1$2). To distinguish the road markings from other regions, a non-linear SVM with a radial basis function kernel is used then. Finally, their experiments show that the system is capable of recognizing 93%, 95% and 91% of striped line segments, blocks and arrows respectively, as well as 94% of the lane markings.

Neural Networks

Artificial neural networks (ANNs) are a family of models inspired by biological neural networks which are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks usually contains sets of adaptive weights, and is capable of approximating non-linear functions of their inputs.

Wendling *et al.* [15] solved a similar problem in Line Drawings area by introducing specific criteria. They found that an isoceles triangle has a unique angle bisector, passing by C, which splits it into two symmetric parts. This bisector is also the median of the rectangle in an arrow. With an assumption that the pattern is symmetric, they have shown that the definition of angular and theoretical signatures and the aggregation of geometric criteria using the Choquet integral can achieve a robust detection of arrows. (Shown in Fig. 2.22) They focus on a subpixel definition to decrease the aliasing resulting from Bresenhams algorithm in order to increase the accuracy of method.

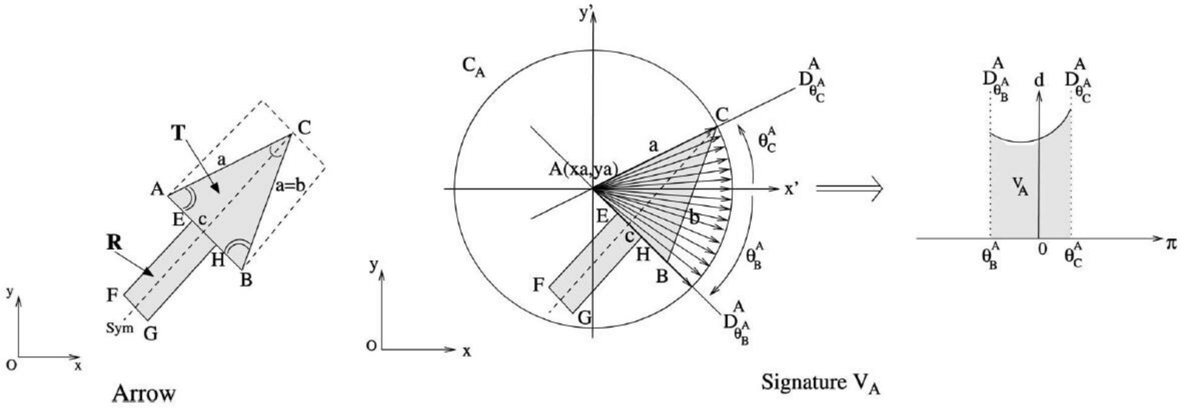


Figure 2.22: Discrete signature of an arrow By [15]

Kheyrollahi *et al.* [77] uses a trained single layer neural network classifier for individual glyph classification prior to glyphword/pattern construction via a simple post-processing matching methodology. After detection, objects are simplified to a contour representation that is passed to an artificial neural network (ANN) classifier for recognition. For an overall of 129 frames, a 85.2% recognition rate was achieved in their experiment.

He *et al.* [87] describes each road marking as a junction string and designed a machine learning framework by using Boosting+CDNN (Convolutional Deep Neural Network) with four million positive/negative training samples.

Liu *et al.* [88] applies prior knowledge of road makings to generate candidate road marking regions. Afterwards, a coarse-to-fine marking recognition method is presented. In the coarse recognition, an Adaboost classifier with Haar-like feature is adopted to fast eliminate non-marking candidates regions. In the fine recognition, an ELM classifier with BW-HOG feature is designed to recognize the types of markings. Finally, a spatial-temporal fusion method is introduced to further enhance the recognition accuracy and reliability of the system. Their training dataset contains 5000 samples of road markings of ten types, about

500 samples per type. The testing dataset contains 4000 samples of ten types, roughly 400 samples per type, and achieved a 97.84% detection rate of arrows.

Poggenhans *et al.* [89] achieved a classification result which is independent of orientation, position or the exact shape by using a histogram of the marking width as main part of the feature vector for line-shaped markings and Optical Character Recognition (OCR) for characters. Classification is done by an Artificial Neural Network (ANN). They have also evaluated the approach over a 10.5 km drive through an urban area. And the recognition rates from a realistic road scenario were at 80% overall, with arrow detection rate at 85% in a 2013 frames database including 105 arrows.

Chen *et al.* [117] present their approach for road marking detection and classification by using binarized normed gradient (BING) method and classification of PCA network (PCANet). Their road marking dataset contains 1,443 road images. Among them, 60% images are randomly choose for training and the remaining 40% images are used for testing. Upon training, the system can detect 9 classes of road markings with an accuracy better than 96.8%.

2.4 Tracking Stage

To enable the following of lane marking over time, a tracking stage is usually incorporated.

2.4.1 Kalman Filter

The most common trackers used in assistant driving systems are Kalman filters and Particle filters.

Lu [123] used Kalman filters to track the endpoint of road lines detected in front of ego vehicle and performed a Point Clustering Comparison of related four points. To normalize the position used for tracking, he selected an appropriate y-value and cut the detected line with the same vertical position. According to the results of his experiment, the process of lane detection and tracking (LDT) system as proposed in the comprehensive module are considered as a Gaussian stochastic process.

Kalman Model

According to [124], the inner connection between the state of a model at time k and the state of $(k-1)$ can be described as follow:

$$x_k = F_k x_{k-1} + B_k u_k + W_k \tag{2.34}$$

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:

$$W_k \sim N(0, Q_k) \tag{2.35}$$

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

$$z_k = H_k x_{k-1} + v_k \tag{2.36}$$

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

$$v_k \sim N(0, R_k) \tag{2.37}$$

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:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{2.38}$$

and the priori estimation error covariance:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{2.39}$$

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:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{2.40}$$

Innovation covariance:

$$\mathbf{S}_k = H_k P_{k|k-1} H_k^T + R_k \tag{2.41}$$

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:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{2.42}$$

Posteriori (after being updated) error estimation covariance:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{2.43}$$

Posteriori (after being updated) state estimation:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{2.44}$$

It is pointed out in [124] that, the formula for the updated estimation and error covariance above are valid only under the condition of the optimal Kalman gain (Equation 2.42). Thanks to the inherent recursive ability, Kalman filter is able to run in real-time by taking advantage of measurement and estimation results.

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 $Pt_1(X_{Pt_1}, Y_{Pt_1})$. 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 2.34 will not be taken into account.

Recalling Equation 2.34, where x_k is the state vector, F_k is the state transition matrix.

We define the state vector as:

$$x_k = [X_{Pt_0} Y_{Pt_0} X_{Pt_1} Y_{Pt_1} X'_{Pt_0} Y'_{Pt_0} X'_{Pt_1} Y'_{Pt_1}]^T \tag{2.45}$$

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:

$$z_k = [X_{Pt_0} Y_{Pt_0} X_{Pt_1} Y_{Pt_1}]^T \tag{2.46}$$

2.4.2 Particle Filter

Particle filter is another reliable option for lane tracking.

Vacek *et al.* [100] successfully uses a particle filter to track a single lane. In his work, each particle represents one sample and the evaluation function determines the probability having the measurements given this particular sample. Each particle represents a particular parameter set of the lane model M_i, described by the four introduced parameters:

$$M_i = \left\{x_0^i, \psi^i, \omega^i, \phi^i\right\}. \tag{2.47}$$

The a-posteriori probability for each particle is calculated by evaluating different cues with each cue representing a specific hint about the observed scene. The cues used in this work are:

- Lane marker cue (LM), estimating the probability of having lane markings under the projected model.
- Road edge cue (RE), estimating the probability of having edge elements at the borders of the lane.
- Road color cue (RC), estimating the probability of having an area of road color under the projected area.
- Non-road color cue (NRC), estimating the probability of having an area of non-road color outside the projected area.
- Elastic lane cue (EL), evaluating the expected offset of the lane.
- Lane width cue (LW), evaluating the expected width of the lane.

Each cue gives a value between 0.0 and 1.0 and the overall rate of a particle $p(M^i)$ evaluates to:

$$p(M^i) = p_{LM}(M^i) \cdot p_{RE}(M^i) \cdot p_{RC}(M^i) \cdot p_{NRC}(M^i) \cdot p_{EL}(M^i) \cdot p_{LW}(M^i) \tag{2.48}$$

The resulting estimation $p(\hat{M})$ is then given by the weighted sum of all particles. This value is compared with two thresholds in order to decide, if a lane was really tracked.

Chapter 3

System Architecture

As explained in Chapter 1, a real-time system is proposed in this thesis aimed at lane marking detection and recognition. Compared with previous work listed in Chapter 2, the proposed method shares a similar but notably different framework. A discussion of characteristics of road line and road markings according to the construction manual of civil engineering was introduced in the beginning of this chapter. Then an MSER-based high-efficiency integrated solution was denoted accordingly. The main idea of the proposed method includes a creative descriptor of the 4 extreme points set and a rule-based cascaded classifier, which are designed to classified detected objects, because this combination makes the best use of the characteristics of road. The whole processing method, including preprocessing by applying a rectangular ROI in the bottom of the image and mapping 3-colour channel image to grayscale images, detects the lines as well as arrows and HOVs using MSER based on grayscale images, calculates and collects 4-points of each contour of interest, then classifies the residual blobs by rules. The process of the proposed algorithm is organized as in Fig. 3.1):

3.1 Preprocessing

Specifically, in the preprocessing step, ROI was chosen to decrease the computation cost while maintaining most of the useful information contained in the raw data. Unlike most of

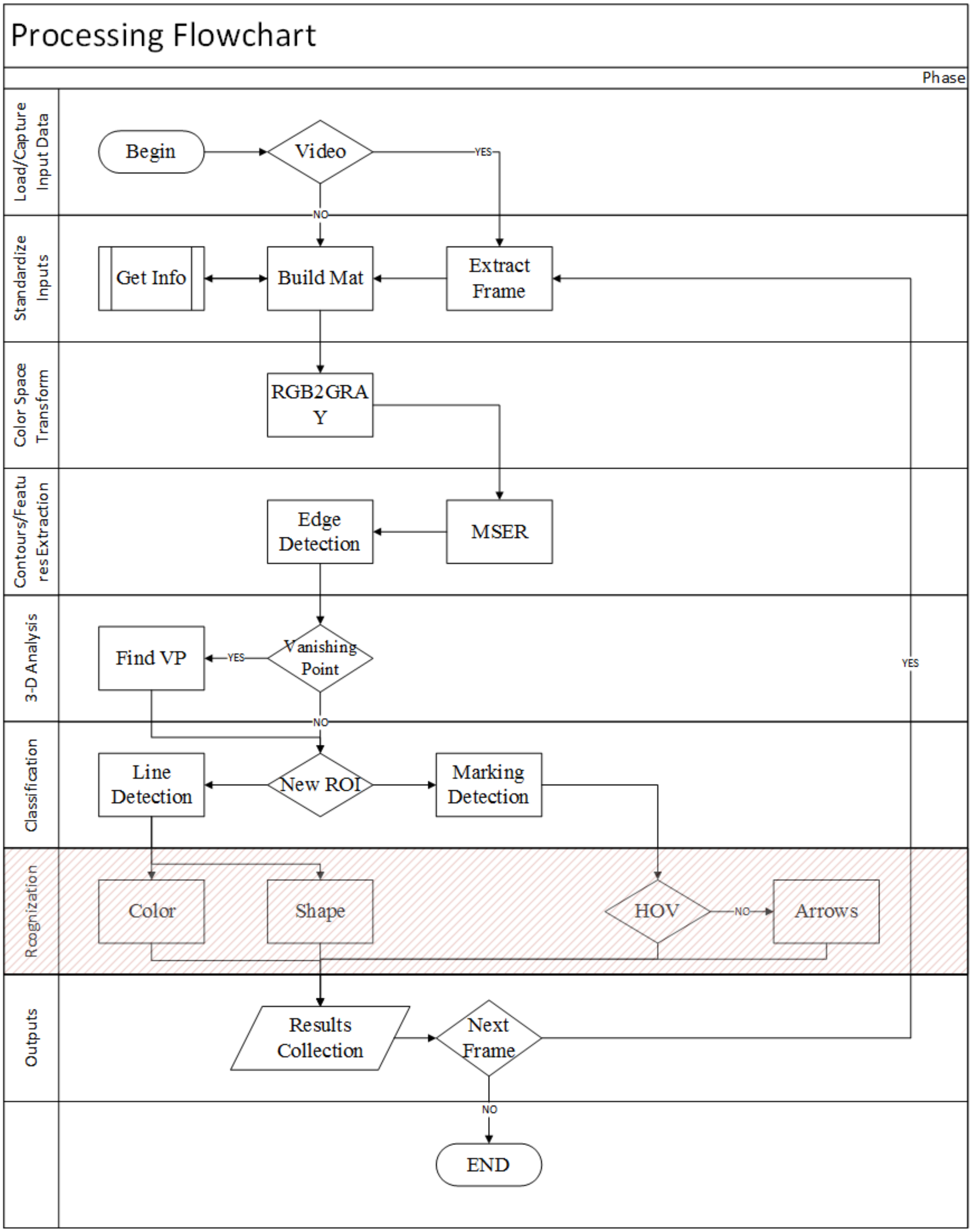


Figure 3.1: Flowchart: Recognition part is the main contribution.

the methods utilized by the references, the smoothing filter has not been used because the proposed detection method is not sensitive to the noise point. Similarly, with an assumption that a moving car will maintain a safe distance from surrounding vehicles, binarization becomes unnecessary since segmentation between foreground and background can be done by approximate estimation based on distance. Inverse Perspective Transform(IPT) was widely used in references cited before. Since the classifier used here (the relationship between 4 extreme points) will not be affected by distortion, IPT is not implemented in this thesis. Vanishing point tracking, as a useful supplement, has been introduced in the enhanced analysis section, the last section of this chapter.

3.1.1 Region of Interest

As introduced in the previous chapter, different strategies were implemented by researchers according to the methods they used. For instance, a rectangular region in front of the vehicle was widely used in this part because of the commonly agreed-upon assumption that this part contains more useful information. On the other hand, trapezoid areas are often used as ROI when IPT was the selected method. Otherwise, the car hood and windscreen wipers are usually important areas of concern.

In our experiments, we installed the camera close to the front windshield of the vehicle. As a result, the lower part of the road captured by the camera was covered by the hood. To reduce the distraction from ego vehicle and sky, we set our ROI at (0,240) to (640,420) (opposite corner of the rectangle, shown in Fig. 3.2), with respect to the original resolution at 640×480.

3.1.2 MSER Extraction

Researchers have employed various methods to attempt to extract the edge information of the ROI. Additionally, they tend to use smoothing methods (Gaussian Filter 2.1.2, Median Filters 2.1.2, etc.) before edge detection to remove unwanted information, blur the

Figure 3.2: ROI of proposed method

difference inside the region and keep regions with stable luminance. This can result in the omission of details, especially in the edges of regions where luminance changes rapidly. To balance between keeping desired details and removing extraneous information, we use Maximally Stable Extremal Region (MSER) for the preprocessing stage. MSER was proposed in [125] in order to Finding reliable correspondences in two images of a Scene and has rarely been used in lane marking detection. Compared to edge-based segmentation, the biggest advantage of MSER is that it only recognizes the stable extremal region (e.g., lane markings, traffic signs or dark parts of cars), and successfully ignores unpredicted undesirable regions (e.g., potholes and obstacles on the road).

In [126], Sun et al. proposed a method which consists of describing MSER patches using SIFT-based descriptors, followed by a graphical model to localize lane markings. Wu *et al.* [10] used MSER to detect traffic signs painted on the road such as arrows, pedestrian crossings, and speed limits by a stereo camera system. In contrast to [126], which involves an unsupervised learning algorithm and off-line training, our proposed system directly takes advantage of MSER blobs to extract the features of lane markings (lines and pictograms). In addition, our proposed method uses MSER results to set the ROI for detection stage, allows the improvement of MSER and Hough transform results, and increases the detection rate of lane markings.

For a gray image, which can be described as a mapping: $(x, y) \in Z^2 \rightarrow L$, where Z^2 represents a set of pixels with coordinates (x, y), and L represents a set of luminance of pixels ranges that vary from 0 to 255. The term region we use here represents a contiguous subset S of the space Z^2 (specifically for 4-neighbourhoods) which satisfies:

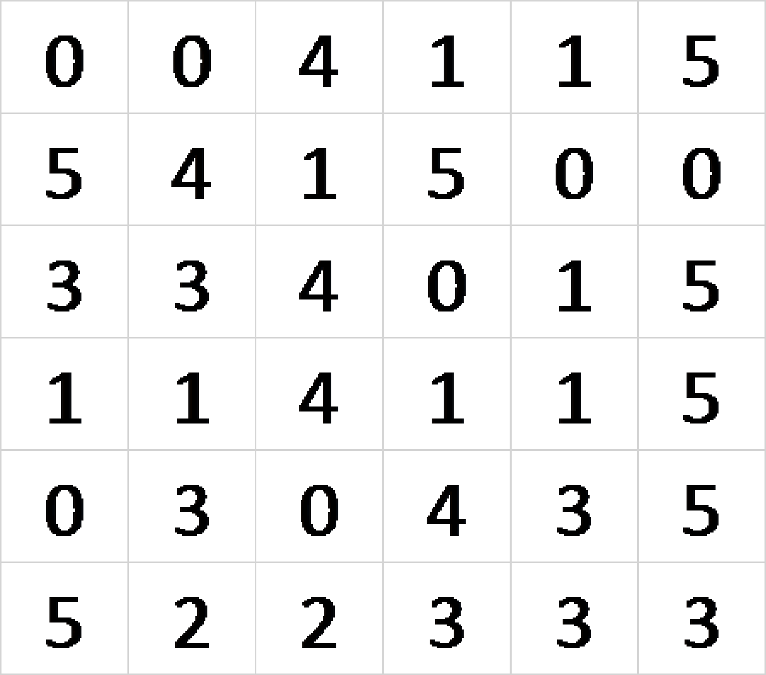
$$\begin{aligned}
&\forall p, q \in S,\ p, q \neq \varnothing; \\
&\exists Series\,\{\ p, a_1, a_2, a_3, \ldots, a_{i-1}, a_i, q\ \}\,, \\
&where\ a_1, a_2, a_3, \ldots, a_{i-1}, a_i \in S. \\
&s.t. \begin{cases} |p - a_1| = 1; \\ |a_i - q| = 1; \\ \sum_{j=2}^{i} |a_j - a_{j-1}| = i - 1; \end{cases}
\end{aligned} \tag{3.1}$$

A region S is called an extremal region when an arbitrary element in the region satisfies the mapping rule $S \rightarrow m \leq l$; where m, $l \in L$, m represents the mapped value in L of an arbitrary element in S, and l is a pre-defined threshold which ranges in [0, 255]. A stable extremal region is an extremal region S that does not change significantly while varies levels applied.

Let: $R(S_l) = S_l, S_{l+1}, S_{l+2}, \ldots, S_{l+\Delta-1}, S_{l+\Delta}$ be a branch of trees rooted in S_l and satisfied: $S_l \subset S_{l+1} \subset S_{l+2} \subset \ldots \subset S_{l+\Delta-1} \subset S_{l+\Delta}$. In order to measure the stability of different extremal region, we use the following equation (as proposed in [125]):

$$q(l) = \frac{card(S_{l+\Delta} - S_l)}{card(S_l)} \tag{3.2}$$

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 when $q(l)$ of S_l is in the lower 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.



0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

Figure 3.3: 6×6 Gray Image

As Fig. 3.3 showed, we have a gray image which contains 6×6=36 pixels. Each pixel has its own gray level (0-5). The smaller its value is, the darker its colour appears. This table can also be represented using a gradient map. Fig. 3.4 and Fig. 3.5 are the side view and top view of such a gradient map respectively.

In Fig. 3.6, the process of iterations of MSER was shown step by step. First of all, only the darkest area (value=0) was picked up and marked into 5 connected regions (4-neighbourhood rules applied).

$$A = \{0, 0\}, B = \{0, 0\}, C = \{0\}, D = \{0\}, E = \{0\} \tag{3.3}$$

Then, taking value 1 into account, the regions have changed but the number of regions remains the same.

$$A = \{0, 0\}, F = \{1, 1, 0, 0, 0, 1, 1, 1\}, G = \{1\}, H = \{1, 1, 0\}, E = \{0\} \tag{3.4}$$

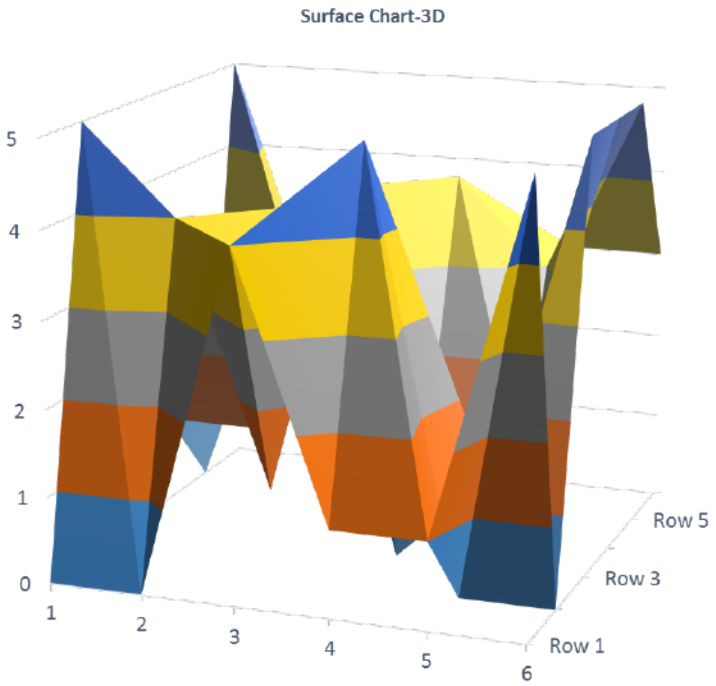


Figure 3.4: Gray level in 3-D space

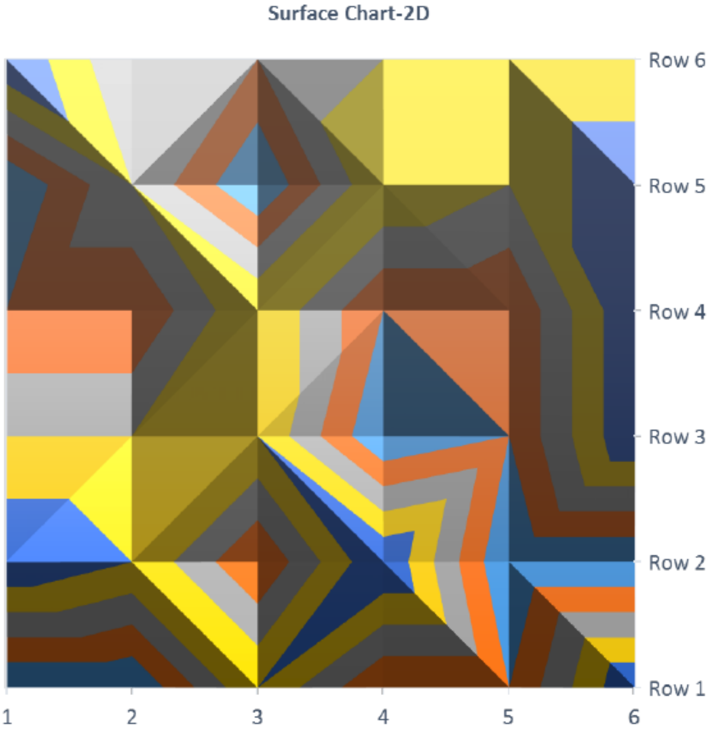
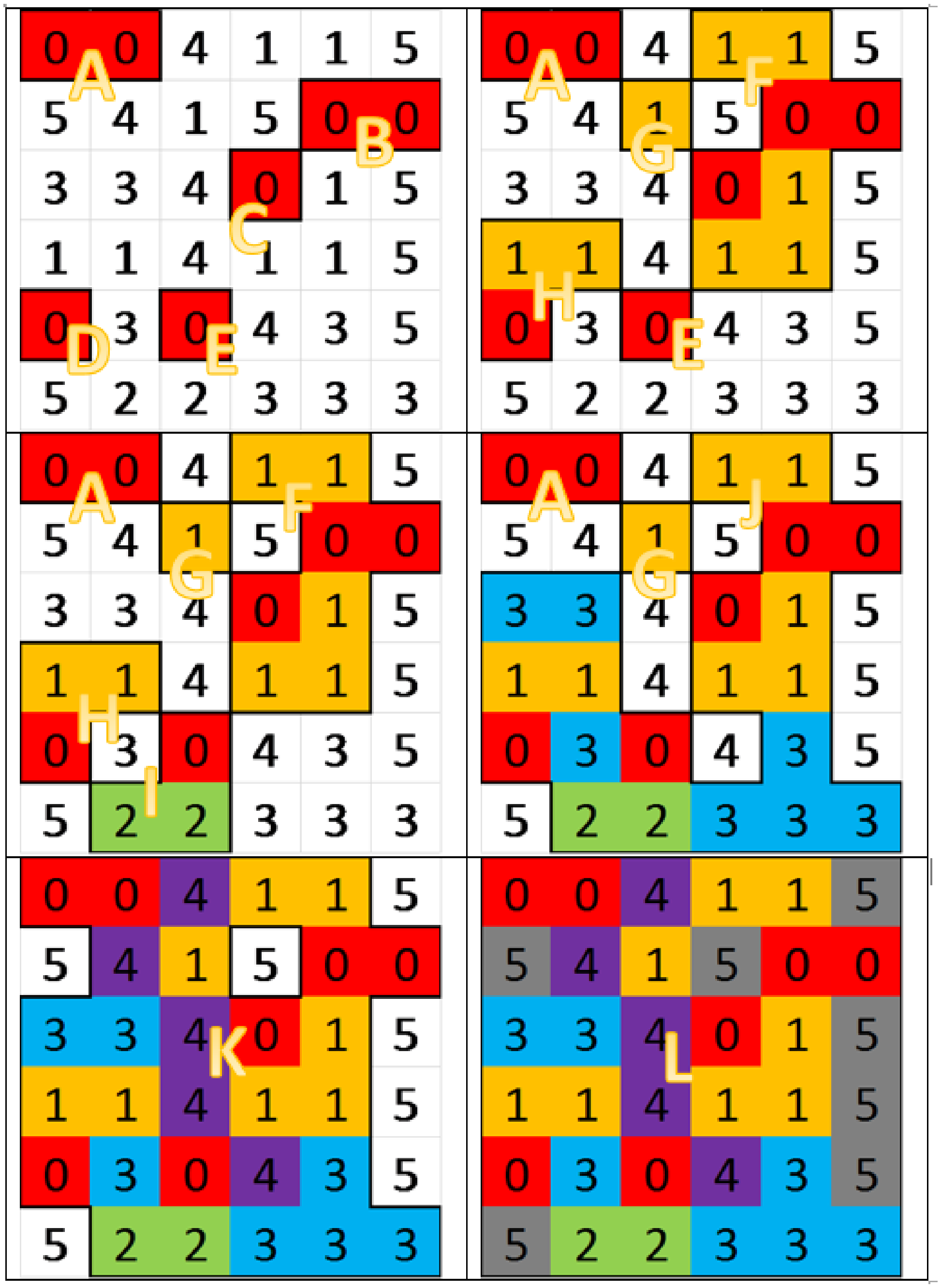


Figure 3.5: Topview of gray level in 3-D space



0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

0	0	4	1	1	5
5	4	1	5	0	0
3	3	4	0	1	5
1	1	4	1	1	5
0	3	0	4	3	5
5	2	2	3	3	3

Figure 3.6: Iterations of MSER

In the third step, value 2 has been added, however the number of connected regions remains unchanged.

$$A = \{0,0\}, F = \{1,1,0,0,0,1,1,1\}, G = \{1\}, H = \{1,1,0\}, I = \{0,2,2\} \tag{3.5}$$

Then, value 3 was included and number of regions changes to 3.

$$A = \{0,0\}, J = \{1,1,0,0,0,1,1,1,3,2,2,3,3,3,0,3,0,1,1,3,3\}, G = \{1\} \tag{3.6}$$

In step 5 and step 6, the components selected are all connected therefore the number of regions is 1. During the iterations, it was clearly shown that most of the regions obtained through the entire process have nested relationships with one or more regions found in previous steps.

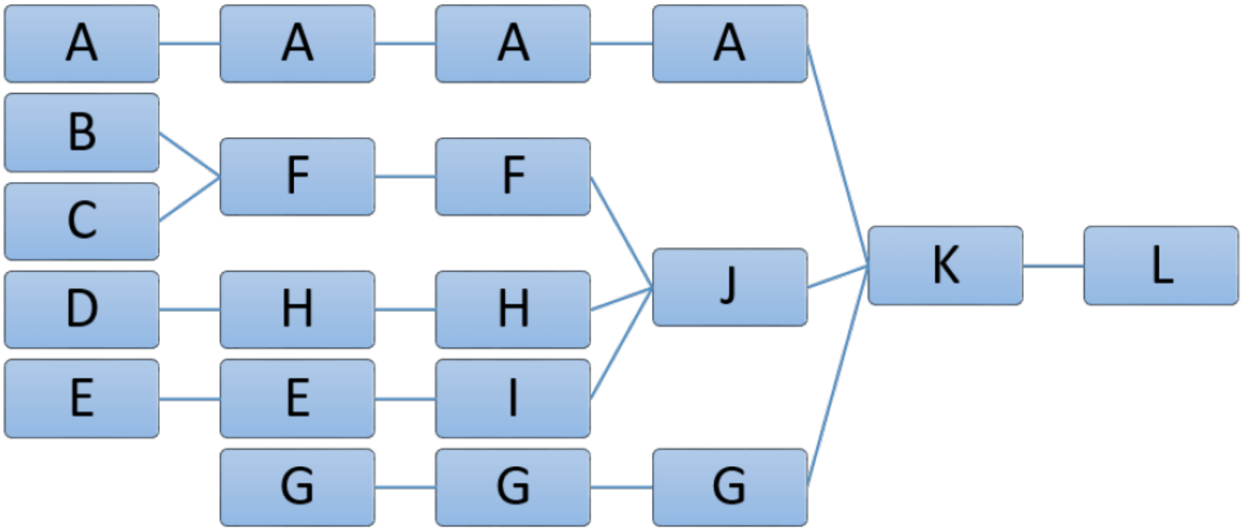


Figure 3.7: Nest relation between regions

By calculating the local minimum change of connected region set's cardinality according to formula 3.2, a threshold Δ can be obtained 3.1, which provides a critical parameter to detect the region that has minor differences of inner luminance. Taking 3 pixels as a threshold of minimum area, any region obtained which is smaller or equal to 3 pixels has been neglected. The residual regions under consideration are F, H, I, J, K, L.

Table 3.1: Path and Relation in MSER

Path/q(i)	q(1)	q(2)	q(3)	q(4)	q(5)
A-K-L	0.00	0.00	0.00	13.50	0.24
B-F-J-K-L	3.00	0.00	1.63	0.38	0.24
C-F-J-K-L	7.00	0.00	1.63	0.38	0.24
D-H-J-K-L	2.00	0.00	6.00	0.38	0.24
E-I-J-K-L	0.00	2.00	6.00	0.38	0.24
G-K-L	N/A	0.00	0.00	28.00	0.24

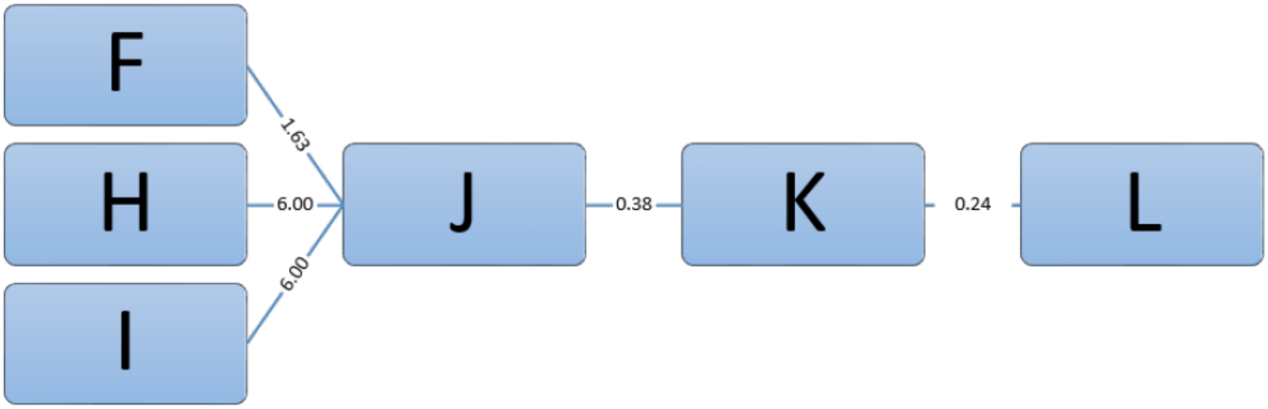


Figure 3.8: Selected path after filtering by threshold

$$Min\{Sequence[F, J, K, L]\} = q(K) = 0.24 \tag{3.7}$$

$$Min\{Sequence[H, J, K, L]\} = q(K) = 0.24 \tag{3.8}$$

$$Min\{Sequence[I, J, K, L]\} = q(K) = 0.24 \tag{3.9}$$

As a result, K is the MSER in this case.

3.1.3 Refinement of MSER results

It must be noted that MSER is however more computationally expensive than edge-based detectors. Moreover, sometimes, MSER-blobs contain unwanted details as well as desired pixels. This makes it necessary to refine the results of MSER. In order to improve the time efficiency of the entire system, and based on the scanning method proposed in [127], a novel refinement scheme is proposed in Section 4.

Recall that the recognition system proposed in this paper does not only recognize the line markings but also recognizes pictogram markings (HOV and arrows painted on the road surface). To keep the details of line and pictogram markings while eliminating extraneous

Algorithm 2: 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 $x = \frac{width}{2}$ to 0 do
            if $P(x, y)! = 0$ then
                $x - -$ ;
                continue;
            else
                $y + +$;
                break;
else
    for $y = 0$ to height do
        for $x = \frac{width}{2} + 1$ to width do
            if $P(x, y)! = 0$ then
                $x + +$ ;
                continue;
            else
                $y + +$ ;
                break;
```

data, three stages are used:

- Stage 1: Finding the Minimum Bounding Rectangle (MBR) of all MSER blobs within input frames;
- Stage 2: Using the length-width ratio of each MBR to filter out non-line-marking blobs;
- Stage 3: Using the scanning method shown in Algorithm [127] to locate line marking edges.

Stages 1 and 2 aim at selecting blobs with similar shapes to the line marking blobs (based on the length-width ratio). Stage 3 is used to reduce the input pixels in the next detection stage by creating a one-pixel-width edge for line marking blobs.

Finding Minimum Bounding Rectangles

Minimum Bounding Rectangle (MBR) is the 2-dimensional form of Minimum Bounding Box and is defined as the smallest rectangle that contains every point within a given blob. MBR is used in our work to envelop desired regions (blobs of line and pictogram markings) located inside lanes, as well as undesired blobs outside lanes. Undesired blobs are then filtered out. Specifically, according to the width-length of each rectangle, we can select appropriate blobs. As a result, rectangles of the minimum area enclosing each input (blobs) must be produced. Fig. 3.9 and 3.10 demonstrate the process of generating an MBR. First, for each blob generated previously (Fig. 3.9), we draw a bounding rectangle which envelops the current blob, and shrink its boundaries until they meet the edge of the contour of the blob (as shown in Fig. 3.9); then, we record the new rectangle (red rectangle in Fig. 3.10). Second, we rotate the bounding rectangle by an angle of 10 degrees and shrink again following the same procedure performed previously. We repeat this process for each angle, increasing by 10 degrees; in this way, we cover the entire 360-degree plan. From the 36 bounding rectangles that have been generated, we select the smallest one that contains every point in the blob, and we eliminate rectangles that do not contain all blobs pixels (which might be smaller than the blob and does not contain all of its pixels). This algorithm is performed for all blobs found in the previous step.

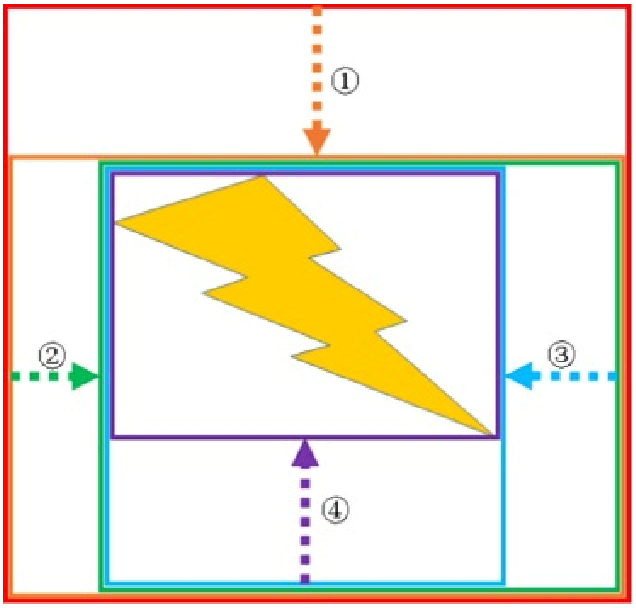


Figure 3.9: Bounding Rectangle

The shrinking process used to generate a Minimum Bounding Rectangle: Step 1 begins at the top of a rectangular box and continues until it reaches an edge point of the closed region; Step 2 and 3 start from the left and right sides, respectively, until edge points are met; Step 4 begins at the bottom and stops when edge points are reached.
Shrunken rectangles for every rotation. The rotation increment is 10 degrees.

Length-width ratio

After MBR-stage, a series of rectangles enveloping blobs are generated. The length-width ratio of each MBR is then calculated. It is commonly known that line markings are usually much slimmer than other objects (as shown in Fig. 3.11). In other words, it is easier to extract road lines from blobs, which have an exclusively large length-width ratio. Empirically, we found that the length of a line marking rectangle is usually more than twice the length of its width. This can be used to differentiate the potential line blobs from other objects. During our experiments, we noticed that some other slim objects (e.g. trees and electrical poles in Fig. 3.11) can be erroneously extracted as line marking blobs. To address

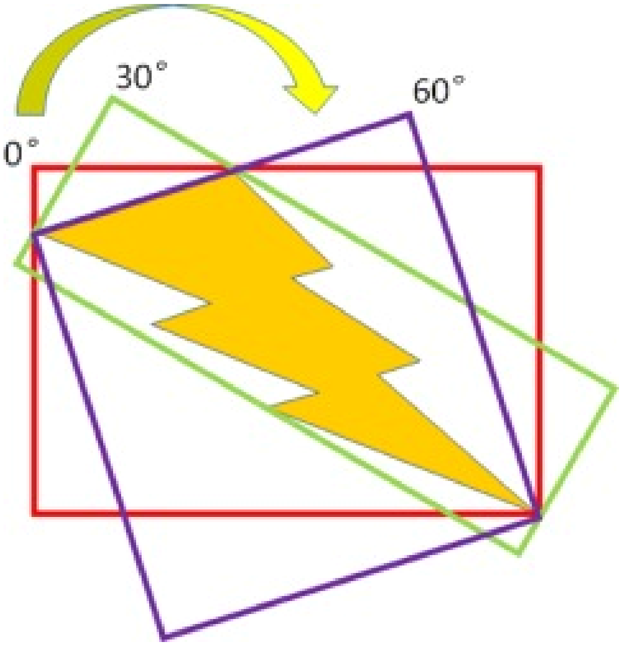
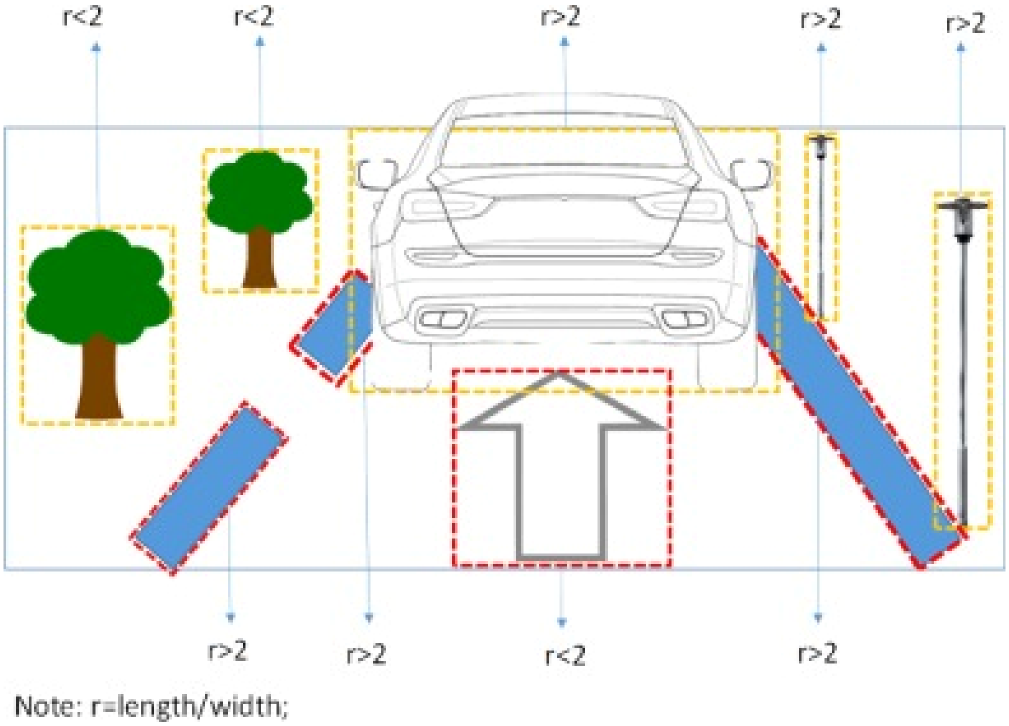


Figure 3.10: Rotate to find Minimum Bounding Rectangle

this, a scanning method called MSER Refinement is applied afterwards to exclude those slim outliers.

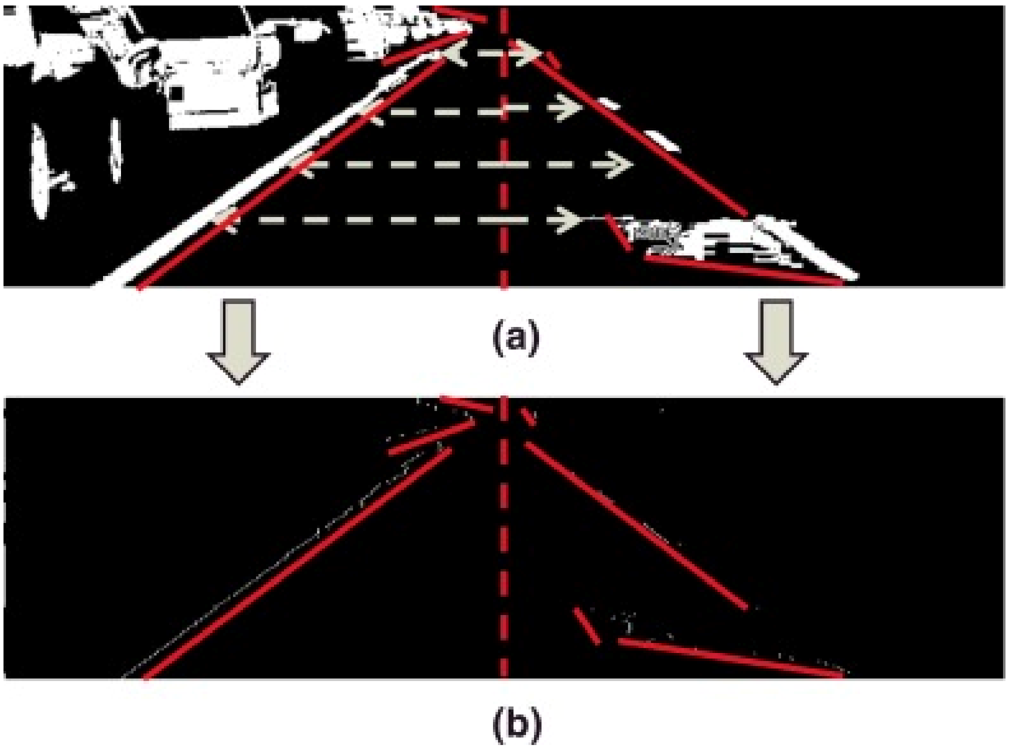
Scanning Method

Empirically, we found that MSER-blobs contain considerable extraneous details as well as desired pixels, as shown in Fig. 3.12. This makes it necessary to refine the results of MSER segmentation stage. Based on the fact that the number of objects located between the left and right line markings is lower than those outside the line markings (as shown in Fig. 3.12), it is reasonable to say that line marking blobs are located near the middle column (as the red dash line in Fig. 3.12) compared to other blobs located outside lane boundaries. As a matter of fact, areas between line markings are mainly road surface, which has very weak luminance compared to other objects in grayscale images. Since MSER only extracts the stable extremal region, undesirable points within the left and the right line markings can be eliminated from MSER-blobs. This is different from edge detection which extracts features of both stable extremal regions and unwanted regions.



Note: r=length/width;

Figure 3.11: MSER blobs detected


(a)

(b)

Figure 3.12: Scanning process starts from the middle column

The length of a line marking rectangle should be more than twice the width in Fig. 3.11. In Fig. 3.12, the red dash lines in (Top) and (Bottom) in the middle indicates the middle column of the ROI, while dash arrows in (Top) indicate the scanning direction of left and right areas divided by middle column. For every row of each area, the scanning process stops when the first white pixel is reached by the arrow. Red solid lines depict the entire contour after refinement.

Hence, we propose a scanning method for the binarized picture as described in Algorithm 2. Starting from the middle pixel of each row, scanning is performed in both left and right directions. MSER blobs are drawn in white, while the non-MSER area is kept black. The scanning of each image row stops when we find the first white pixel in left and right areas, respectively. The output of the proposed scanning method is MSER blobs shrunken into pieces of lines, which are actually partial contours of MSER blobs (as shown in Fig. 3.12). More importantly, because the scanning process starts from the middle column, these contours only belong to blobs that are close to the middle column in left and right areas. At its most extreme, this method actually depicts the contours of line-marking-blobs which are close to middle column. Moreover, the proposed scanning rule results in selected contours that are one pixel in width, which minimizes extraneous blobs and makes long linear features more easily recognizable by Hough transform. However, the proposed scanning method has two drawbacks.

First, even though extraneous points within areas between left and right line markings rarely form MSER-blobs, real-life scenarios occasionally have MSER-blobs located between left and right line markings (coming from cars or an area of erosion on the road, shown as red circular areas in Fig. 3.13). Our proposed scanning method might inevitably take the contour of those extraneous blobs as line marking candidates, and then feed those pixels together with the real line marking pixels to PPHT. To eliminate unwanted blobs, as shown in Fig. 4, we proceed as follows: we know that the scanning method only selects at most two pixels in a row (one pixel per area), which results in selected line candidates that are only one pixel in width. This dramatically weakens the contour of annoying blobs between line markings and makes the continuous contours of line marking blobs more prominent. On the other hand, PPHT can further remove the contours of MSER-blobs by thresholding length and angle of detected line segments.

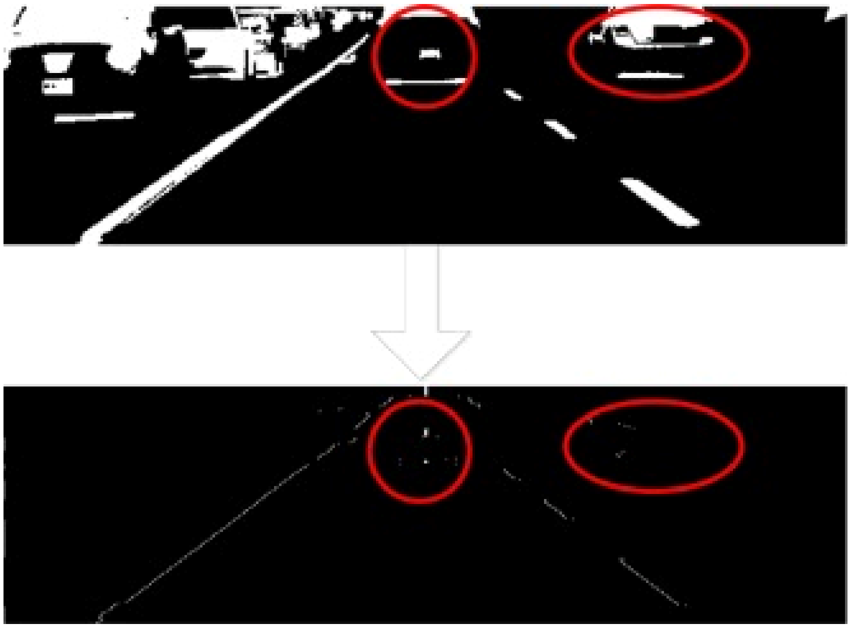


Figure 3.13: Drawbacks of the proposed scanning rule: blobs between line markings and outside the current lane.

The second drawback of the proposed scanning method is that, for dashed lines, blobs outside line boundaries (described as a red ellipse area in Fig. 3.13) have white pixels in rows between dashes. This might create additional undesireable contours for those rows. Experimentally, we found that PPHT can handle the above issues by thresholding the length and angles of line candidates. That is, by an appropriate thresholding, lines located in irrelevant regions will not be selected as line markings.

3.2 Line Detection and Recognition

Line detection, as well as line recognition, was applied to help to detect lane markings since both line and markings detection share a common process which enabled us to integrate our previous work about line detection and other applications based on MSER in a systematic structure.

3.2.1 Features of Road Line

Line markings are the markings painted on the road surface as the boundary of sub-lane for vehicles travelling in the same or different directions. The most common line types in North America are dashed lines and solid lines.

A solid line is a long rectangle painted on the road parallel to the direction of the road itself. For instance, there usually exists one solid line in the middle of roads which divides the road surface into two parts. Cars travelling in different directions use only their own side, e.g. right in North America, according to the traffic rules to avoid collision.

Dashed lines, also known as broken lines, appear as discrete rectangles painted on the road surface, which have the characteristic of collinearity on their long-sided edges. This type indicates that lane changing through this line is allowed.

Colours, such as white and yellow, contribute a lot of information to the traffic system. Generally, white lines, regardless of whether they are solid or dashed, represent that all the vehicles adjacent to it are moving in the same direction. On the contrary, vehicles from different sides of a yellow line, regardless of whether the line is solid or dashed, are moving in opposite directions.

3.2.2 Road Line Detection

By applying Hough Transform after MSER, lines pointed toward the vanishing point can be extracted and clustered. The processing can be shown in Fig. 3.14.

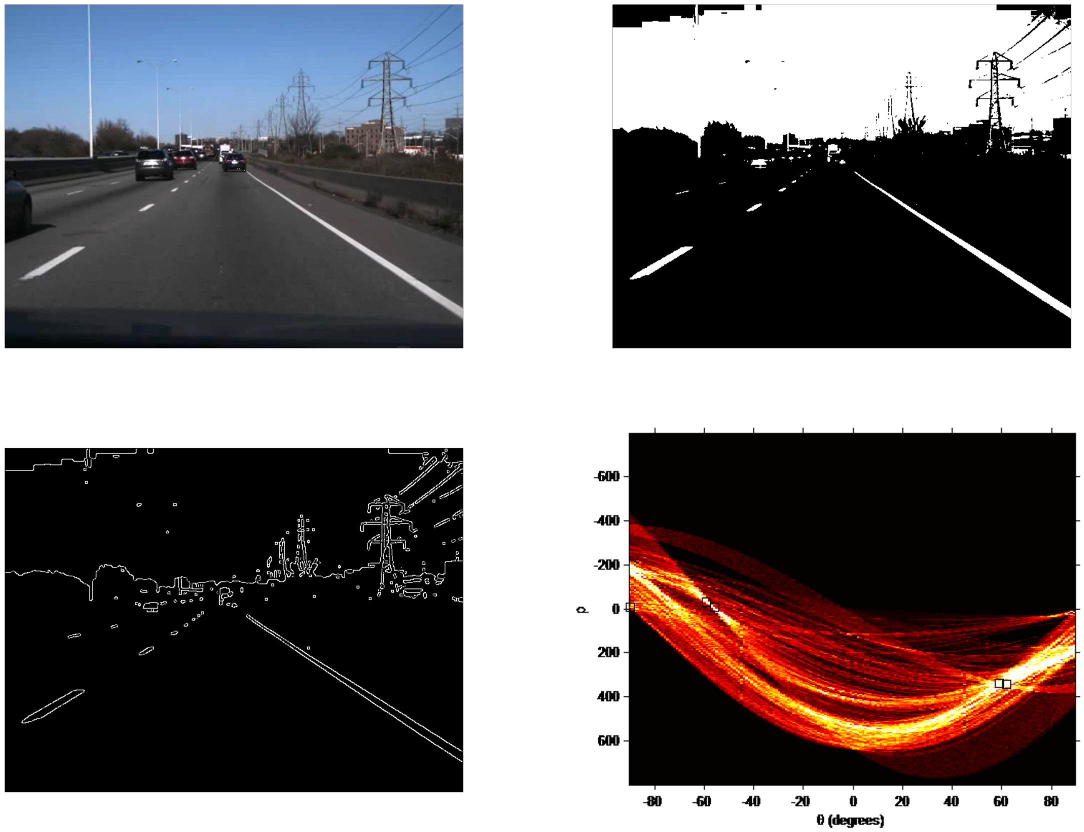


Figure 3.14: Images from left to right in first row and second row are: Original Image, Polarized Image, Edges detected in Image, Lines in Hough Space

After polarization, edge detection, and line detection in Hough Space, the number of candidates has been dramatically decreased. An example was shown in Fig. 3.15.

3.2.3 Road Line Recognition

After Hough transform, a set of lines were extracted where the lines we are interested in usually lie around us with shortest distance among all candidates. Applying this feature, the longest and closest lines from the left side and right side were chosen as the selected lines for further analysis.

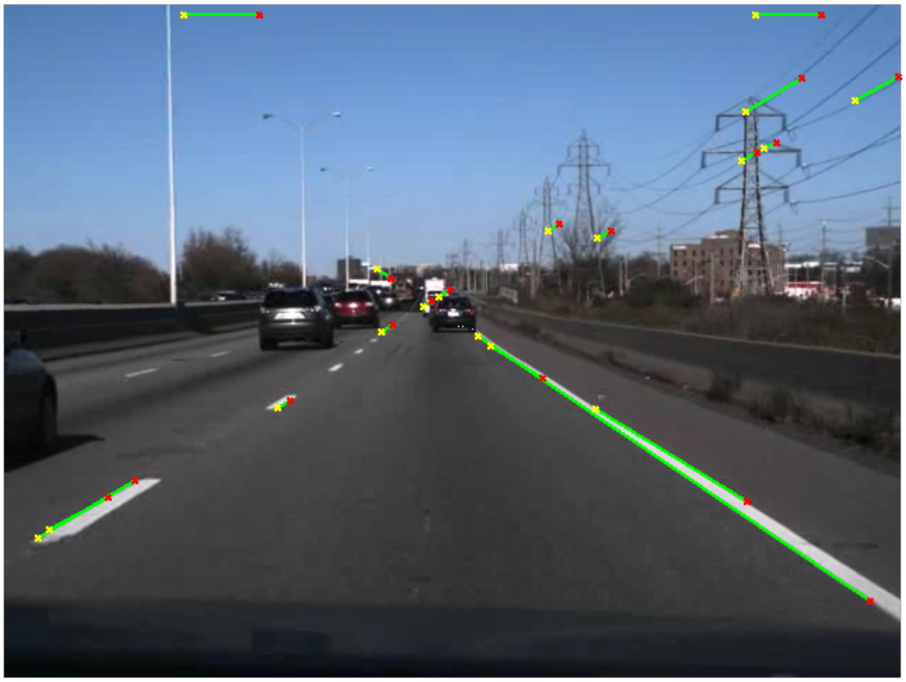


Figure 3.15: Candidates Lines extracted by Hough Transform

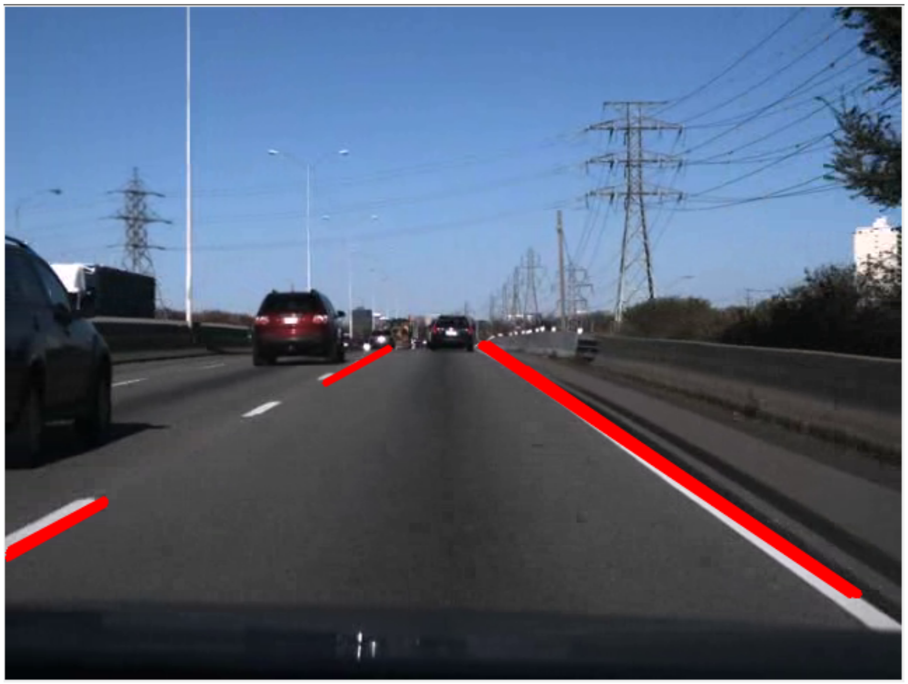


Figure 3.16: Chosen Lines after selection

Color Recognition

The original image captured by the camera was coded in RGB space. While it required too many cuts, when separated into white and yellow,

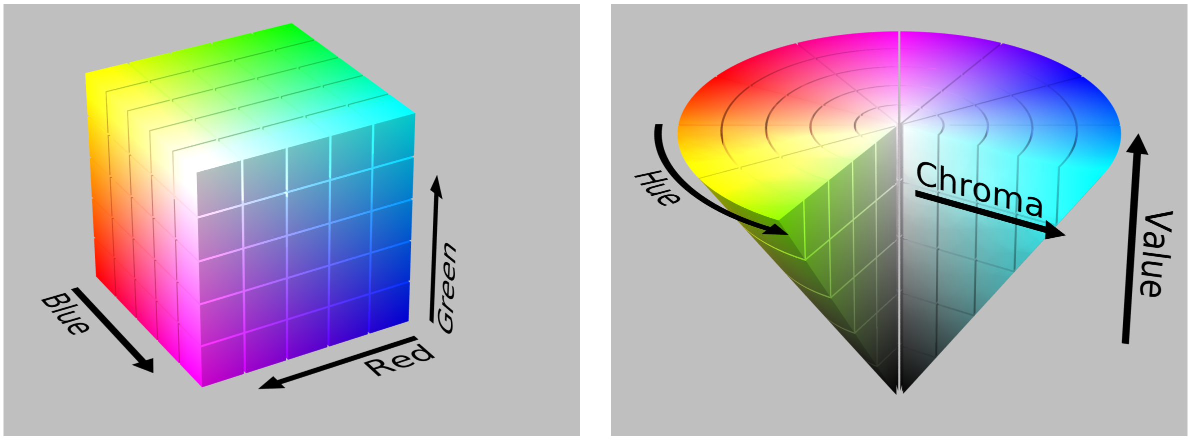


Figure 3.17: RGB-Color Space(LEFT) and HSV-Color Space(RIGHT)

Two dimension thresholds (Hue and Chroma) can easily separate white and yellow in HSV space.

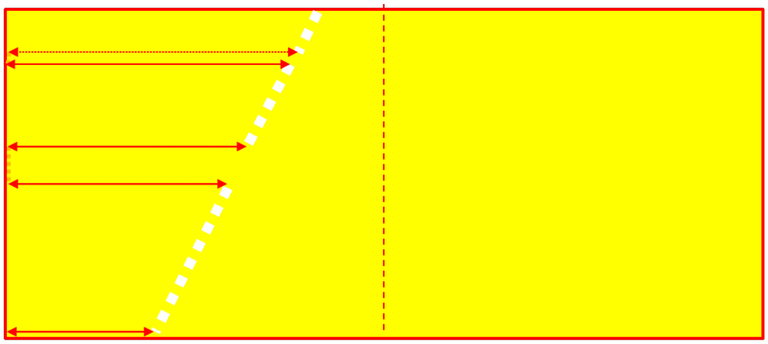


Figure 3.18: Color of pixels in the edge

By analysing the distributions of kinds of colors, several groups are collected from the points produced by MSER. Among all, the main color represents the colour of lines in each side separately.

Shape Recognition

According to the features of the related graphs, we can see dashed lines and solid lines display obviously different behaviours.

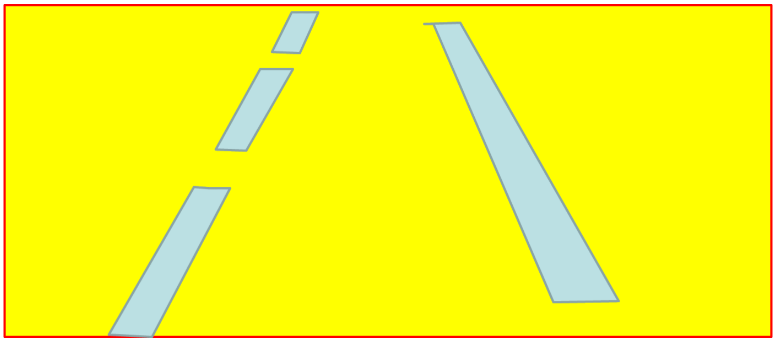


Figure 3.19: Original road line blobs

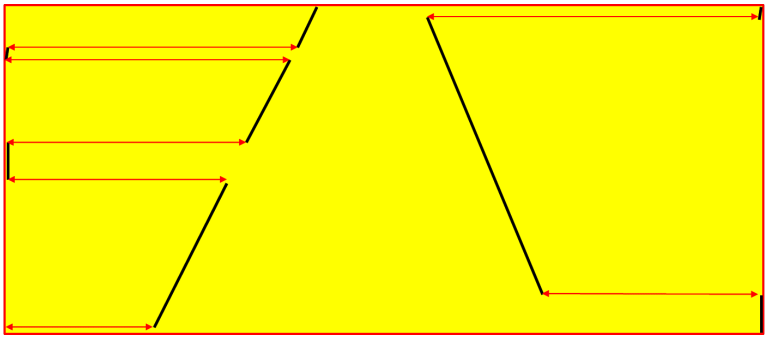


Figure 3.20: Edges of road line after scanning

$$\Delta_h = \left| \frac{\rho(h-1,h) + \rho(h,h+1)}{2} \right| \tag{3.10}$$

$$h\ is\ an\ edge\ of\ a\ line\ iff\ \Delta_h \gg \Delta_{h-1}.$$

Most notably, each point of the fitting line of the dashed line has more discontinuity than others, which can be explained by the natural shape of the dashed line. What should be pointed out is that the threshold of the numbers of gaps and other properties of the dashed line rely on ROI and the results of our experiments.

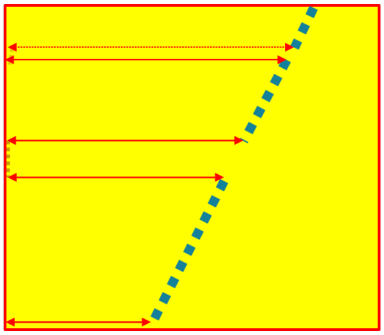


Figure 3.21: Gaps between dash lines

3.3 Markings Detection and Recognition

3.3.1 Features of Markings

The traditional method used for marking recognition is based on the invariant of the pattern on a 2-D plane. For instance, Hu-moment and HOG features are invariants with respect to translation, scale, and rotation. Such methods can perfectly match the requirement of a road sign. The competition results of German Traffic Sign Recognition Benchmark (GTSRB) indicates that the latest CNNs performance has already exceeded human beings, reaching 99.64% recognition rate. However, the performance of recognition of road surface markings is not good enough. The main reason for this phenomenon is the distortion of the image. Due to variability of distance and depression angle, the image of the object has unique distortion, which is hard to classify through traditional features mentioned in references.

In order to solve this problem, IPT was used by most of the researchers, where the method relies on the internal parameters and external parameters measured in real time. Although the residual processing after IPT can be transformed into a traditional recognition problem which was already learned in 2.3, this serialization method also decreases the overall performance.

It was clearly shown in the history of image processing that for a specific purpose the optimized method has its comparative advantage. For example, SIFT is considered most effective in face recognition when it was published in 1999 for general object detection. Similarly, HOG was first described in 2005, and was initially used for pedestrian detection in static images. Since road surface markings are usually standard geometric diagrams or combinations thereof, we believed that there existed some feature which can be used to describe the markings with constant distortion.

In a model of the ego vehicle and its closed area, a metric based on Euclidean space assumption can be used. According to a method of geometric-transillumination, the distortion of a pattern is caused by distance. For instance, a house looks larger when it is close to the observer. For a line lying perpendicular to the optical axis, the width of it may increase evenly when it moves from far away to the closing area. For lines which do not lie perpendicular to the optical axis, sections of the line closer to the closing area will increase in width faster than sections further away from the closing area, taking length and perspective into account.

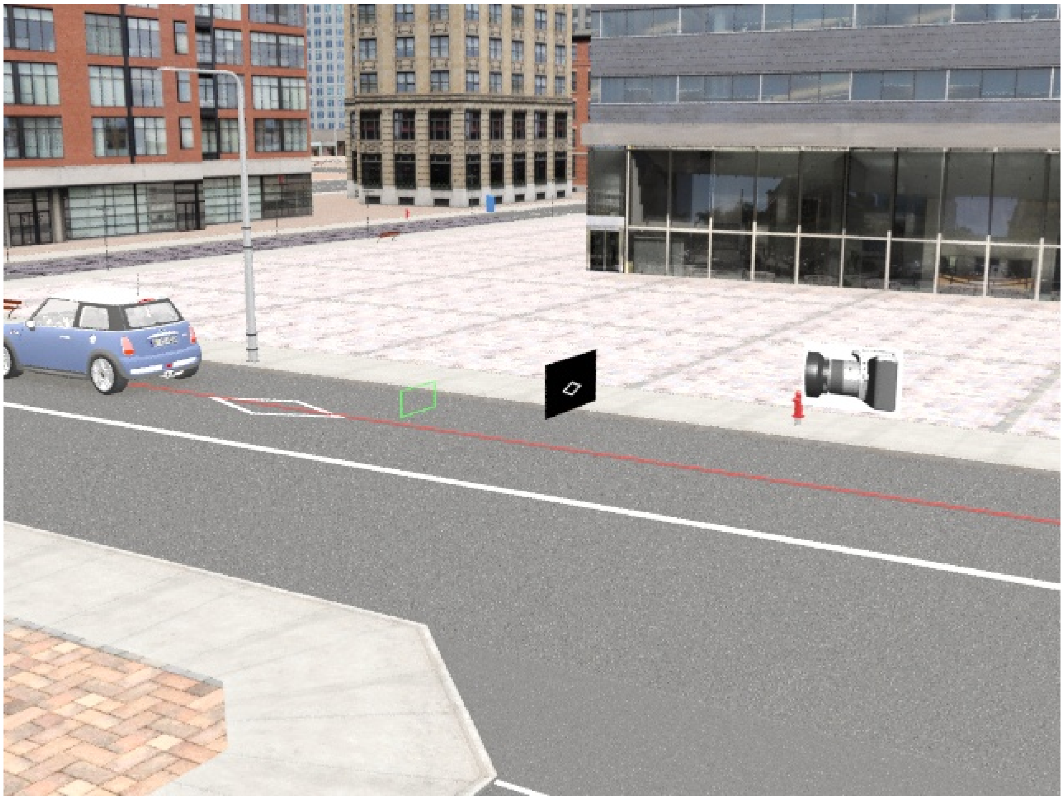


Figure 3.22: Relation between object in real world and its image

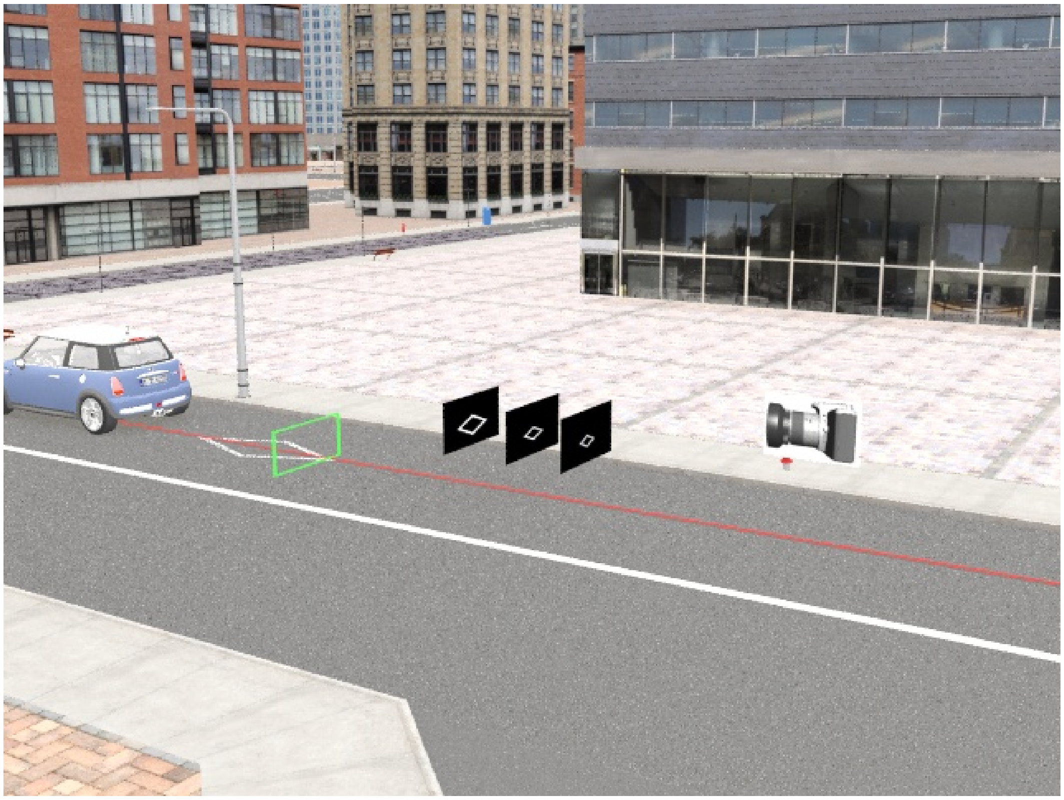


Figure 3.23: Change of image in different distance

As shown in the figures above, three parallel images indicate the image captured by a camera from a moving vehicle successively. While different from the original image of diamond which has four axises of symmetry, the image with distortion has only one axis of symmetry. In particular, the image is asymmetric if the moving direction of the car is different from the road lane direction but can be calibrated. In this perspective, it is possible to use the symmetry of paintings to analyse and classify. In a generalized view, such symmetry features can be understood as the object has maintained its width relatively in the virtual plane perpendicular to the focal plane. Through using this relation, it is easy to classify the markings with distortion even though some markings like arrows are not symmetrical. The following content will give the feature of HOV and arrow markings.

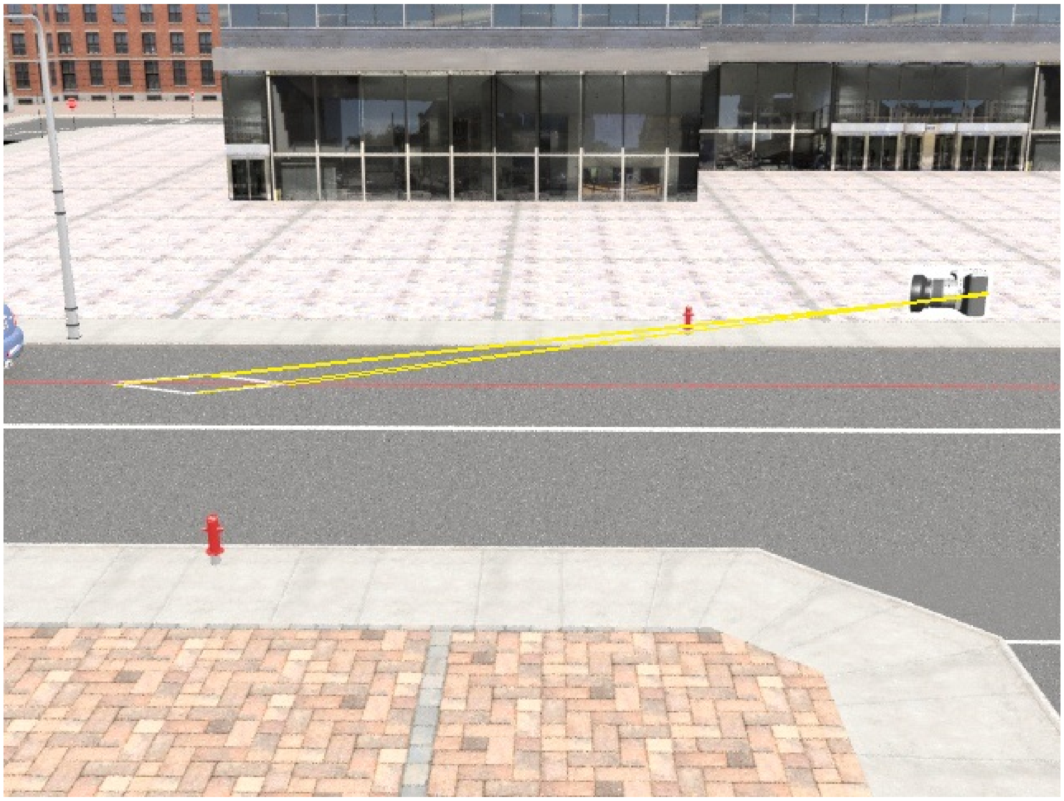


Figure 3.24: Sideview of the whole system

HOV Markings

HOV lane is short for High Occupancy Vehicle lane. It was also known as an HOV, carpool lane, diamond lane, and transit lane or T2 or T3 lanes in Australia and New Zealand. This restricted traffic lane is reserved at peak travel times or longer for the exclusive use of vehicles with a driver and one or more passengers (regulations vary in different countries), including carpools, vanpools, and transit buses.

According to a report from Transport Canada [128], Canada's first HOV lanes were developed in Greater Vancouver and Toronto in the early 1990s. These were followed shortly afterwards in Ottawa, Gatineau, Montreal and later in Calgary. There are now approximately 150 lane kilometres of highway HOV facilities in 11 locations in B.C., Ontario, and Quebec. There are also over 130 lane kilometres of arterial HOV facilities in 24 locations in Greater Vancouver, Calgary, Toronto, Ottawa, and Gatineau. Additional highway and arterial facilities are being studied or are planned in locations across Canada, including large-scale extensions of existing HOV lanes and new facilities in Toronto and Vancouver.

Even with such widely usage of the HOV sign and its derivatives, there are no official requirements or design standard in this field for the urban area. (Regulations already exist for the symbol in air transportation and rail transportation. Guidance for transportation practitioners like the Ontario Traffic Manual can be found in [16] but are not legally binding.) That leads to a situation where we cannot use specific measurement information to judge if a marking is or not an HOV marking. As a result, we found that the universal character of an HOV sign is the nest relation between two diamonds and its two axes of symmetry.

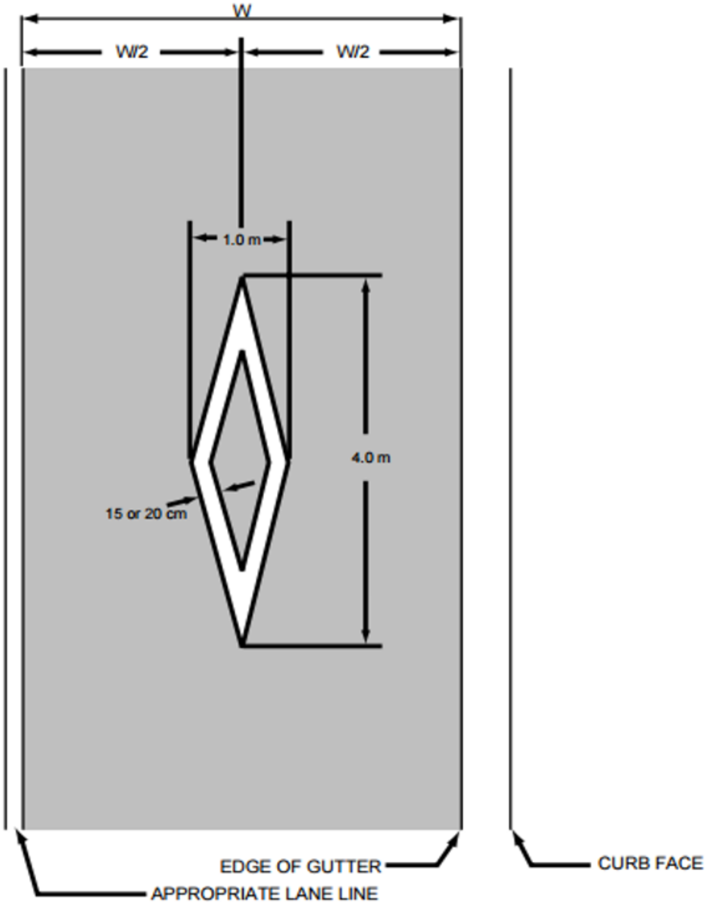


Figure 3.25: HOV sign definition by [16]

Arrow Markings

Similar to HOV detection, it was also hard to define what an arrow is generally. In this thesis, we treated them as a combination of a triangle and a body. While most of the triangles used in road marking were isosceles triangles, it was extremely difficult to describe the bodies since the shape and position of each marking differs considerably. Nevertheless, we still found through observation that as a rule, the centroid of the body was closer to ego vehicle than the triangle when the whole arrow was in front of us.

There are numerous types of arrow markings in the world. With different sizes and shapes in different countries, arrows can hardly be recognized simply based on the proportion of each part. However, finding the boundary of of arrows body can help distinguish their types. An arrow marking contains a body which is similar to a rectangle just like short dashed lane markings (as shown in Fig. 3.19). Similar to HOV detection, we found multiple axes of symmetry existed in the body and triangle part of the arrow separately.

3.3.2 Markings Detection

In this section, we will discuss a new method to recognize the contour of objects we are interested in. As was shown in Chapter 2, most of the methods already implemented are based on a template. Specifically, some of them are based on data-based training and others rely on precise measurement (especially in German).

Boundary set by Line Detection

Methods of line detection as well as methods of line recognition were discussed in Section 3.2. Since our final purpose is to design an ADAS, a framework including multiple functions is significantly important to us. As a result, line recognition results are used as part of the classifier to contribute to the recognition of other road markings. Specifically, it was used by defining new ROIs for the later filter. As an assumption of the proposed method, road markings including HOV and arrow markings are supposedly located inside of the two boundary lines of the road lane we are driving on. Such an accurate trapezoidal zone (or triangular zone if the vanishing point is in the ROI) can dramatically decrease the computation cost for marking detection.

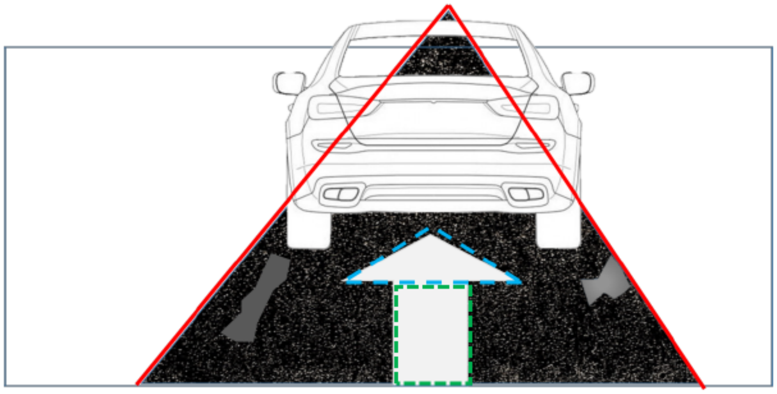


Figure 3.26: Cooperation between line detection and markings detection

Contour filter by area

There are fewer contours left between lane markings, while most of the irrelevant contours so far should be vehicles in front or road surface patches, etc. Comparing the markings in the ROI, it is reasonable to assume the areas of most vehicles are bigger than markings, while the patches are smaller. That led to the threshold of the area, which we defined in this case as big as 20 pixels (based on 640×480 resolution of the camera).

Contour filter by color

Different from the method used in Section 3.2.3, the colour of road markings should be close to pure white in RGB space. Hence in our case, we should at least calculate each object three times in all layers. Moreover, for each contour, the color distribution is uneven. We used the mean value of R, G, B layer separately and reserved only the contours which satisfied the condition ($R > 200, G > 200, B > 200$).

Relative location filter

After color filtering, most non-lane marking contours can be eliminated. However, in some challenging situations, multiple markings may exist within a single lane. In order to deal with this issue, we have to choose the marking which is vertically closest to the driver. To do so, we traverse all contour points and denote the top, bottom, leftmost, and rightmost points of all. By comparing the bottom points of contour candidates, we can choose the contour which is vertically closest to us. (as shown in Fig. 3.27)

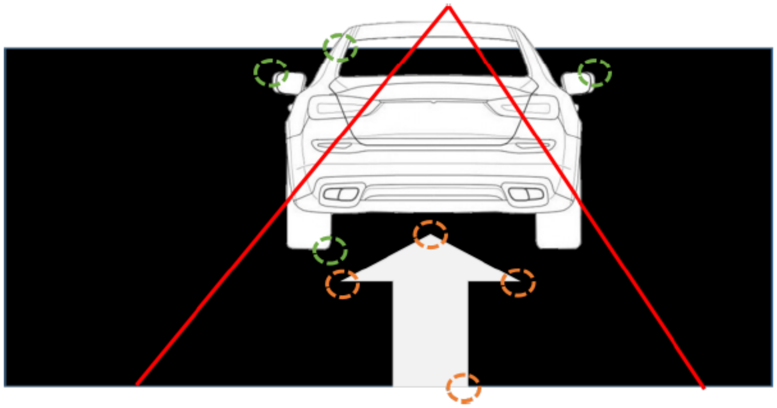


Figure 3.27: Performance of 4 extreme points set Descriptor with different objects

Convex justification

After the previous steps, the contours we obtained should be a convex or non-convex figure. For instance, convex graphs include HOV (high occupancy vehicles) signs, words enclosed within a frame, triangles, rectangles, etc. Non-convex graphs include arrows (straight, left, right, combinations thereof), X (railway crossings), words without frames, bicycle patterns, any other irregular figures, etc. Hence, it is necessary to distinguish convex and non-convex figures. To do so, we need to extract the point set of the outer edge of a contour. Then, we check if there are defect points between two adjacent points one by one followed the edge. If there are no defect points, we consider this contour as a convex figure, otherwise the figure is non-convex. (as shown in Fig. 3.29)

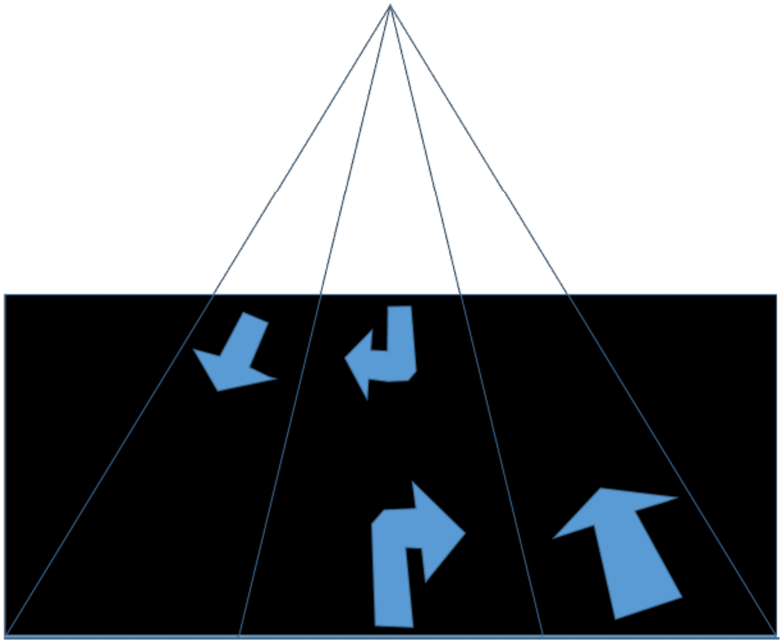


Figure 3.28: Segmentation by different lanes

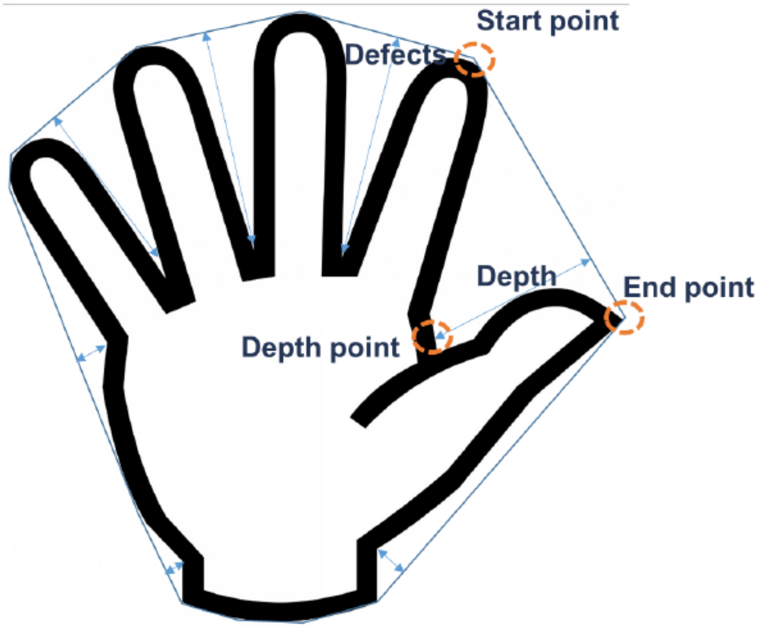


Figure 3.29: Measurement of Convex pattern

3.3.3 Markings Recognition

As introduced in Section 3.3.1, the features of HOV and arrow markings are different. To measure the differences between different classes, 4 extreme points set descriptor was introduced first in this subsection. The following sections will describe the recognition process of HOV signs and arrow markings separately.

4 Extreme Points Set Descriptor

Some similar work in the field was conducted using FAST corner detection [10], however in the real world, the corner detected only by FAST was not accurate enough because of the erosion of markings themselves. Although the combination of high definition mapping and corner detection results can provide good performance, the computation cost is still considerable. To solve this problem, we extracted a similar feature to "corners" but avoided traditional complex algorithms in our proposed method. Specifically, 4 extreme points set was used here which includes leftmost point, rightmost point, top point and bottom point. These four points are easier to extract from each blob detected. By scanning rows or columns from four directions, the run time of this algorithm is only $o(n)$. Fig. 3.30 and Fig. 3.31 are examples of usage of 4 extreme points set descriptor.

As shown in Fig. 3.31, we can obtain top point, bottom point, leftmost point and rightmost point separately following a similar method in Fig. 3.30. Among these 4 points, top point provides the most important information because it usually contains the orientation of the arrow. Although bottom points may shift/extend from the left to the right of the bottom line, the connection lines between the bottom point and other three points can also be analyzed to extract the information about the type of arrow.

HOV Markings Recognition

So far we know an HOV sign is made by two nested diamonds while the contents between the two contours are filled by white colour. In traditional Euclidean space, the diagonals

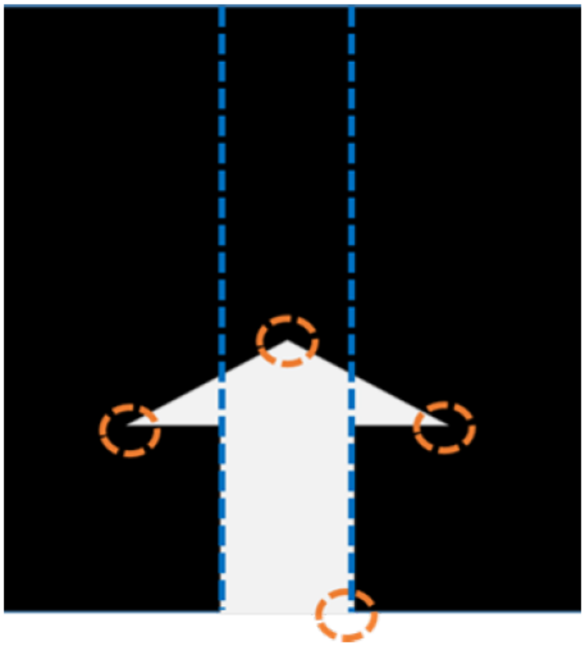


Figure 3.30: 4 extreme points set descriptor

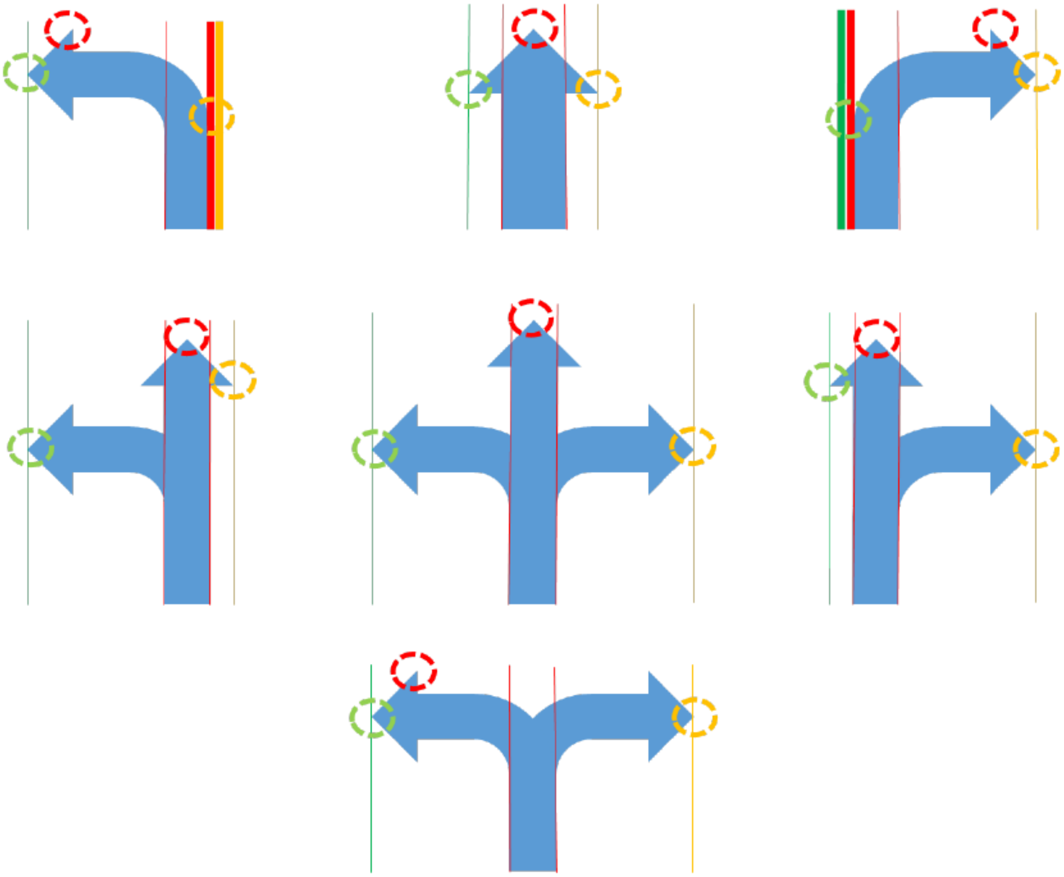


Figure 3.31: Distinguish between different arrow directions

of a rhombus are perpendicular (Shown in Fig. 3.32.c). In order to verify this character, we need to find the nodes of a rhombus.

To achieve this, we used two groups of vertical lines to approach to the node of rhombus from outside. [Fig. 3.32.d] After this, we verified the rhombus with an appropriate tolerance value of the angle between diagonals of the rhombus.

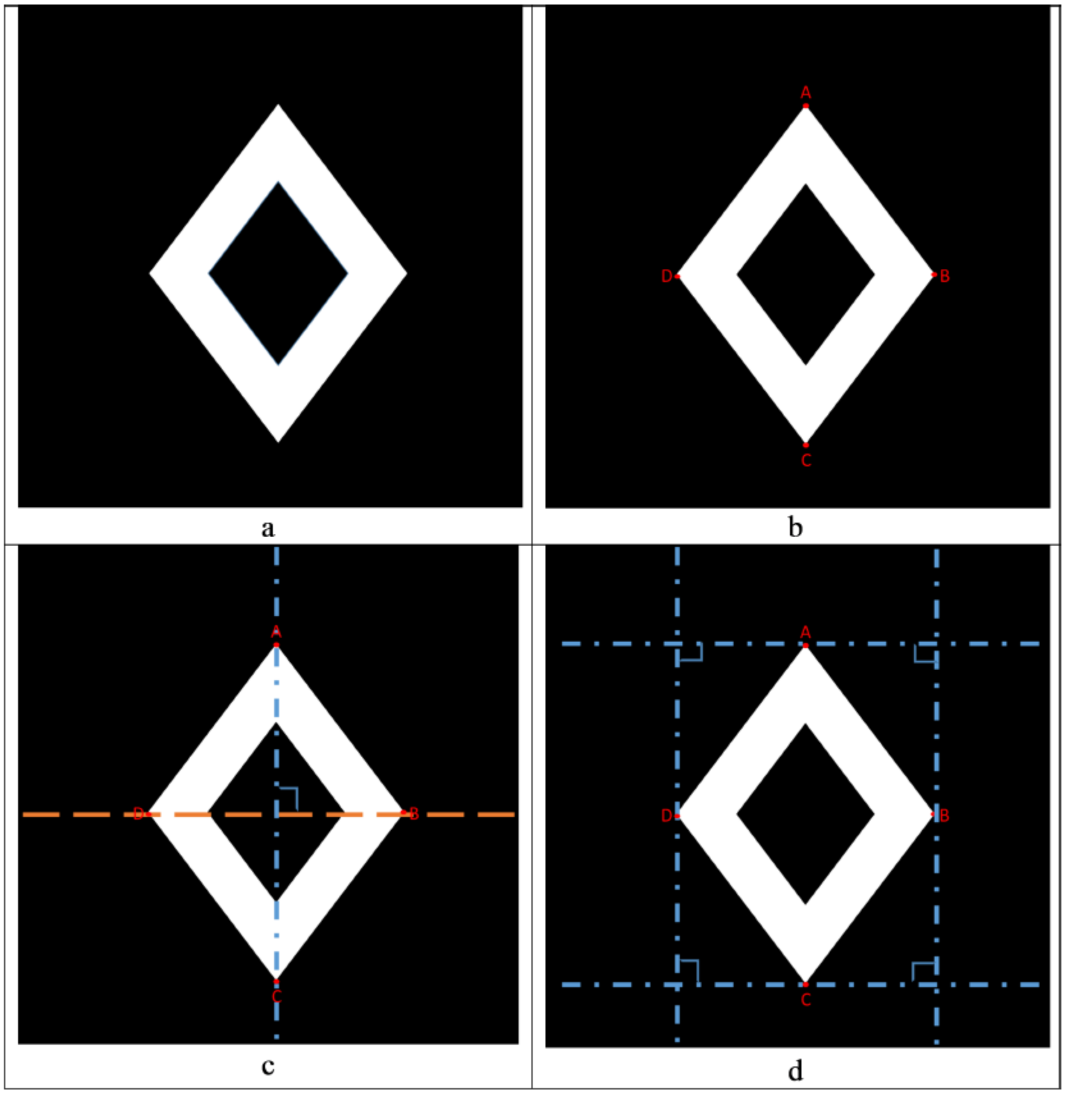


Figure 3.32: 4 extreme points of HOV sign

To narrow the detection range and focus on the contents on the road surface, lane markings should be used to separate contents on a road surface with outliers outside road surface.

In this case, only two lines should be selected.

In order to do so, we extended all the lines to reach the bottom of ROI. Within both the left and right area of the middle line, the point which is closest to middle line must be selected from all bottom points of lines (as shown in Fig. 3.33). The line with the bottom point closest to the middle line is considered as the left lane marking.

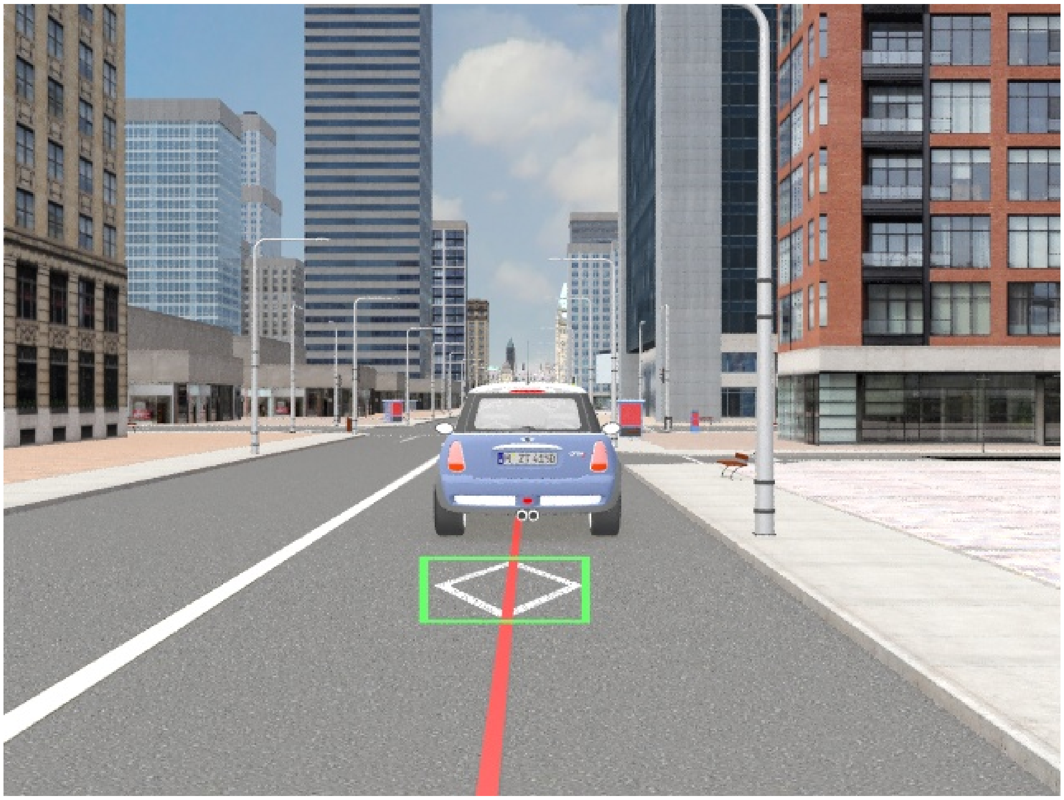


Figure 3.33: Virtual middle line and marking

The following step is to justify if a candidate in accord with the HOV sign. HOV is a special sign among all convex markings because its character is unique and easy to extract. Since we already obtained the top, bottom, leftmost, and rightmost points, the line which links top and bottom points should be perpendicular to the line connecting leftmost point and rightmost point. Also, the orientation of the line linking top and bottom points should point to the vanishing point. If a convex figure satisfies the requirements above, we regard this sign as an HOV sign.

Arrow Markings Recognition

The rules used in this part can be outlined in the table below:

Table 3.2: 4 extreme points set rules

	Left	Middle	Right
Top Point	Turn Left	Go Straight	Turn Right
Left Point	Turn Left	n/a	n/a
Right Point	n/a	n/a	Turn Right

Where the "Left" means the point is far to the left side (with a thresholder distance to left boundary) of the left straight boundary, the "Middle" means the point is between left and right straight boundaries, while the "Right" means the point lies far to the right side of the right boundary (a threshold for the distance to right boundary is needed as well). Fig. 3.34 describes how the rules associate with actual arrow directions.

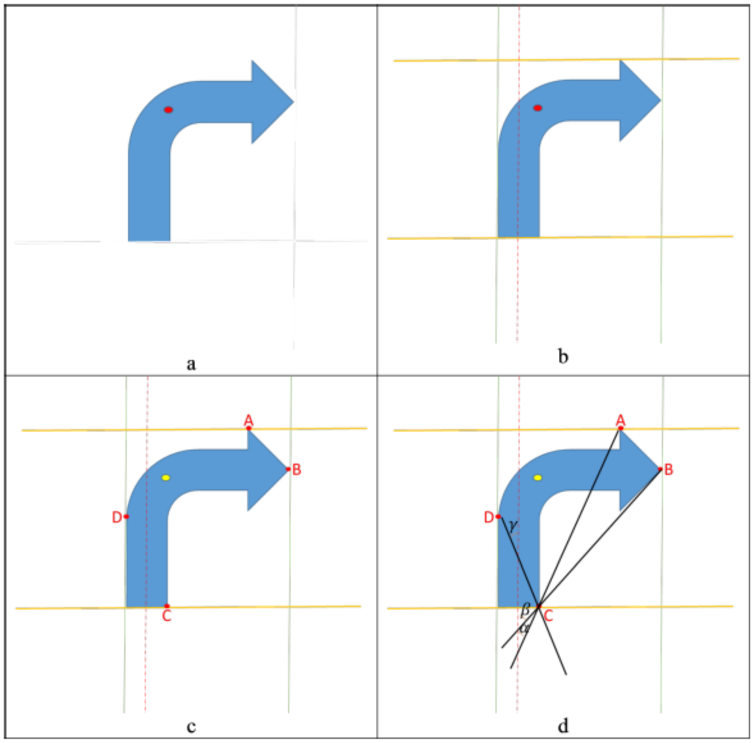


Figure 3.34: Relationship between 4 extreme points

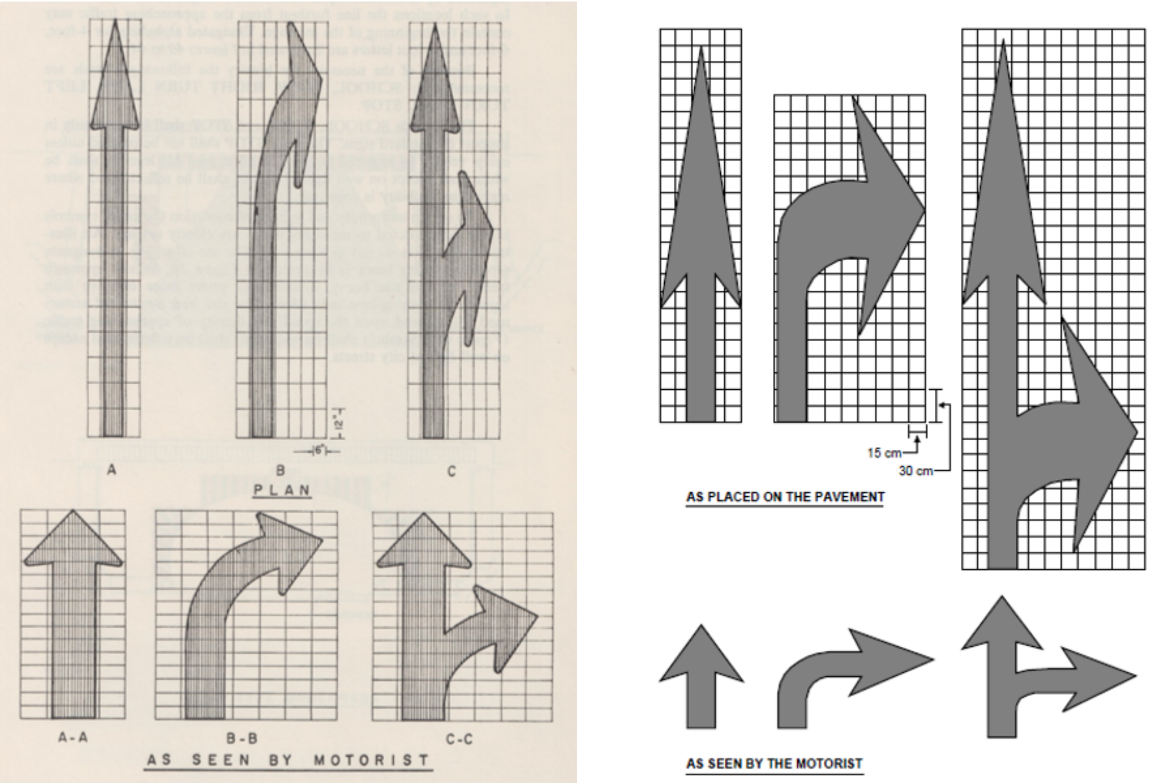


Figure 3.35: Standard of Arrow Painting in 1960 [17] and 2014 [18]

According to Manual of Uniform Traffic Control Devices for Canada 1st edition [17] and 5th edition [18], Standard arrows on pavement are defined as above. After calculation, we can grab some information from the key points of each sign.

In Fig.3.36.d, we assumed that the suspected object was adjusted properly with respect to the orientation of the road lane itself. For instance, we made a simple rule to distinguish most common arrows in Ottawa follows the range of parameters list in the Table 3.3 below.

Table 3.3: 4 extreme points based classification rules

	Left Turn	Straight	Right Turn
α	(- $\pi/4$, - $\pi/6$)	(- $\pi/6$, $\pi/6$)	($\pi/6$, $\pi/4$)
β	(- $\pi/2$, α)		(α, $\pi/2$)
γ	[0, $\pi/2$)	$\lvert\beta + \gamma\rvert < \pi/6$	(- $\pi/2$, 0]

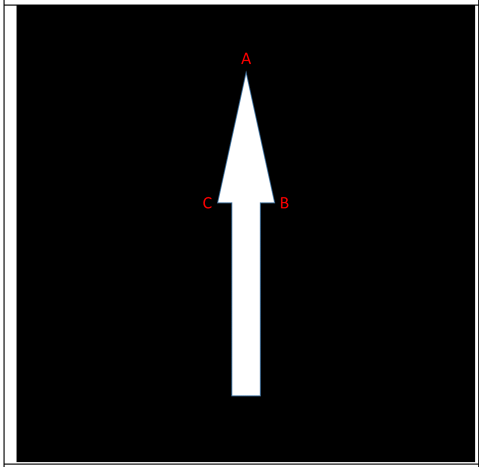
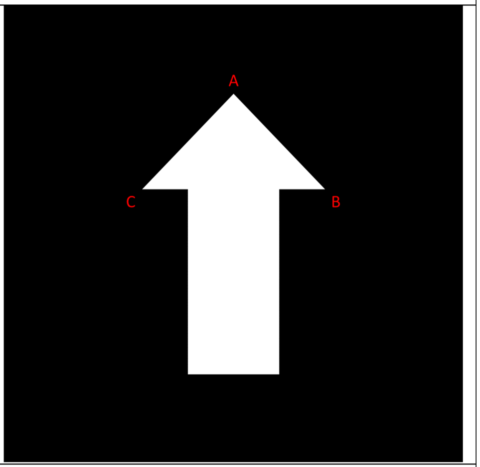
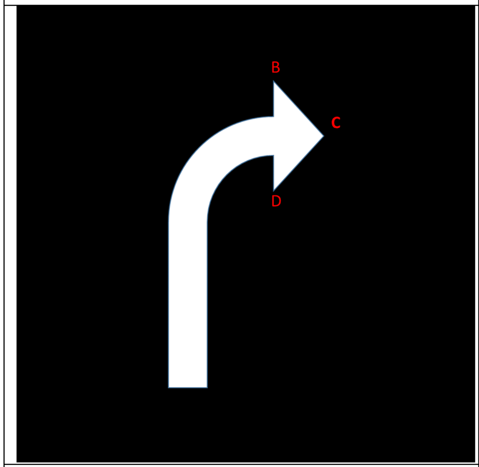
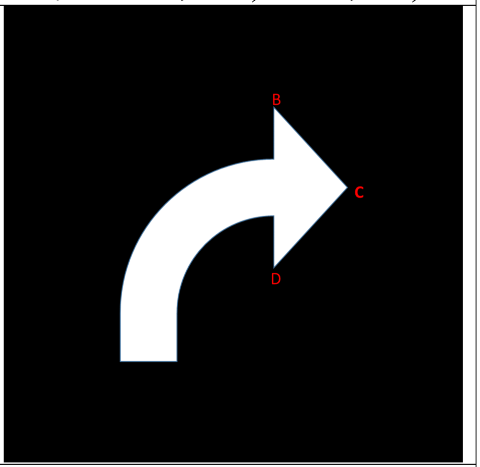
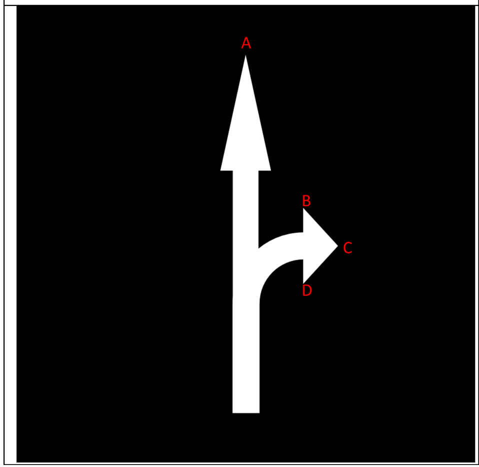
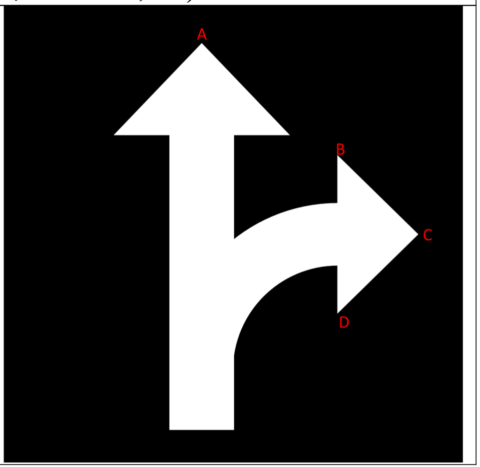
Bird View	Camera View
	
$A=2\arctan(0.25)=28^\circ$	$\alpha \in (-5^\circ, 5^\circ)$, $\beta, \gamma \in (-23^\circ, -10^\circ) \cup (10^\circ, 23^\circ)$
	
$B=\arctan(0.5)=27^\circ$, $C=139^\circ$, $D=\arctan(0.25)=14^\circ$	$\alpha \in (8.9^\circ, 18.9^\circ)$, $\beta \in (27^\circ, 35^\circ)$, $\gamma \in (-90^\circ, 0^\circ)$
	

Figure 3.36: Distortion from bird-view to horizontal view

Meanwhile, we found that there exist combinations of different arrow on the road. With the same numbers of samples, it was hard to differentiate them from each other. To solve this problem, we introduced the middle line of the road according to the line detection section.

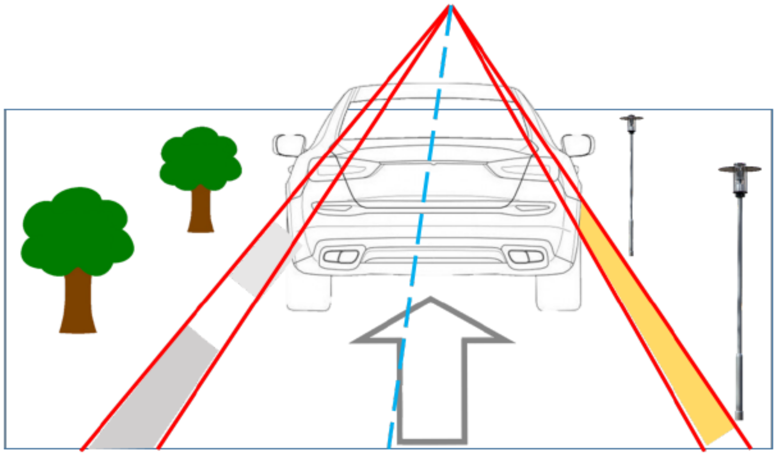


Figure 3.37: Blobs selected with road line constraint

The two lines obtained by 3.2 can be used to narrow the detection range. For each MSER contour, we checked if the contour is located between these two lines. To avoid traversing all points of MSER contours, we can randomly select one point from the contour points, compare its position with the line position (as shown in Fig. 3.38). This is quite a compromise to take only one point as the representative of each contour, which is actually based on the tolerance of contours sitting across lane markings.

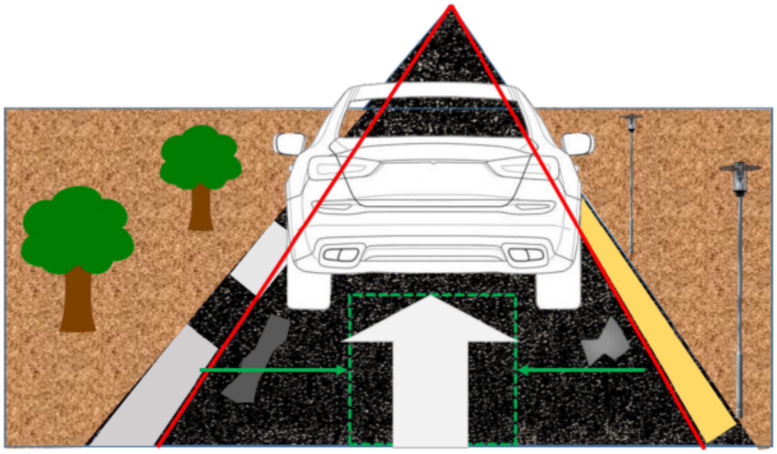


Figure 3.38: Contours filter by road line

Through this, we can not only compare the sample points with the pattern itself, but also acquire the relationship between arrow and road line, which can obviously increase the accuracy of our system.

3.4 Tracking and Enhanced Analysis

After the process of preprocessing, detection and recognition, we have already obtained enough information about road lines and road markings. However as mentioned before, changes of the ego vehicles moving direction may cause errors in the proposed method. To solve this problem, an innovative tracking mechanism and enhanced analysis method are introduced here.

3.4.1 Vanishing Point Tracking

Vanishing point is the basis of perspective relationships in the proposed model. To improve the performance of the algorithm, calibration of road direction is necessary. Since the direction of the road here is defined as the sequence of the vanishing point, tracking vanishing point naturally becomes the main problem which urgently needs to be addressed.

As quoted in Chapter 2, Kalman Filter and Particle Filter are two main popular tools used for tracking. An obvious inertia of the vanishing points movement can be observed after multiple experiments. To verify our assumption, a clip Demo3 was chosen to explore the characteristics of the vanishing point.

As is shown in Fig. 3.40 ,the sequence of the vanishing points' position is smooth. In other words, the series shows similarity between observations as a function of the time lag between them. In order to prove our assumption, we have obtained a data series of Demo3 and used Autoregressive Integrated Moving Average (ARIMA) model to analyze it.
In order to use ARIMA(p,d,q) model, it is necessary to obtain three parameters p,d,q. Among them, p is the number of autoregressive terms, d is the number of nonseasonal

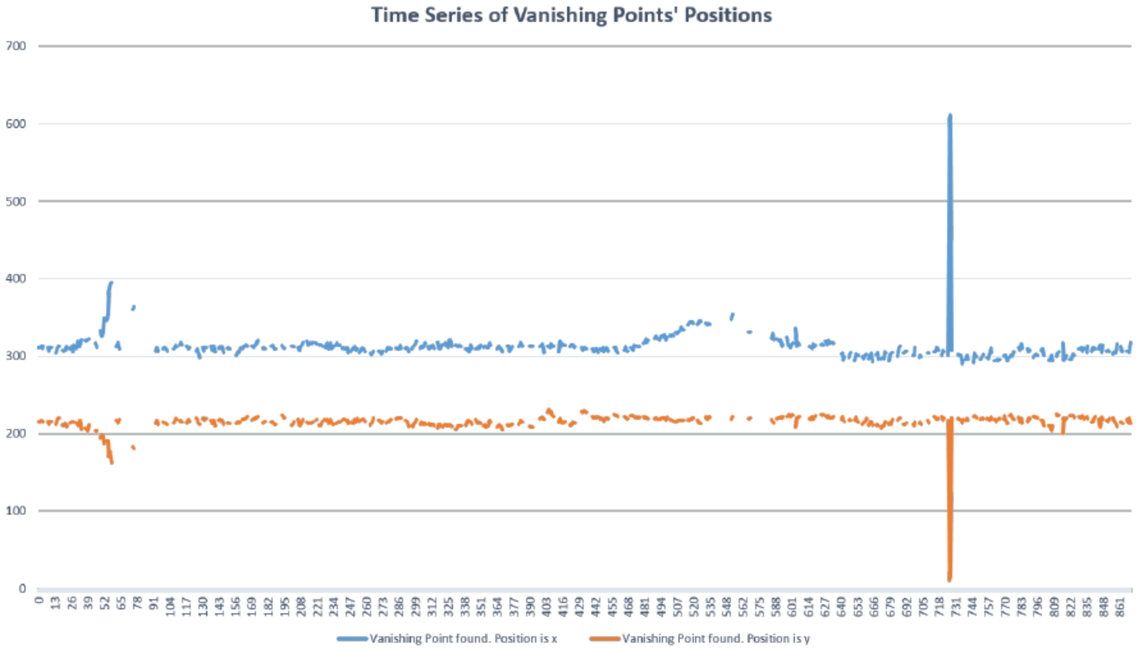


Figure 3.39: Time Series of vanishing points' position in Demo3

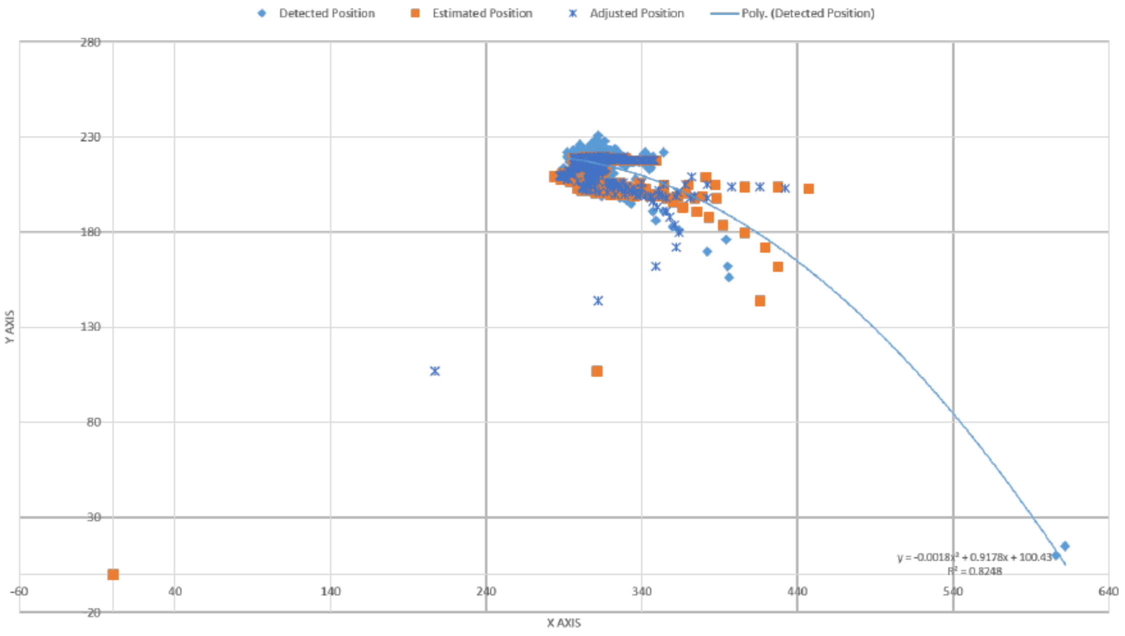


Figure 3.40: Comparision of tracking result and real value

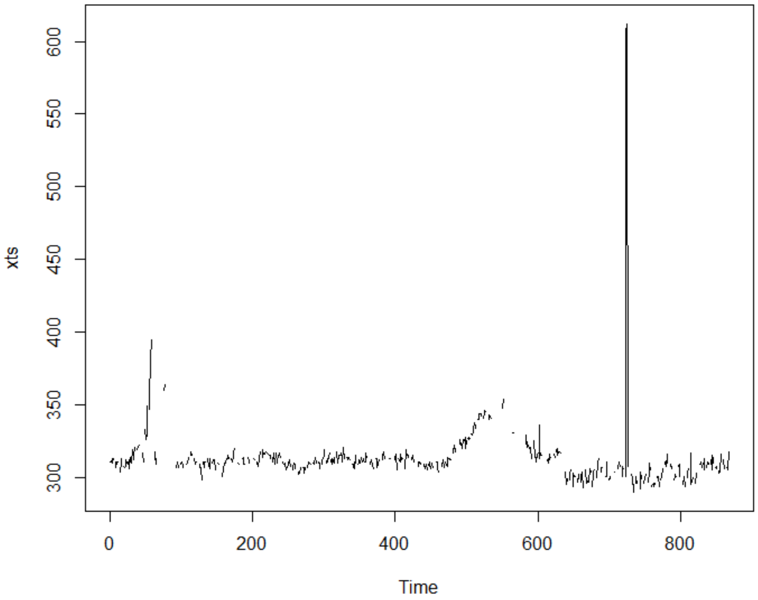


Figure 3.41: 2nd difference of time series for Demo3

differences needed for stationarity, and q is the number of lagged forecast errors in the prediction equation.

First of all, the x value of the vanishing point was plotted as a time series. Since there is an obvious trend in the graph, differential data was applied here. With 2nd derivative the figure of time series clearly shown that most of points are stable while there are two peaks in the left side and right side separately. So far d=2 was obtained.

Then, an autocorrelation factor (ACF) and partial autocorrelation factor (PACF) were created and plotted to identify patterns in the above data which is stationary on both mean and variance. The idea is to identify presence of AR and MA components in the residuals. Through reading the ACF figure, p=1 was obtained since the value after 1 was stable inside of the boundary. Similarly, q=11 since it was the most abnormal one in the PACF figure. Those results together lead to an estimated model: ARIMA(1,2,11).

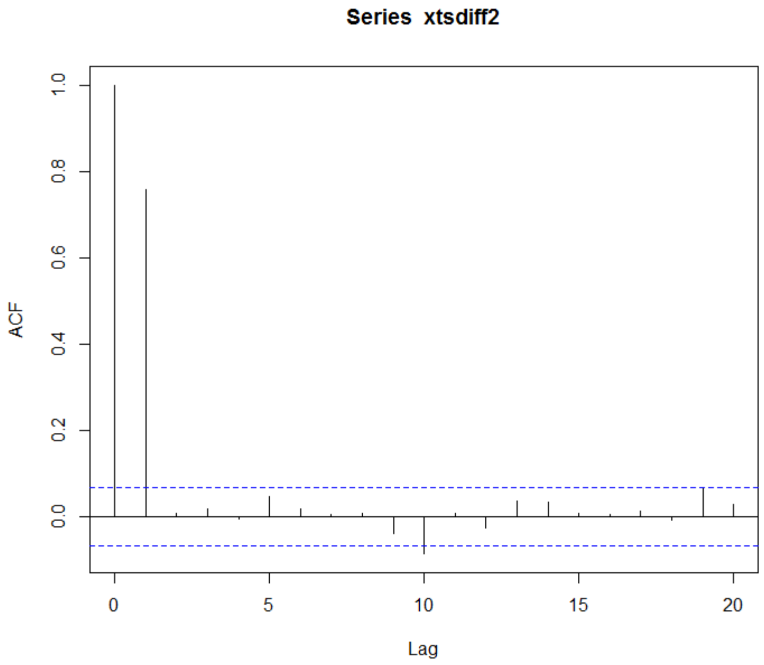


Figure 3.42: ACF graph of vanishing point in Demo3

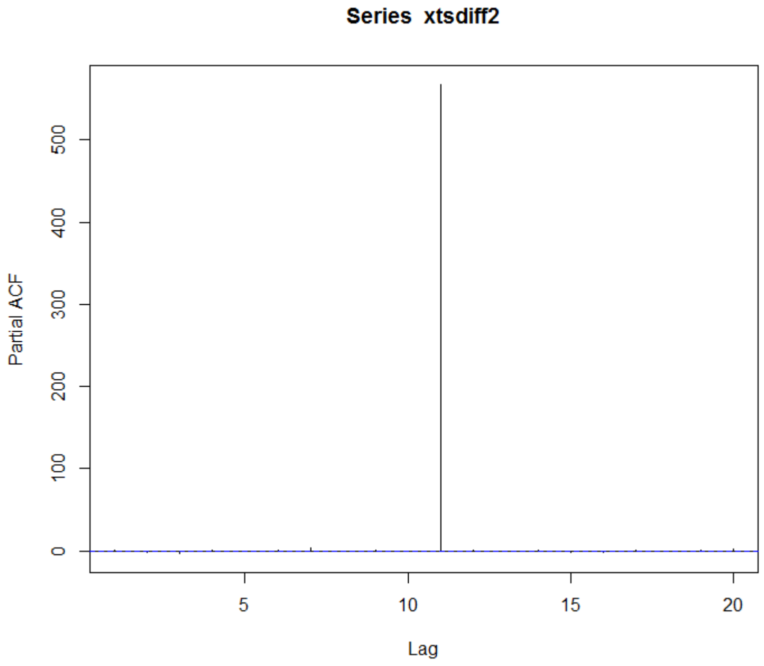


Figure 3.43: PACF graph of vanishing point in Demo3

Figure 3.44: ARIMA-Coefficients

	ar1	ma1	ma2	ma3	ma4	ma5
	-0.0766	-0.3185	-1.5929	-0.2405	1.1371	0.8679
s.e.	NaN	NaN	NaN	NaN	NaN	NaN
	ma6	ma7	ma8	ma9	ma10	ma11
	-0.704	-0.5051	0.164	0.272	0.0082	-0.0883
s.e.	NaN	NaN	NaN	NaN	NaN	NaN

σ^2 estimated as 95.9: log likelihood = -2238.92, aic = 4503.84

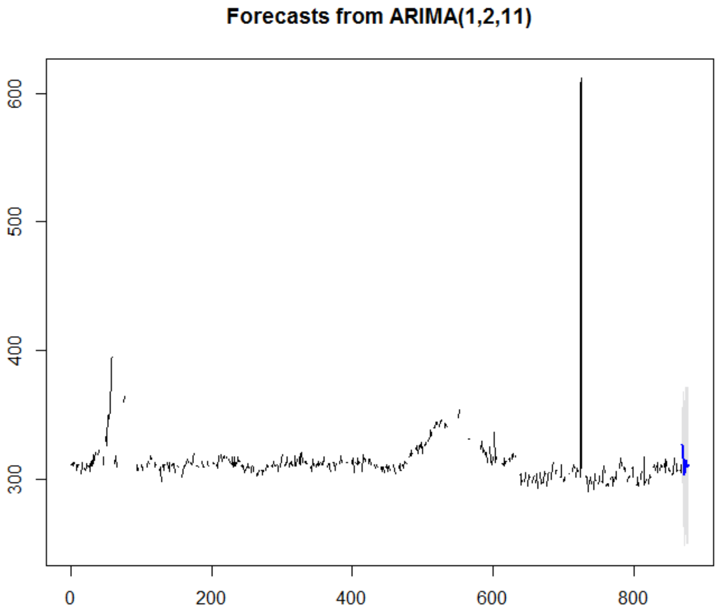


Figure 3.45: Forecast graph of vanishing point in Demo3

Based on this model, future data was predicted in Fig. 3.45, which matched the real data appropriately. Since our assumption about vanishing point has a characteristic of autocorrelation, Kalman Filter was used since it was suitable for the moving object in an inertial system. Beside the vanishing point, the intersection points of the bottom line of image with two closest road lines around ego vehicle were taken into account because as the corner of a triangle, the positions of these three points are related. The algorithm of tracking was shown below: To explain the principle of Kalman Filter, the core formula used were list:

$$X\left(k\right) = AX\left(k-1\right) + BU\left(k\right) + W\left(k\right) \tag{3.11}$$

$$Z\left(k\right) = HX\left(k\right) + V\left(k\right) \tag{3.12}$$

$$X\left(k|k-1\right) = AX\left(k-1|k-1\right) + BU\left(k\right) \tag{3.13}$$

$$Cov\left(k|k-1\right) = ACov\left(k-1|k-1\right)A^{T} + Q \tag{3.14}$$

$$X\left(k|k\right) = X\left(k|k-1\right) + K_{g}\left(k\right)\left(Z\left(k\right) - HX\left(k|k-1\right)\right) \tag{3.15}$$

$$K_{g} = Cov\left(k|k-1\right)H^{T}\left(HCov\left(k|k-1\right)H^{T} + R\left(k\right)\right)^{-1} \tag{3.16}$$

$$Cov\left(k|k\right) = \left(E - K_{g}H\right)Cov\left(k|k-1\right) \tag{3.17}$$

Considering the possible distortions, arrows need to be rotated before obtaining new top points, bottom points, leftmost points and rightmost points, by using the method in Section 3.3.3. Specifically, we may rotate the orientation of the whole arrow in line with the optical axis according to the angle between the middle line of road lane and the middle line of the frame we captured. Similarly, we can easily get the new boundary by calculating the coordinates of the old boundary vertexes.

3.4.2 Depth-based Complex Information Extraction

LIDAR and/or binocular vision system (or inferred camera) was widely used in ADAS fields because of its capability for 3-D reconstruction of the scenario and therefore detected objects in environments around ego vehicle.

Algorithm 3: Triangle Endpoint Tracking using Kalman Filter

```
Input: Detected Triangle Endpoint Position
        $dynamParams$: Dimensionality of the state
        $measureParams$: Dimensionality of the measurement
        $controlParams$: Dimensionality of the control vector
        $type$: Type of the created matrices ($CV\_32F$ or $CV\_64F$)
        $MVP$: Detected Vanishing Point Position
        $MLBP$: Detected Left Bottom Point Position
        $MRBP$: Detected Right Bottom Point Position
Output: Estimated Triangle Endpoint Position
        $PVP$: Predicted Vanishing Point Position
        $PLBP$: Predicted Left Bottom Point Position
        $PRBP$: Predicted Right Bottom Point Position
        $EVP$: Estimated Vanishing Point Position
        $ELBP$: Estimated Left Bottom Point Position
        $ERBP$: Estimated Right Bottom Point Position

if First Time Detected VP, LBP, RBP then
    $KF\_VP.statePre = (320, 240, 0, 0)$;
    /* predicted state $(X'_k) : X_k = AX_{k-1} + BU_k$ */
    $KF\_LBP.statePre = (MLBP.x, MLBP.y, 0, 0)$;
    $KF\_RBP.statePre = (MRBP.x, MRBP.y, 0, 0)$;
    for $i$ = VP,LBP,RBP do
        $KF\_(i).transitionMatrix = (1,0,1,0; 0,1,0,0; 0,0,1,0; 0,0,0,1)$;
        /* state transition matrix (A) */
        $setIdentity(KF\_(i).measurementMatrix)$;
        /* measurement matrix (H) */
        $setIdentity(KF\_(i).processNoiseCov)$;
        /* process noise covariance matrix (Q) */
        $setIdentity(KF\_(i).measurementNoiseCov)$;
        /* measurement noise covariance matrix (R) */
        $setIdentity(KF\_(i).errorCovPost)$;
        /* posteriori error estimate covariance matrix
            $(P_k) : P_k = (I - K_k H)P'_k$ */
else
    predict:
        $PVP = KF\_VP.predict()$;
        $PLBP = KF\_LBP.predict()$;
        $PRBP = KF\_RBP.predict()$;
    update:
        $EVP = KF\_VP.correct(MVP)$;
        $ELBP = KF\_LBP.correct(MLBP)$;
        $ERBP = KF\_RBP.correct(MRBP)$;
```

Unfortunately, for the monocular vision system, normally it was hard to conduct such observations. Since the single camera can only catch a 2-D snapshot of the object on a focal plane without depth information, many methods were investigated to help grab depth information. For instance, shade information in the image can be used to grab the depth of normal information of the object surface. An update of Shape-of-shading method, Photometric Stereo are used to solve the depth information by calculating images taken in different lighting conditions. Moreover, Shape-from-texture method was applied when an object with a smooth surface covered by replicated texture units, and the projection of the object from 3D to 2D causes distortion and perspective.

The method applied here was based on the man-made construction, specifically the features of the road, like width and shape. The road itself was treated as a ruler, which enables us to build a 3-D coordinate system among the objects which interacted with the road (like vehicles on the road or poles on the pavement).

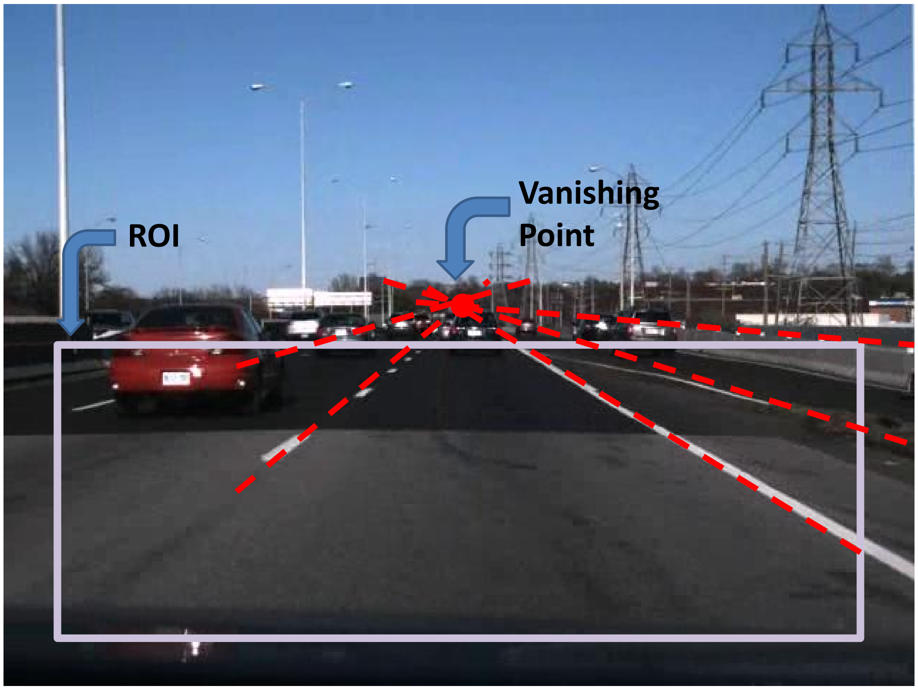


Figure 3.46: ROI and vanishing point: the red point where all lines intersect is the vanishing point; the rectangular region which specifies the detection range is referred as the ROI

As is discussed in the previous two chapters, we can get X-Z plane based on the road surface detected by us through using vanishing point and inverse perspective transform. Moreover, objects above the road surface, like vehicles, will be detected and bounded by hexahedrons with an appropriate position above X-Z plane. Any other objects which are far from the road on either side or without obvious features will be treated as 2-D patterns and projected as two slides of the Y-Z plane corresponded to the left-side view and right-side view respectively. Admittedly, such projections will lead to a distortion of the objects in two sides (e.g. Hills in the distance or vehicles not detected).

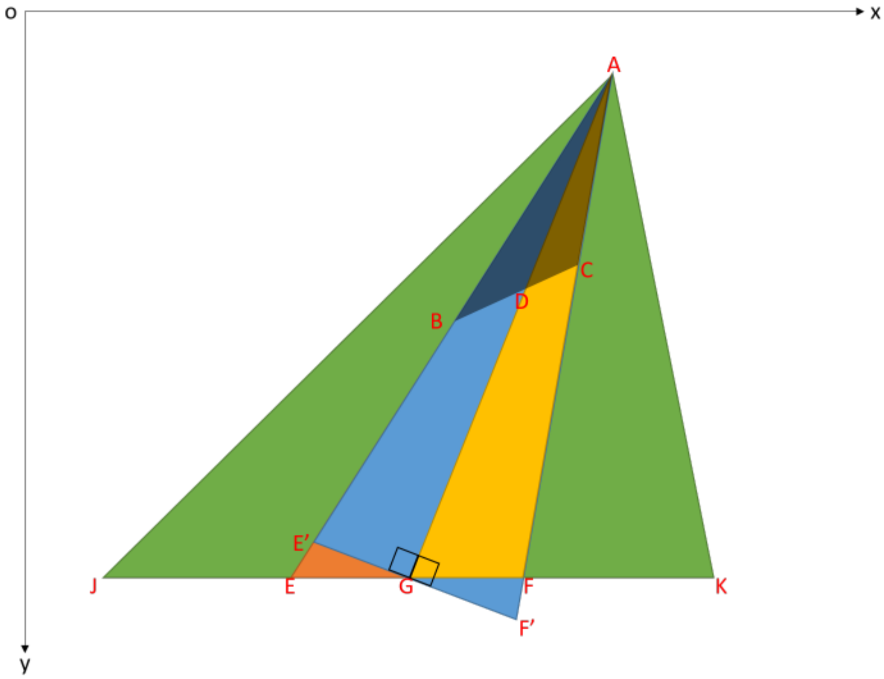


Figure 3.47: Road network model

In Fig. 3.47, point A represents the vanishing point. B and C represent the bottom point of the road lines close to ego vehicle. Based on this information, AD, as a virtual middle line of the road lane we are driving on and point G, the intersection of AD and bottom of the image may be obtained through the following formula:

1. Calculate the intersection G of bisector of top angle A and bottom line BC

$$AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \tag{3.18}$$

$$AC = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2} \tag{3.19}$$

$$\lambda = \frac{AB}{AB + AC} \tag{3.20}$$

$$x_G = (1 - \lambda)x_B + \lambda x_C \tag{3.21}$$

$$y_G = (1 - \lambda)y_B + \lambda y_C \tag{3.22}$$

2. Calculate formula of the line bisector of top angle A

$$\frac{y - y_G}{y_A - y_G} = \frac{x - x_G}{x_A - x_G} \tag{3.23}$$

3. Calculate the cross point G of bisector of top angle A and bottom line BC

$$\begin{cases} x_G = \frac{y_C(x_C - x_A)}{y_A - y_C} + x_C \\ y_G = 0 \end{cases} \tag{3.24}$$

With respect to perspective view, EG, as the distance from E to G is not equal to GF, the distance from G to F. To get a correct road lane width, two points E and F are added to help calculation. Points E and F are symmetric about line virtual middle line AG. The formulas related are listed below:

4. Calculate formula of the line pass point G and perpendicular to line AG

$$bx - ay - bx_G + ay_G = 0 \tag{3.25}$$

5. List simultaneous equations and calculate the position of left Corner Point E and right Corner Point F

$$\begin{cases} bx - ay - bx_G + ay_G = 0; \\ ax + by + c + d\sqrt{a^2+b^2} = 0; \end{cases} \Rightarrow (x_0, y_0) \tag{3.26}$$

$$\begin{cases} bx - ay - bx_G + ay_G = 0; \\ ax + by + c - d\sqrt{a^2+b^2} = 0; \end{cases} \Rightarrow (x_0^{'}, y_0^{'}) \tag{3.27}$$

$$(y_A - y_G)x - (x_A - x_G)y - (y_A - y_G)x_G + (x_A - x_G)y_G = 0 \tag{3.28}$$

$$\begin{cases} x_E = \dfrac{-ac - ad\sqrt{a^2+b^2} + b^2x_G - aby_G}{a^2+b^2}; \\ \\ y_E = \dfrac{-bc - bd\sqrt{a^2+b^2} - abx_G + a^2y_G}{a^2+b^2}; \end{cases} \tag{3.29}$$

$$\begin{cases} x_F = \dfrac{-ac + ad\sqrt{a^2+b^2} + b^2x_G - aby_G}{a^2+b^2}; \\ \\ y_F = \dfrac{-bc + bd\sqrt{a^2+b^2} - abx_G + a^2y_G}{a^2+b^2}; \end{cases} \tag{3.30}$$

6. Calculate formula of the line pass point A and E, line pass point A and F

$$\begin{cases} \frac{y-y_E}{y_A-y_E} = \frac{x-x_E}{x_A-x_E} \quad \Rightarrow LineAE \\ \\ \frac{y-y_F}{y_A-y_F} = \frac{x-x_F}{x_A-x_F} \quad \Rightarrow LineAF \end{cases} \tag{3.31}$$

7. Draw lines as reference network for compare with key points of object of interest on the road

$$\begin{cases} x_{E'} = x_E + \frac{(x_A-x_E)\times(y_{E'}-y_E)}{y_A-y_E} \\ \\ y_{E'} = 479 \end{cases} \tag{3.32}$$

$$\begin{cases} x_{F'} = x_F + \frac{(x_A-x_F)\times(y_{F'}-y_F)}{y_A-y_F} \\ \\ y_{F'} = 479 \end{cases} \tag{3.33}$$

So far we have the formula of the closest road lines and the width of the road lane in the image. According to the standard of construction that the width of lanes is approximately the same, it is possible to draw the rest road lines (e.g. line AG and line AK) on both sides.

8. Calculate the line in two sides of current lane

$$\begin{cases} q_1 = Ax + By + Au + Bv + 2C \\ \\ q_2 = Bx - Ay - Bu + Av \end{cases} \tag{3.34}$$

$$\begin{cases} x_J = \frac{-2AC - A^2 x_B + B^2 x_B - 2AB y_B}{A^2 + B^2} \\ \\ y_J = \frac{-2BC + A^2 y_B - B^2 y_B - 2AB x_B}{A^2 + B^2} \end{cases} \tag{3.35}$$

$$\begin{cases} x_K = \frac{-2AC - A^2 x_C + B^2 x_C - 2AB y_C}{A^2 + B^2} \\ \\ y_K = \frac{-2BC + A^2 y_C - B^2 y_C - 2AB x_C}{A^2 + B^2} \end{cases} \tag{3.36}$$

Last but not the least, the lines drawn in the last step should be calibrated by least square method and compared to the pixels of interest with an extended region based on our estimation. Through this, a road network model which contains distance information can be built. This model can also provide additional information for further research like car detection, pedestrian detection and so on.

Chapter 4

Experimental Results

This chapter is organized into 3 sections, which cover the experimental platform and overview, performance evaluation methodology, as well as detection and recognition results performance, respectively. Emphasis will be paid to the last section for presenting the performance evaluation on the proposed module.

4.1 Experimental Platform and Overview

In our experiment, we installed the camera close to the front windshield of the vehicle. As a result, the lower part of the image captured by the camera was covered by the hood. The original resolution of the camera was 1920*1080. With the purpose of reducing the computational cost as well as increasing generalization of our method, we transcoded the video file at a resolution of 640*480 and cut superfluous parts in two sides with respect to the natural aspect ratio.

The camera, tripod and laptop together make up the test bed for our system. Through this system, real-time feedback is available and both the original and processed images, as well as the log file of the system were stored in the hard drive of the laptop. Details of the camera and laptop used in our experiment are listed in tables below:

Figure 4.1: The Testbed used to test our system

Table 4.1: Camera Parameters

Camera attribute	Camera parameters
Company	SONY
Model No.	HDR-PJ710V
Image Sensor	Back-illuminated ”Exmor R” CMOS Sensor
Image Sensor Size	Diagonal 1/2.88”, 6.3mm
Maximum Resolution	1920*1080 at 60FPS
Internal Memory	32GB
Image Processor	BIONZ
System	TTL contrast detection
Color Filter System	ClearVid array
Digital Video Format	AVCHD, MPEG-2
Image Stabilizer	Optical

Table 4.2: Lens Parameters

Lens attribute	Lens parameters
Company	Carl Zeiss
Model No.	Vario-Sonnar T*
Focal Length (mm)	3.8-38.0
Filter diameter	52 mm (2 1/8 in.)
Aperture (F/#)	1.8-3.4
Color temperature	3,200K
Optical Zoom	10x
Focus Distance (cm)	1.0-32.0
Aperture Blade	6 blades

Table 4.3: Computer Parameters

Hardware attribute	Hardware parameters
Computer Model	Thinkpad X201
CPU Model	Intel Core i5-520M @ 2.40GHz dual core
GPU Model	Intel GMA HD
Memory	4 GB
Operating System	Windows10 Pro
OpenCV version	v2.48

4.1.1 Camcorder System

The video capture system we used in our experiment is the Sony HDR-PJ710V. As a High Definition Handycam Camcorder, it shoots 1920* 1080 Full HD 60p/24p video and takes 24.1 megapixel still images. It features 32GB of embedded flash memory, which provides up to eleven hours of seamless, continuous recording. Additionally, the camcorder is equipped with a back-illuminated Exmor CMOS sensor which is ideal for low-light situations. Sony has outfitted the HDR-PJ710V with a 26mm wide angle Carl Zeiss Vario-Sonnar T Lens that supports 10x optical zoom and 17x extended zoom. It also comes with expanded audio headphone and geotagging with an integrated GPS receiver and NAVTEQ maps.

On standard SteadyShot camcorders a single ”shift lens” moves within the optical block to offset camera shake. Balanced Optical SteadyShot image stabilization moves the entire optical block at once, maintaining the integrity of the optical path and improving shake reduction by up to thirteen times. It also reduces vignetting (light fall-off towards the edge of a scene) and lowers distortion in videos and still photos.

Figure 4.2: Sony HDR-PJ710V

4.1.2 Real-life Video Collection

Generally, we considered both day-time and night-time situations. Seven videos were collected from highway and urban roads in Ottawa during day-time, and three videos from urban areas of Ottawa at night. These videos represent common scenarios with different weather conditions, road surfaces, lighting conditions, and traffic density, which people might encounter in real-life situations.

10 clips were selected as examples. As shown in Fig. 4.3 and Table 4.4, Clip #1 and Clip #2 were collected on Highway 417. The black line in Fig. 4.3 is the only specific HOV lane in Ottawa, according to the Ontario Ministry of Transportation. Among these two, Clip #1 represents the westbound segment while Clip #2 represents the eastbound segment. To ensure we had enough frames to analyze, we set the camera to 50 FPS mode to adapt the speed limit of 100 km/h.

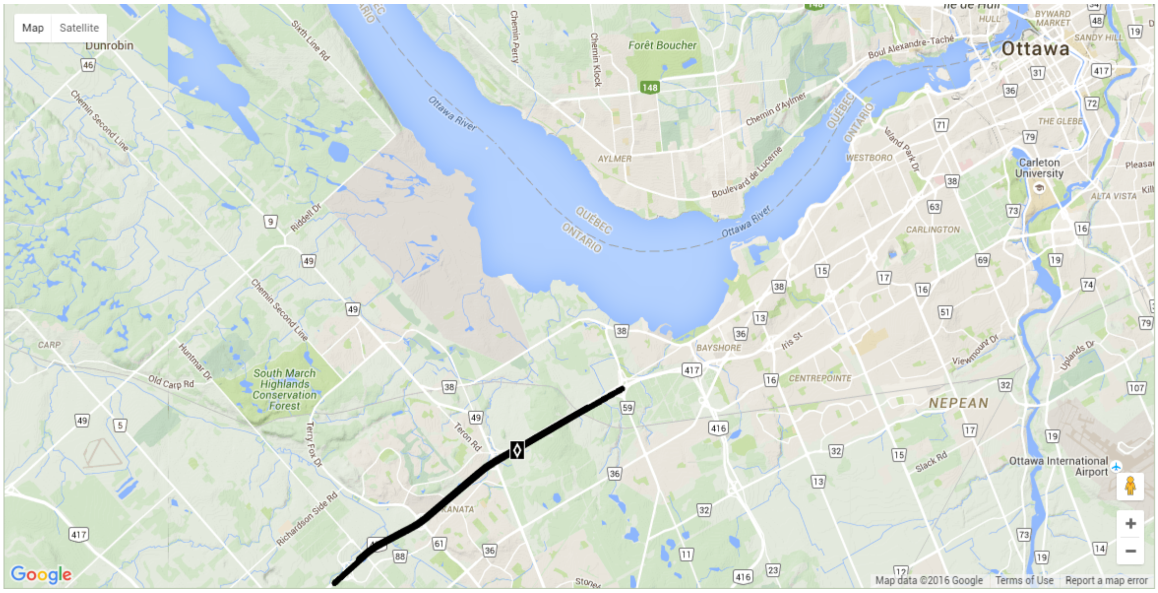


Figure 4.3: HOV Lane Map

Figure 4.4: Snapshot of HOV clips

Despite the presence of HOV lanes included in Clip #1 and Clip #2,several arrows in

Table 4.4: Video Clips for HOV testing

	Clip #1	Clip #2
Weather	Cloudy	Cloudy
Location	Highway	Highway
Traffic condition	Light	Light
Road surface	Flat and Smooth	Flat and Smooth
Frame NO.	13645	12229
Frame speed	50fps	50fps
Markings type	HOV	HOV
Number of frames containing departure	2	2

Ottawa-Gatineau Area were chosen for test. However, we found that the signs in Gatineau are different to the ones in Ottawa, as the traffic rules in Quebec and Ontario are not identical with each other. Therefore, only the clips collected in Ottawa were retained. Moreover, we have also excluded some clips with noticeably low quality (e.g., part missing or blurring of markings, temporary signage which is not accordance with related rule, see Fig. 4.5).

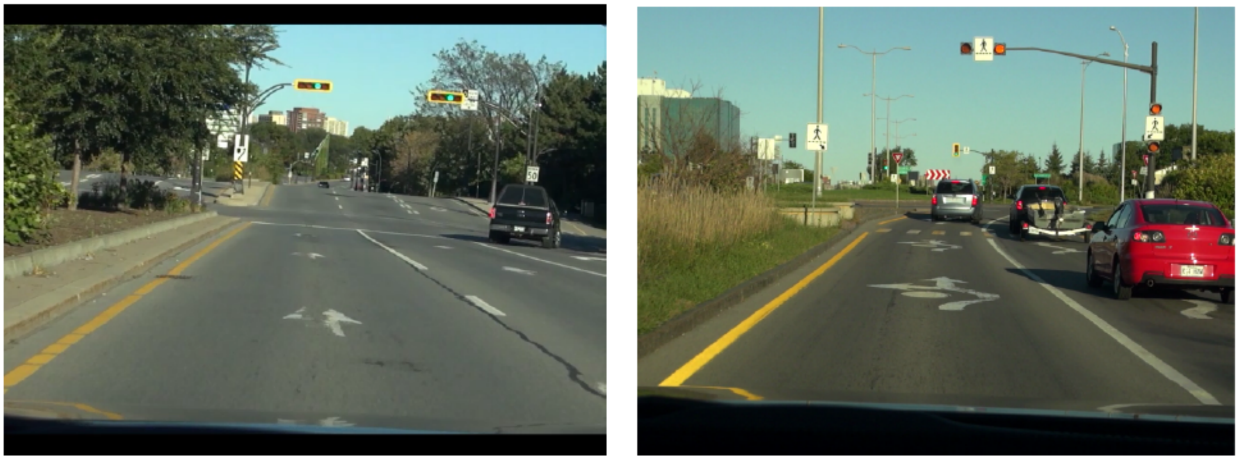
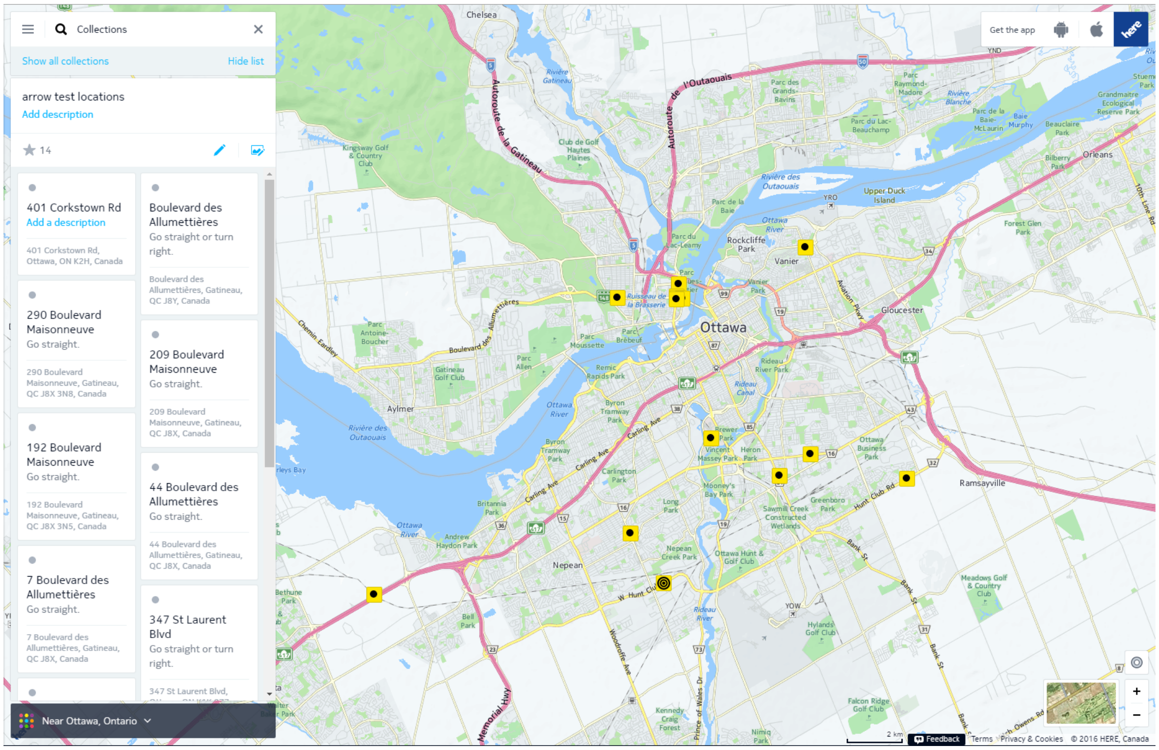


Figure 4.5: Samples of clips excluded in our experiment



arrow test locations
Add description

★ 14

401 Corkstown Rd
Add a description
401 Corkstown Rd, Ottawa, ON K2H, Canada

Boulevard des Allumettières
Go straight or turn right.
Boulevard des Allumettières, Gatineau, QC J8Y, Canada

290 Boulevard Maisonneuve
Go straight.
290 Boulevard Maisonneuve, Gatineau, QC J8X 3N8, Canada

209 Boulevard Maisonneuve
Go straight.
209 Boulevard Maisonneuve, Gatineau, QC J8X, Canada

192 Boulevard Maisonneuve
Go straight.
192 Boulevard Maisonneuve, Gatineau, QC J8X 3N5, Canada

44 Boulevard des Allumettières
Go straight.
44 Boulevard des Allumettières, Gatineau, QC J8X, Canada

7 Boulevard des Allumettières
Go straight.
7 Boulevard des Allumettières, Gatineau, QC J8X, Canada

347 St Laurent Blvd
Go straight or turn right.
347 St Laurent Blvd,

Near Ottawa, Ontario

Figure 4.6: Arrows Test Map

4.2 Performance Evaluation Methodology

Sensitivity, specificity, and accuracy, as well as precision and F-Score, are introduced as statistical measures of the performance of our proposed system, which suits the assumptions of a binary classification test.

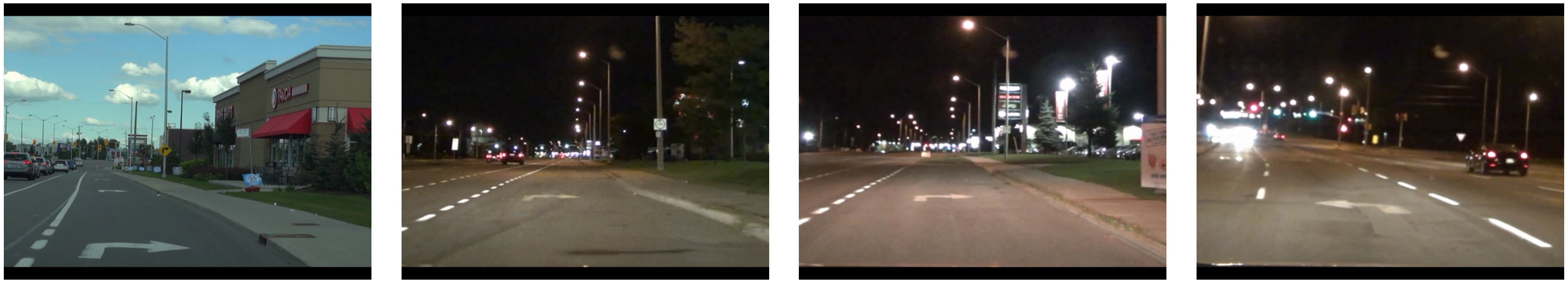
Sensitivity (also called the true positive rate, or the recall in some fields) measures the proportion of positives that are correctly identified as such (e.g., the percentage of sick people who are correctly identified as having the condition).

$$Sensitivity(TPR) = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Negatives} \tag{4.1}$$

$$= \frac{Number\ of\ True\ Positives}{Total\ Number\ of\ Signs} \tag{4.2}$$

$$= Probability\ of\ a\ positive\ test\ of\ the\ sign\ appealed \tag{4.3}$$

Clip #3 Clip #4 Clip #5 Clip #6



Clip #7 Clip #8 Clip #9 Clip #10

Figure 4.7: Snapshot of arrow clips

Figure 4.8: Video Clips for Arrows testing

	Clip #3	Clip #4	Clip #5	Clip #6	Clip #7	Clip #8	Clip #9	Clip #10
Weather	Cloudy	Cloudy	Sunny	Cloudy	Sunny	Night	Night	Night
Location	Rural	Rural	Rural	Rural	Rural	Rural	Rural	Rural
Traffic condition	Light	Light	Light	Light	Light	Light	Light	Light
Road surface	Flat and Smooth	Rough	Flat and Smooth	Flat and Smooth	Flat and Smooth	Rough	Rough	Rough
Frame NO.	960	1081	673	720	480	839	1441	384
Frame speed	24fps	24fps	24fps	24fps	24fps	24fps	24fps	24fps
Lane markings/frame	1	1	1-2	1	1	1	1	1-2
Markings type	Left	Right	Left	Left	Right	Right	Right	Left

Specificity (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g., the percentage of healthy people who are correctly identified as not having the condition).

$$Specificity(SPC) = \frac{Number\ of\ True\ Negatives}{Number\ of\ True\ Negatives + Number\ of\ False\ Positives} \tag{4.4}$$

$$= \frac{Number\ of\ True\ Negatives}{Total\ Number\ of\ Frames\ contains\ No\ Sign} \tag{4.5}$$

$$= Probability\ of\ a\ Negative\ test\ of\ Frames\ contains\ No\ Sign \tag{4.6}$$

As shown in the table above, we gain the relationship between variables listed.

Table 4.5: Relationship between indexes

	True	**False**	
Positive	TP	FP	P=TP+FP
Negative	TN	FN	N=TN+FN
	T=TP+TN	F=FP+FN	
	$TPR = \frac{TP}{TP+FN}$	$SPC = \frac{TN}{TN+FP}$	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
	$Precision = \frac{TP}{TP+FP}$	$Recall = TPR$	$F\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$

$$Accuracy(ACC) = \frac{\Sigma\ True\ Positive + \Sigma\ True\ Negative}{\Sigma\ Total\ Population} \tag{4.7}$$

Precision is the probability that a (randomly selected) retrieved document is relevant.

$$Precision(PPV) = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Positives} \tag{4.8}$$

$$= \frac{Number\ of\ True\ Positives}{Total\ Number\ of\ Frames\ retrieved} \tag{4.9}$$

$$= Probability\ of\ a\ Positive\ test\ of\ Frames\ contains\ Sign \tag{4.10}$$

A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$F_\beta = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{4.11}$$

$$= \frac{2 \times Precision \times Recall}{Precision + Recall}\ (When\ \beta = 1) \tag{4.12}$$

This measure is approximately the average of the two when they are close, and is more generally the square of the geometric mean divided by the arithmetic mean. There are several reasons that the F-score can be criticized in particular circumstances due to its bias as an evaluation metric. Specifically in formula (4.12) $F_\beta = F_1$. This is also known as the F_1 measure, because recall and precision are evenly weighted.

4.3 Markings Detection and Recognition Results Performance

Using the valuation methodology introduced in the previous section, several tables and charts are obtained, which are listed below sequentially for vertical and horizontal comparison. Overall, our proposed algorithm achieved an average accuracy rate of 96.77% while F_1 Score also achieved 90.57%.(See Fig. 4.30)

Before analysis by statistical methods, certain rules were introduced in the data cleaning process. Since missing data has no influence in a continuous series of detected frames, this means the few false negative frames cannot be observed by driver because their presence does not impact the reaction of the true positives preceding or following the false negatives.) For instance, if 10 frames containing an HOV sign were evaluated while only frames No.5-No.9 are true positives, the criteria will be loosened up in No.1-No.4 and No.10 if they are all false negatives and the total duration of which is less than 1 s.(Compared with the traditional reaction time of a typical driver, 0.5s).

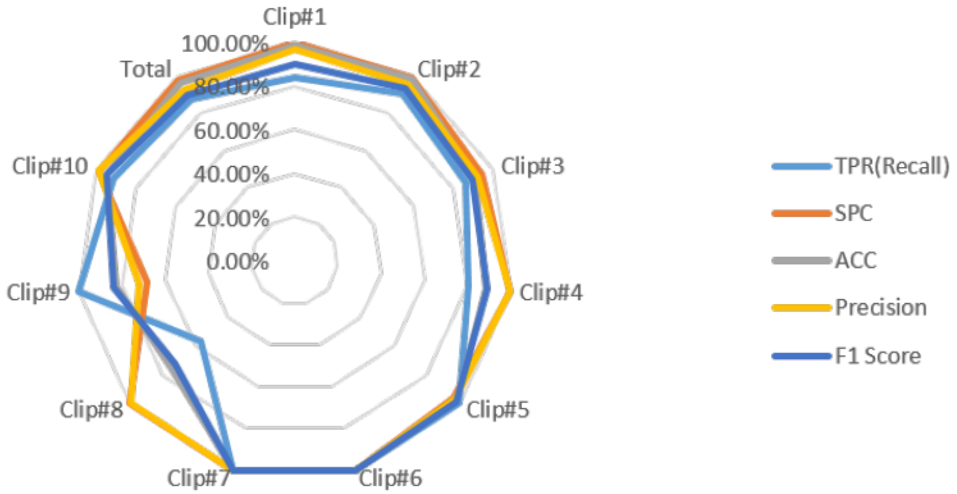


Figure 4.9: Statistical Indicators between Clips

To rank a number of variables including TPR, SPC, ACC, precision and F_1 Score of 10 clips, a radar chart was created (See Fig. 4.9). Each of the 10 clips form individual axes which have been arranged radially around a point. The value of each aspect is depicted by the node (anchor) on the spoke (axis). The lines of each statistical indicator are drawn connecting the data values for each spoke. The radar chart shows that the proposed algorithm achieved good results in Clip #1, Clip #2, Clip #3, Clip #4, Clip #5, Clip #6, Clip #7 and Clip #10. However, the performance in Clip #8 and Clip #9 was abnormal compared with other clips. Specifically, the precision rate in Clip #8 is high while TPR is

Table 4.6: Markings Results Table

	Clip#1	**Clip#2**	**Clip#3**	**Clip#4**	**Clip#5**	**Clip#6**	**Clip#7**	**Clip#8**	**Clip#9**	**Clip#10**	**Total**
TP	904	1007	362	514	368	290	471	276	651	187	5030
TN	12536	11083	511	444	293	430	9	351	540	177	26374
FP	29	41	29	0	12	0	0	1	249	2	363
FN	176	98	58	123	0	0	0	211	1	18	685
TPR	83.70%	91.13%	86.19%	80.69%	100.00%	100.00%	100.00%	56.67%	99.85%	91.22%	88.01%
SPC	99.77%	99.63%	94.63%	100.00%	96.07%	100.00%	100.00%	99.72%	68.44%	98.88%	98.64%
ACC	98.50%	98.86%	90.94%	88.62%	98.22%	100.00%	100.00%	74.73%	82.65%	94.79%	96.77%
Precision	96.89%	96.09%	92.58%	100.00%	96.84%	100.00%	100.00%	99.64%	72.33%	98.94%	93.27%
Recall	83.70%	91.13%	86.19%	80.69%	100.00%	100.00%	100.00%	56.67%	99.85%	91.22%	88.01%
F1 Score	89.82%	93.54%	89.27%	89.31%	98.40%	100.00%	100.00%	72.25%	83.89%	94.92%	90.57%

Table 4.7: Processing Time Results Table

	Clip#1	**Clip#2**	**Clip#3**	**Clip#4**	**Clip#5**	**Clip#6**	**Clip#7**	**Clip#8**	**Clip#9**	**Clip#10**	**50FPS**	**24FPS**
Time Consume (ms)	1420901	1276763	107319	89896	72237	66832	45107	75287	140807	39339	2697664	636824
Time Consume (s)	1420.90	1276.76	107.32	89.90	72.24	66.83	45.11	75.29	140.81	39.34	2697.66	636.82
True Time Duration (s)	272.90	244.58	40.00	45.04	28.04	30.00	20.00	34.96	60.04	16.00	517.48	274.08
Process FPS	9.60	9.58	8.95	12.03	9.32	10.77	10.64	11.14	10.23	9.76	9.59	10.33
Origin FPS	50	50	24	24	24	24	24	24	24	24	50	24
Delay (ms/ Frame)	84.13	84.40	70.12	41.49	65.67	51.16	52.31	48.07	56.05	60.78	84.26	55.14

obviously low. On the other hand, TPR of Clip #9 is high but SPC of it as well as precision is low. To explore the reason for these anomalies, analysis of each clip is presented in the following section.

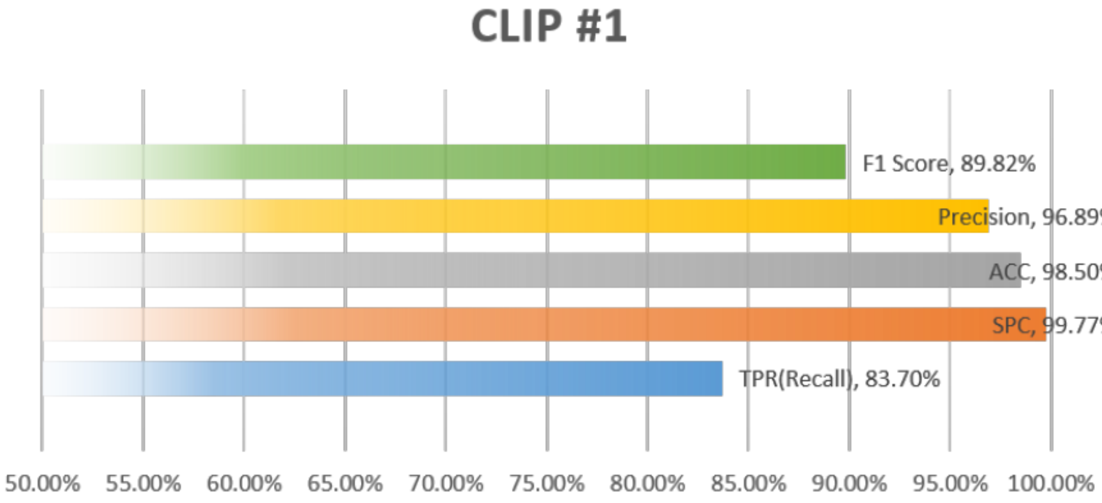


Figure 4.10: Statistical Results of HOV, Clip #1

As shown in Fig. 4.10, precision, accuracy and specificity all achieved a rate above 95%, which means among all detection results, only a few objects/frames were false detected. However, TPR of Clip #1 was as low as 83.70%, indicating that although we have recognized detected objects correctly, a certain number of frame containing objects were undetected. After reviewing the experimental data in Fig. 4.11, we reasoned that a HOV sign (as same as the one shown as sample of FN in Fig. 4.11) contributed the most to FN since the top point of this HOV symbol was eroded.

As shown in Fig. 4.12, precision, accuracy and specificity all achieved a rate above 95% like Clip #1. However, TPR of Clip #2 as low as 91.13%, indicates that although we have recognized detected objects correctly, a certain number of frames containing objects were missed. After reviewing experimental data in Fig. 4.12, two reasons contributed the most significantly to this. Firstly, the isolation belt contained parallelograms (similar to the one shown as a sample of FP in Fig. 4.11) which were easily mistaken for the HOV symbol.

TP TN

FP FN

Figure 4.11: Samples of Clip #1

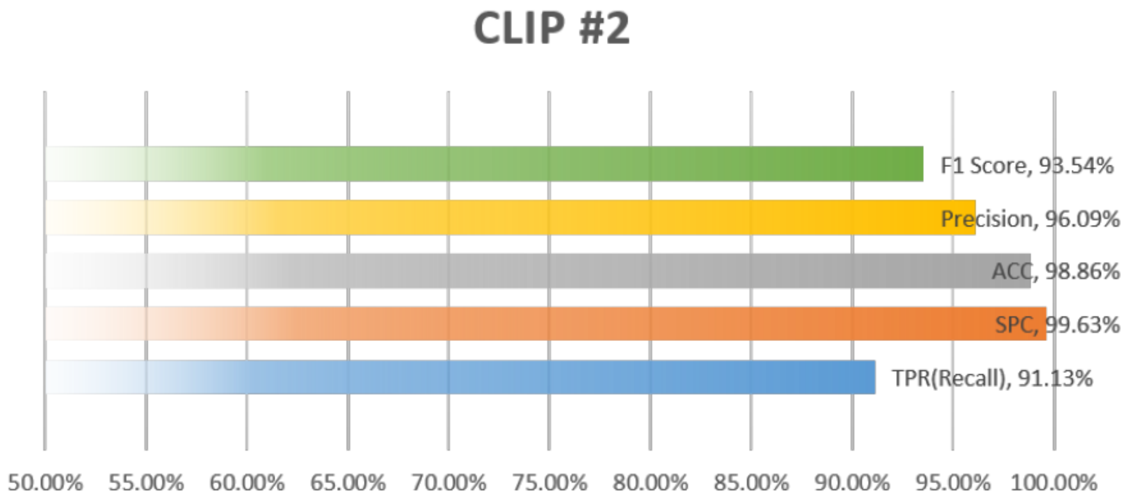


Figure 4.12: Statistical Results of HOV, Clip #2

TP TN

FP FN

Figure 4.13: Samples of Clip #2

Secondly, since the ROI is in close proximity of the ego vehicle, some markings out of range in front of vehicle were not taken into consideration.

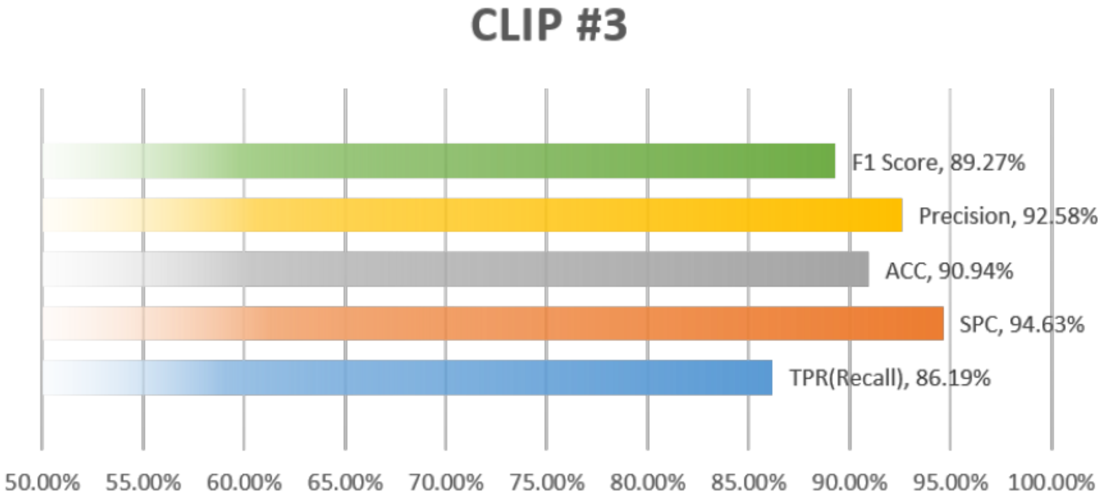


Figure 4.14: Statistical Results of Arrows, Clip #3

Due to cloudy weather, the average performance of Clip #3 was not as high as in Clip #1 and Clip #2. As shown in Fig. 4.14, precision, accuracy and specificity all achieved a rate between 90% and 95%. TPR of Clip #3 as low as 86.19%, indicating that a number of frames containing objects were missed. After reviewing experimental data in Fig. 4.14, two contributing reasons were found. Firstly, lane merging conditions exist in this clip, which led to disorientation of the proposed system. Secondly, similar to Clip #2, some markings out of range immediately in front of the vehicle were not taken into consideration.

As shown in Fig. 4.16, precision and specificity both achieved a rate of 100%, which means all detection results were properly recognized. However, Accuracy of Clip #4 as 88.62%, indicating that there are a certain number of false positives or false negatives. Considering that both the precision and specificity results implied the FP of this clip is 0, it is most likely FN is the main reason. TPR of Clip #4 was as low as 80.69%, thus verifying our hypothesis. It also indicates that some of the frames containing arrow object were missed. After a review of experiment data in Fig. 4.17, two reasons contributed the most

TP TN

FP FN

Figure 4.15: Samples of Clip #3

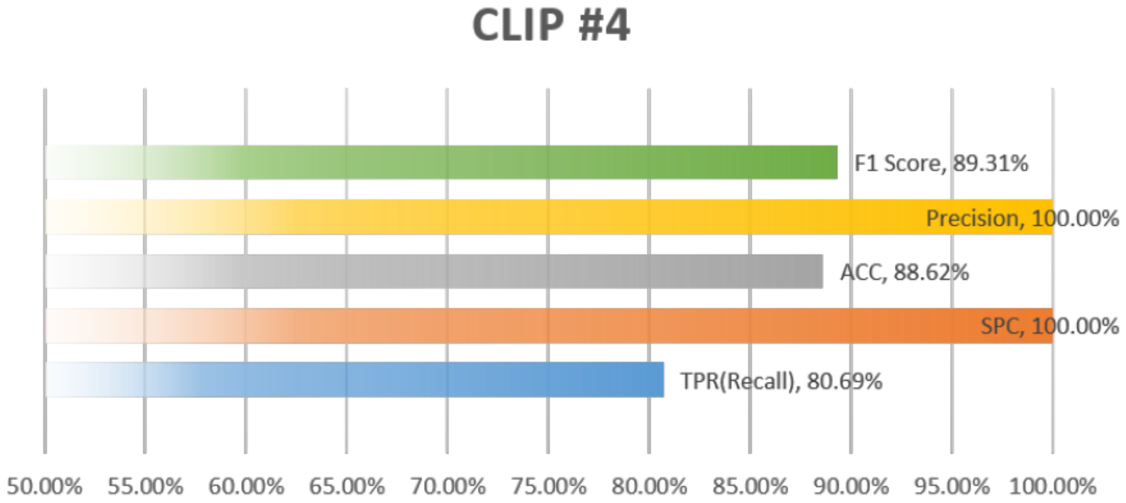


Figure 4.16: Statistical Results of Arrows, Clip #4

TP

TN

N/A
FP

FN

Figure 4.17: Samples of Clip #4

to this. First and most important, poorly maintained markings lost their contrast and turned blurry (similar to the one shown as an example of FN in Fig. 4.17. Secondly, some markings out of range in front of the ego vehicle were not taken into consideration.

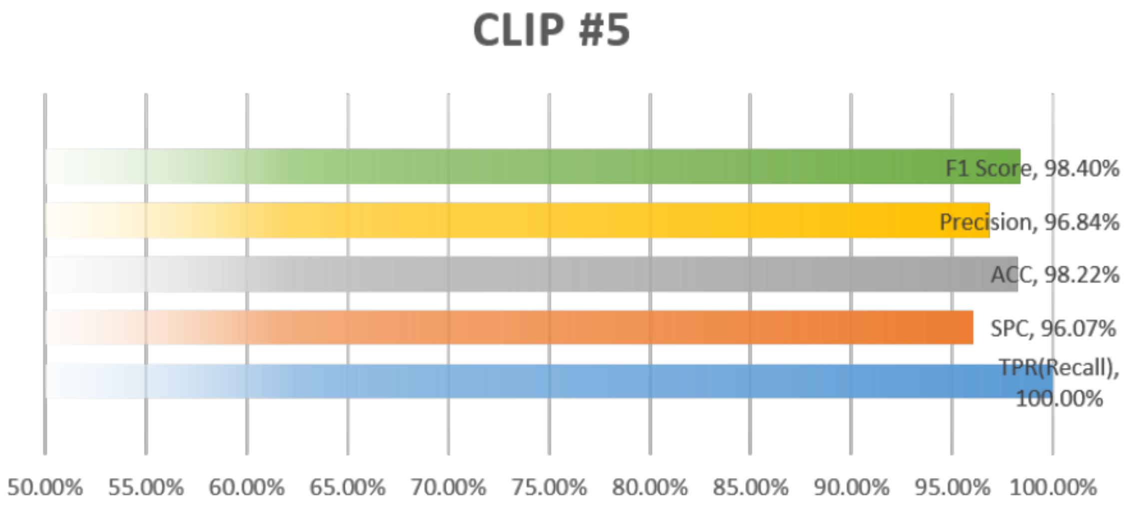
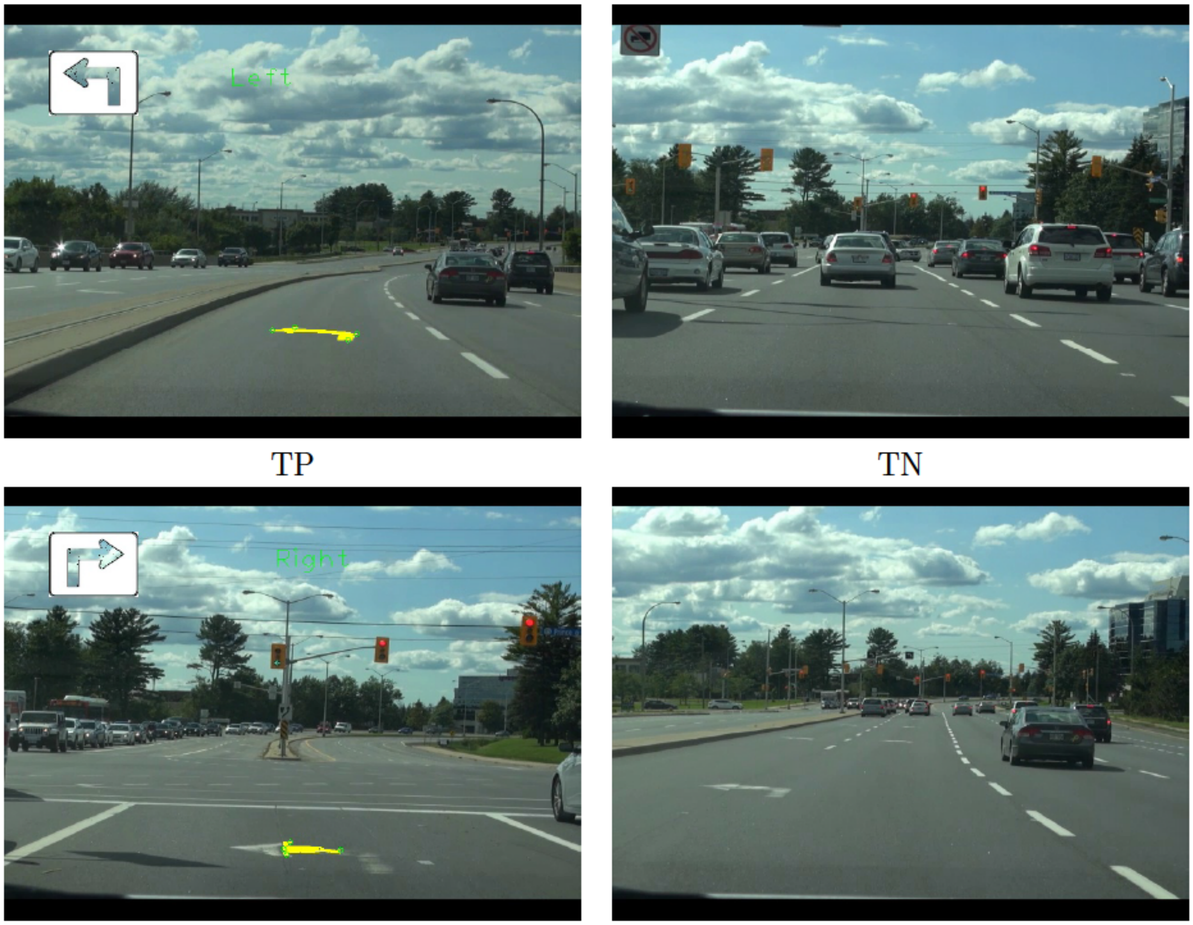


Figure 4.18: Statistical Results of Arrows, Clip #5

As shown in Fig. 4.18, precision, accuracy and specificity all achieved a rate above 95%, meaning almost all detection results were properly recognized. In particular, TPR of Clip #5 got a rate above 100%, which indicates that all frames containing arrow objects were properly recognized. After a review of experimental data in Fig. 4.19, the main reason for this minor defect is that poorly maintained markings lost contrast and turned blurry (similar to the one sample of FP in Fig. 4.19.

As shown in Fig. 4.20, precision, accuracy, specificity and TPR all achieved a rate of 100%, which means all detection results were properly recognized and all frames containing arrow objects were properly recognized in Clip #6. After a review of experimental data in Fig. 4.21, some sample snapshots of TP and TN are provided accordingly.

As shown in Fig. 4.22, the performance of Clip #7 is similar to Clip #6s. Specifically, precision, accuracy, specificity and TPR of Clip #7 all achieved a rate of 100%, which



TP

TN

FP

FN

Figure 4.19: Samples of Clip #5

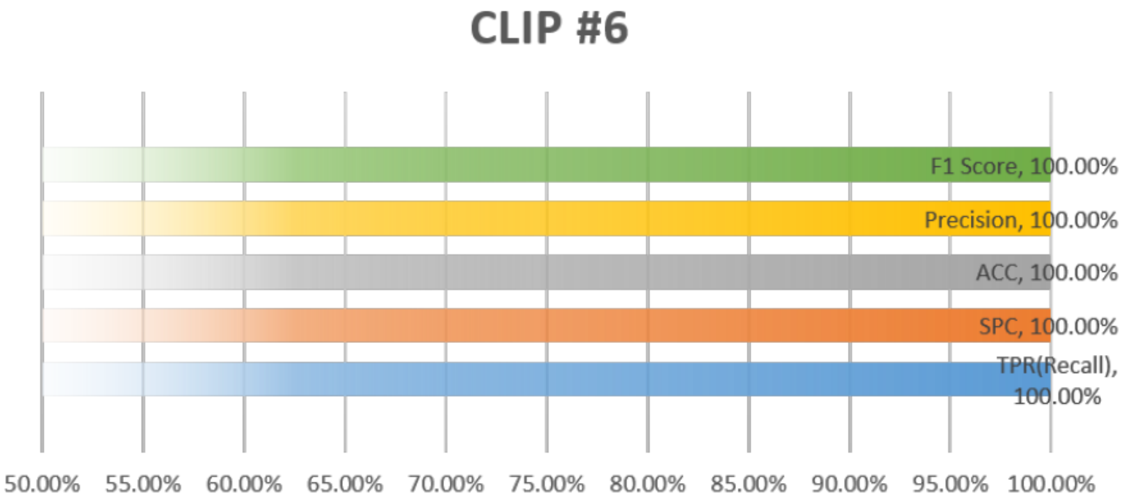


Figure 4.20: Statistical Results of HOV, Clip #6

TP TN

Figure 4.21: Samples of Clip #6

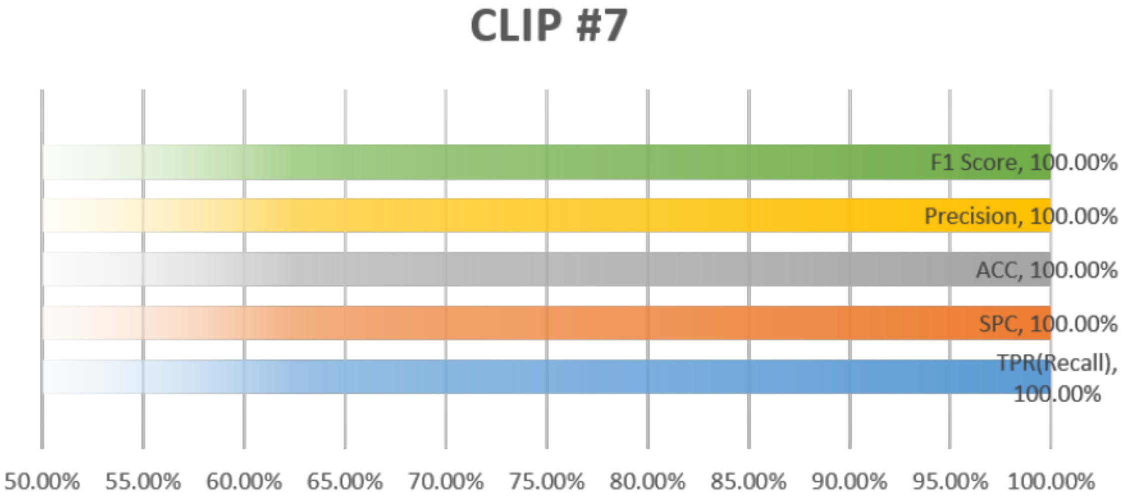


Figure 4.22: Statistical Results of Arrows, Clip #7

TP TN

Figure 4.23: Samples of Clip #7

means all detection results were properly recognized and all of the frames containing arrow objects were properly recognized. After a review of experimental data in Fig. 4.23, some sample snapshots of TP and TN are provided accordingly.

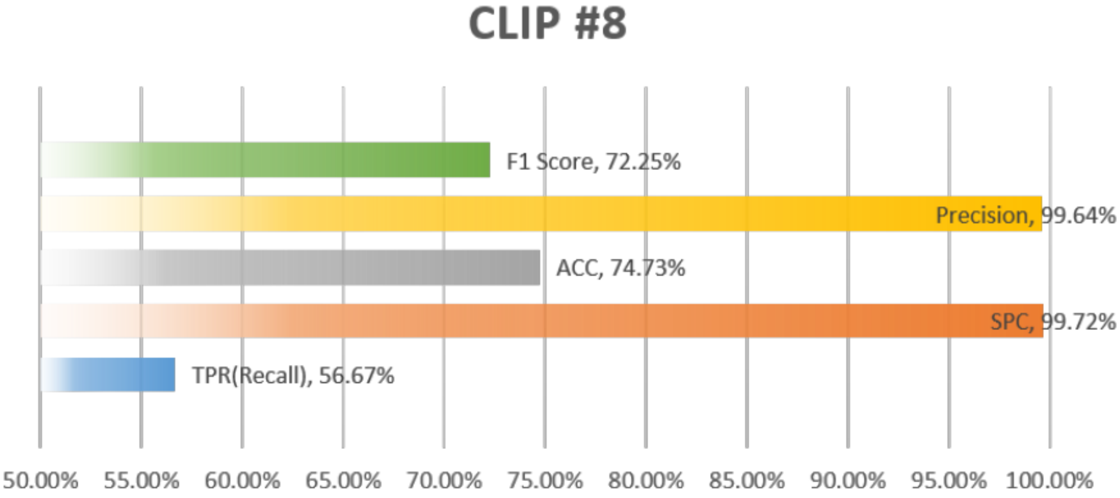
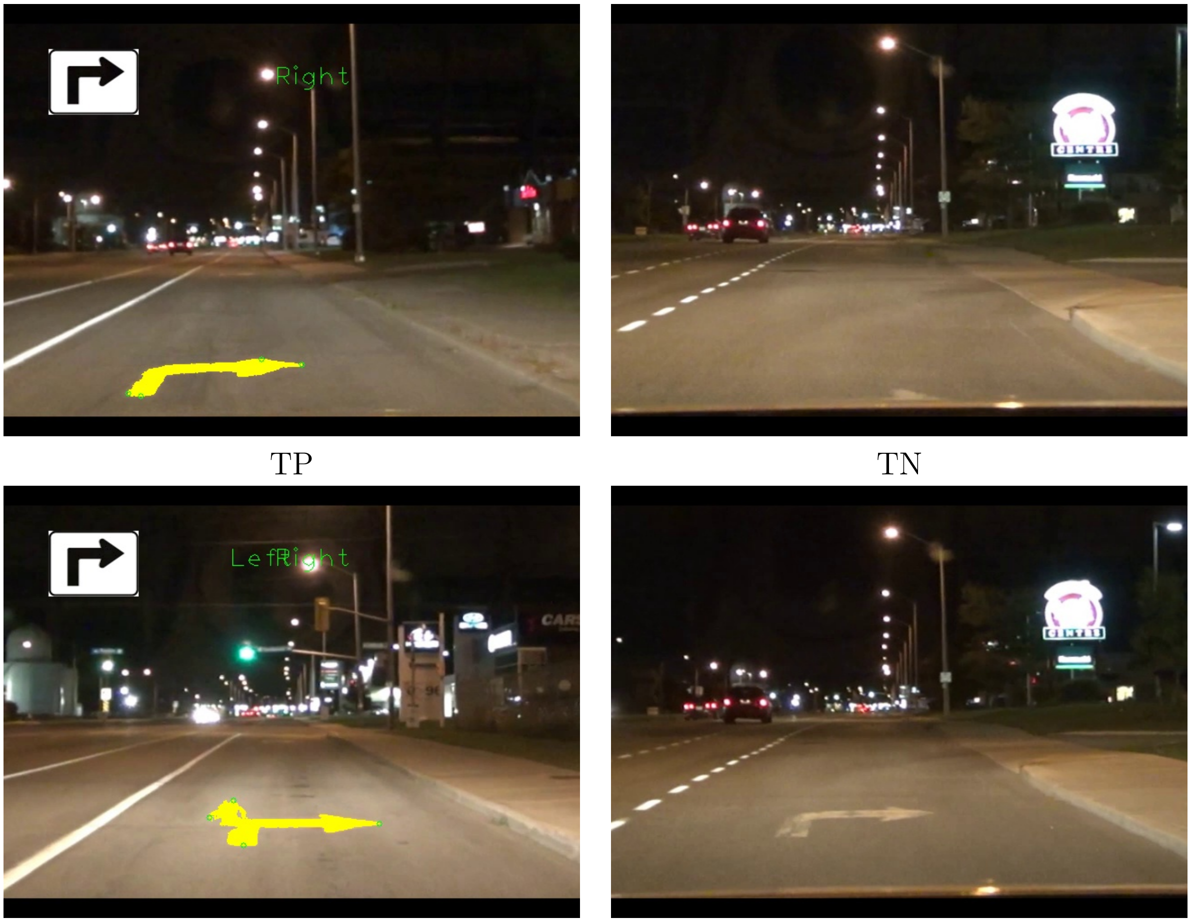


Figure 4.24: Statistical Results of Arrows, Clip #8

Due to the darkness in the scenario, the average performance of Clip #8 was not as high as in clips listed before. As shown in Fig. 4.24, precision and specificity both achieved a rate above 99.5%, which means all detection results were properly recognized. However, Accuracy of Clip #8 was 74.73%, which indicated that there were a certain number of



TP

TN

FP

FN

Figure 4.25: Samples of Clip #8

false positives or false negatives. Considering that both precision and specificity suggest the FP of this clip is 0, it is most likely FN is the main reason. TPR of Clip #8 was as low as 56.67%, which verified our hypothesis. It also indicated that many frames containing arrow objects were missed. After reviewing experimental data in Fig. 4.25, three reasons contribute most significantly to this. First and most important, a combination of low illumination and reflection from streetlamps together caused the low contrast in images (similar to the one shown as an example of FN in Fig. 4.25. Secondly, poorly maintained markings lost their contrast and turned blurry. Thirdly, some markings out of range in front of the ego vehicle were not taken into consideration.

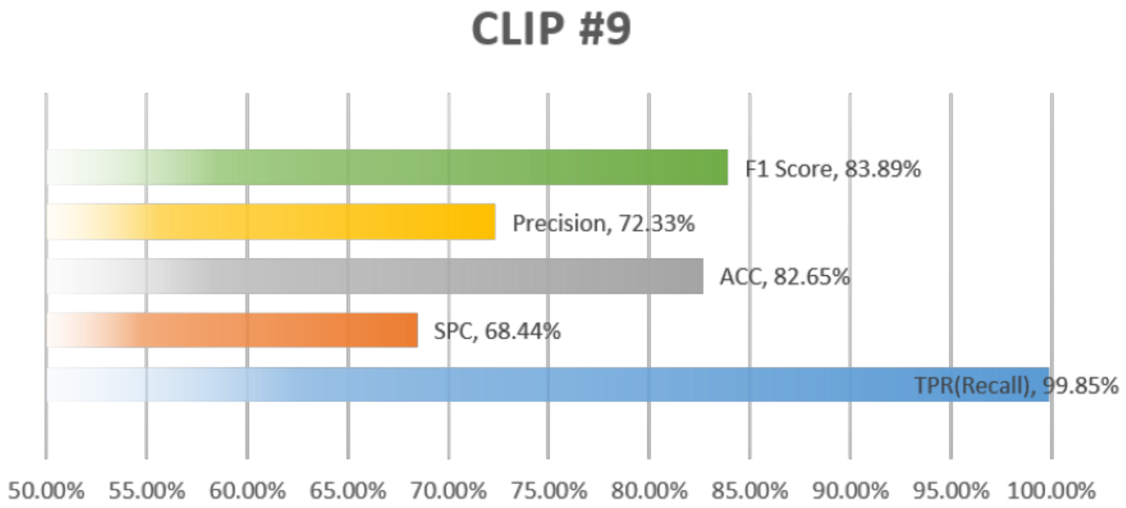
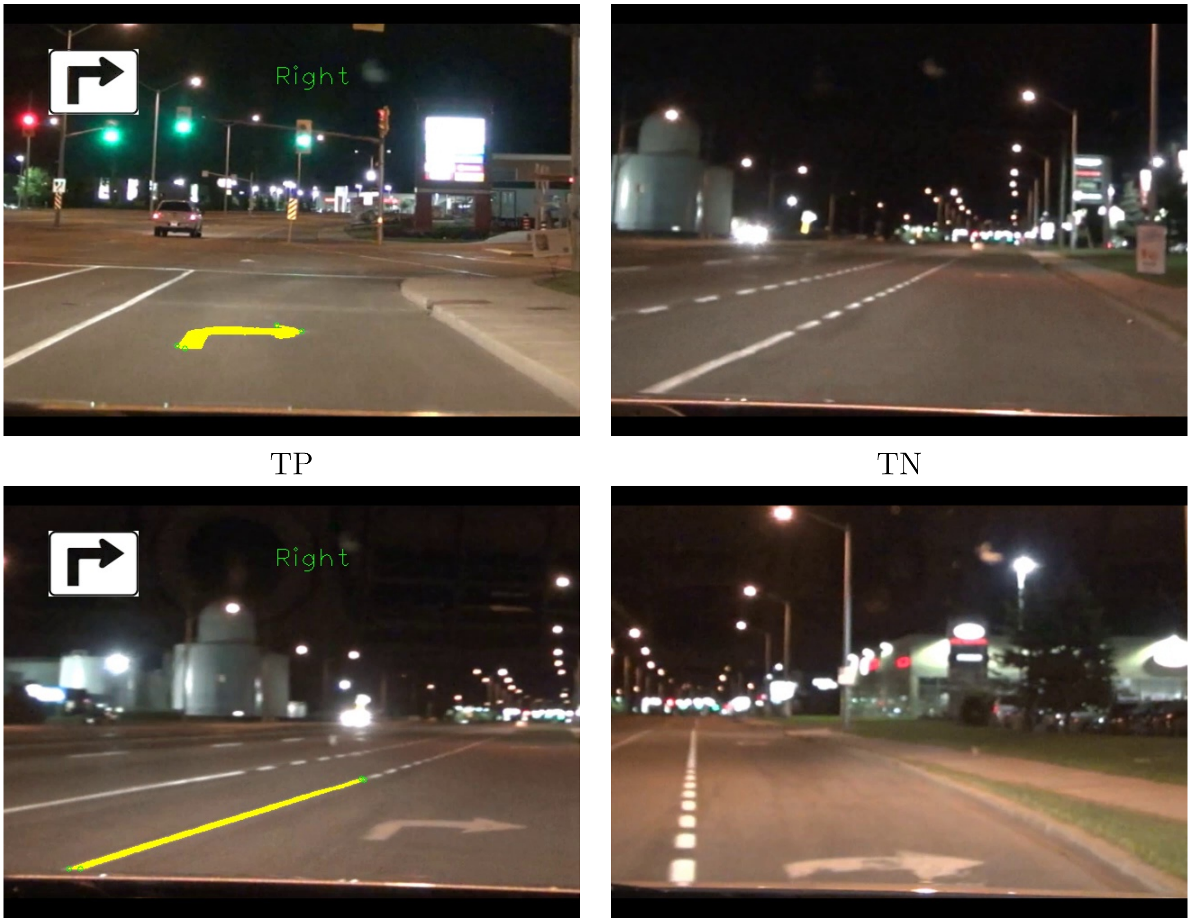


Figure 4.26: Statistical Results of Arrows, Clip #9

Based on similar illumination conditions, the same concept applies when analyzing the performance of Clip #9. As shown in Fig. 4.26, precision and specificity achieved a rate of 72.33% and 68.44% respectively, which means a number of detection results were not properly recognized. However, the accuracy of Clip #9 was 82.65%, indicating that there were a few false positives or false negatives. Specifically, TPR of Clip #9 was a high rate of 99.85%, which indicates that almost all frames containing arrow objects were properly recognized while many non-significant objects were wrongly recognized by the algorithm.



TP TN

FP FN

Figure 4.27: Samples of Clip #9

After a review of experimental data in Fig. 4.27, we identified 4 major contributors to this error. First and most important, width varies among lanes in this clip. This feature caused lots of mistaken recognition which treated sections of dashed lines as arrows (shown as a sample of FP in Fig. 4.27). Secondly, low illumination and reflection of streetlamp together resulted in low contrast of images. Thirdly, poorly maintained markings lost their contrast and turned blurry. Finally, some markings out of range in front of the ego vehicle were not taken into consideration.

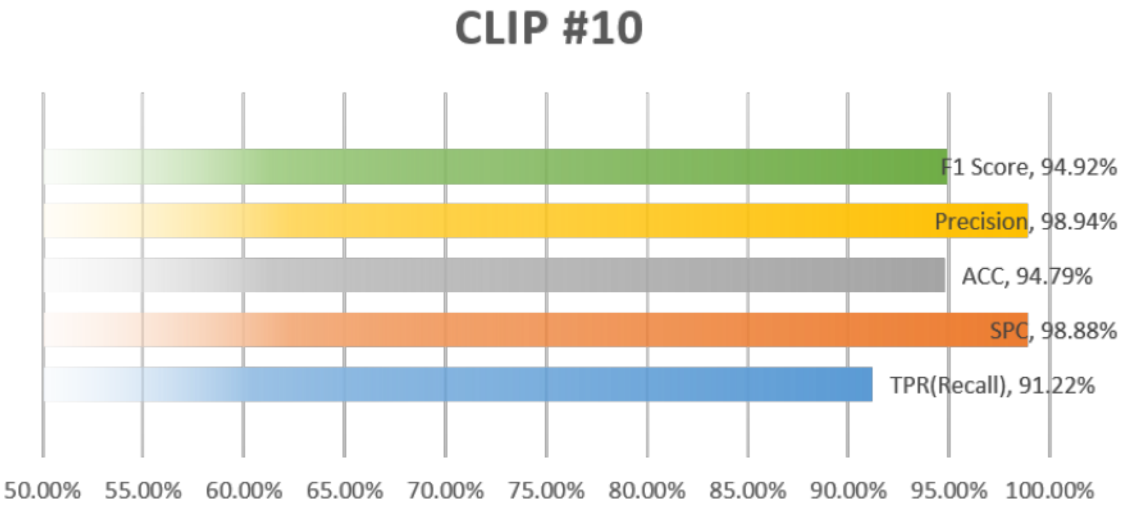


Figure 4.28: Statistical Results of Arrows, Clip #10

Finally, as shown in Fig. 4.28, precision and specificity both achieved a rate above 98%, which means almost all detection results were properly recognized. However, the accuracy of Clip #10 was 94.79%, indicating that there were a few false positives or false negatives. Considering that both Precision and Specificity suggest the FP of this clip is minimal, it is most likely FN is the main reason. The TPR of Clip #10 was 91.22%, thus verifying the conclusion. It also indicates that some frames containing arrow objects were missed. After a review of experimental data in Fig. 4.29, two reasons contribute the most significantly to this. First and most important, poorly maintained markings lost their contrast and turned blurry (similar to the one shown as an example of FN in Fig. 4.29. Secondly, some markings out of range in front of the ego vehicle were not taken into consideration.

TP TN

FP FN

Figure 4.29: Samples of Clip #10

Compared with Clip #9, the performance of Clip #10 is much better because the width of lane is constant which also support the fact that the proposed method adapts for most of conditions even in night time.

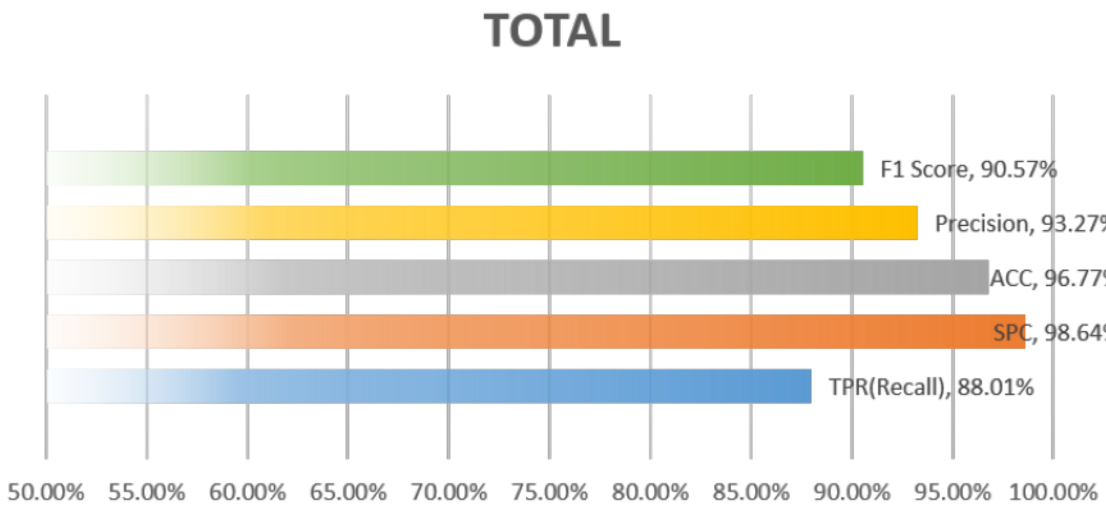


Figure 4.30: Summarized Result of All Clips

As shown in Fig. 4.30, overall accuracy and specificity both achieved a rate above 96%, which means almost all irrelevant detection results were properly processed. However, precision achieved a rate of 93.27%, indicating that there are a few false negatives. TPR of all clips as 88.01%, thus verifying the conclusion. It also indicates that some frames containing arrow objects were missed. After all the experimental data was reviewed, we found the total results are more strongly biased towards Clip #1 and Clip #2 since they contained the most frames among all. The two strongest factors which affected our indicators were poorly maintained road markings, and exclusion of markings close to the ego vehicle. Another indicator which needs to be mentioned is F score. As a composed parameter, it combines precision and TPR as a harmonic mean. Because of the fact that both of precision and TPR are remarkably good in our proposed experiment, F score (and especially F_1 score) achieved a similar rate of 88.01% accordingly, which indicates the proposed algorithm exhibited a state-of-the-art performance and represents a significant advancement of understanding.

At last, a comparison of processing time of all clips is listed below. According to Fig.

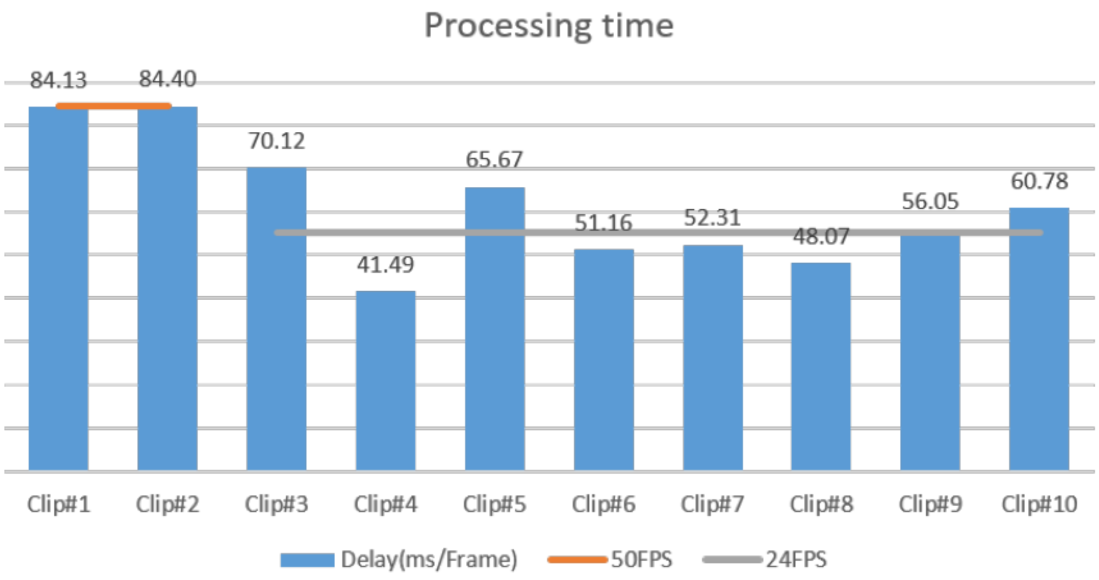


Figure 4.31: Summarized Processing Time of All Clips

4.31, Clip #1 and Clip #2 consume more time than any other clips. Due to a high volume of frames present in Clip #1 and Clip #2, their average processing time was more stable, at about 84.26 $ms/Frame$. Other clips, from Clip #3 to Clip #10, achieved an average processing time of 55.14 $ms/Frame$ however the individual performance was quite disparate. Specifically, the average processing time of Clip #4 was the smallest, at 41.49 $ms/Frame$. The main reason for that was the false negative rate was higher than average level and the duration of Clip #4 is short. In contrast, the average processing time of Clip #3 was the highest among all clips for arrow detection, reaching a rate of 70.12 $ms/Frame$. Considering that the secondary highest average processing time was obtained by Clip #5, it indicates that the processing time is more related to the conditions of the scenario. Because Clip #5 shared the same scenario but under night-time illumination as in Clip #3.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we have introduced a real-time lane marking detection framework for ADAS, which included 4 extreme points set descriptor and a rule-based cascade classifier. The main features applied in the proposed method obey Newtons Laws of Motion(e.g. the movement of vanishing point), therefore it is easy to examine and understand for all interested parties including automobile manufactories and regulators. Unlike traditional views which use statistical features with uncertainty to describe roads, we developed a series of clear and simple rules to describe road structure. Paved roads have a very long history since the oldest discovered road was constructed in Egypt circa 2200 BC. Some predominate rules of the road have been agreed upon by human society and massive clauses related to road construction and traffic control has been implemented according to the requirement of each city and country specifically. Like any other tools created by human beings with rational and scientific thinking, roads were designed and operated following a restricted discipline. Therefore, a branch of research work has developed in regards to finding the nature of road structure in this thesis.

In Chapter 1, the history of ADAS and the development of autonomous cars in academia and industry was reviewed first. Then a route map of related parties was summarized and exhibited. Since autonomous cars have become a favored research field recently, solutions for autonomous technology and a corresponding supervision system are urgently required. Some of the research conducted in Canada and worldwide was introduced afterward. By analyzing the requirements of the autonomous industry, some problems like the low reliability and lack of ability for environment sensing of existing autonomous system were indicated accordingly. These problems are also the driving force of our research into a lane marking detection framework for ADAS. Lastly, the main contribution of proposed method was provided and the thesis outline was then listed.

In Chapter 2, technology used for lane marking detection was summarized according to the sequence of standard image processing. It was believed that to explore a better method for road marking detection, reviewing the research in this field since the very beginning is necessary. After a comparison and analysis of the content among 81 academic articles, it became clear that most studies followed four steps: pre-processing, detection, recognition and tracking. After comparison of previous works, a trend that seeking an appropriate combination of descriptor and classifier were shown as the primary domain of this research. Because the descriptor and classifier used by these papers were mostly inherited from other fields of research and not optimized for road surfaces, it was possible to design an innovated descriptor and classifier based on the nature of road surfaces and thereby these two subjects constitute a large proportion of this chapter.

In Chapter 3, the proposed method in our experiment has been exhibited following a similar sequence as shown in Chapter 2 for comparison. After a discussion of characteristics of road lines and road markings according to the construction manual of civil engineering, an MSER-based high-efficiency integrated solution was denoted accordingly. The main idea of the proposed method includes a creative descriptor of the 4 extreme points set and a rule-based cascaded classifier, which was designed to classify detected objects, because this combination makes the best use of the characteristics of the road. Part of our previous work in line detection and recognition has been included in the frame of the whole system. A vanishing-point-tracking based environment sensing module was discussed to eliminate the error caused by instability in the ego vehicle's direction and provides the direction of the road network to calibrate the whole process in the end. A Kalman filter was utilized for tracking key points in this stage.

In Chapter 4, the experiment results were demonstrated in different aspects. This chapter is organized in 3 sections, which covers the experimental platform and overview, performance evaluation methodology, as well as detection and recognition results performance. Specifically, a platform including a laptop and digital video recorder mounted on the vehicle was constructed to test the proposed algorithm. 10 clips of real life videos (which included 32452 frames) were collected around the City of Ottawa. The program built with VC++ and QT was run on the laptop and gave feedback to the drivers in real time. After each road test, both video records and logs were stored. Then standard discriminant rules including True Positive Rate (TPR), Specificity (SPC), F-score, etc. were implemented on these experiment data to compare with other researchers' work. At the end, it was found that overall accuracy and specificity of proposed method both achieved a rate above 96%, indicating almost all irrelevant detection results were properly processed. Although the precision of 93.27% and TPR of 88.01% indicate that there are a few false negative frames, the proposed method exhibited a state-of-the-art performance and represent a significant advancement of understanding. Besides, an average processing time of 55.14 $ms/Frame$ in 24FPS clips indicates that proposed method is able to run in real time (with a delay of 13 $ms/Frame$).

5.2 Future Work

Some future work can be considered in order for the improvement and extension of our work.

Linear density could be applied in the marking detection, because the results of the proposed system are easily impacted by lane changes, thus it was of significant importance to judge if a series of pattern belonged to a dashed line or continuous markings like arrows. ROC analysis would be applied in the marking detection, because the regions are extracted by MSER which relies on a proper threshold. Although Moment-based methods have limitations when facing perspective distortion issues, it is worth trying to use clustering algorithms with time series analysis to improve the accuracy of arrow recognition. Additionally, cascade classifiers used in the proposed algorithm can be optimized through applying parallel computing structure likes CUDA or multi-processing.

As a part of intelligent transportation systems (ITS), lane markings detection and tracking can be combined with other detection. It can also be integrated with wireless sensor networks ([129], [130], [131], [132], [133], [134], [135], [136], [137], [60]). That is because advanced sensors like LIDAR, millimeter radar as well as hyperspectral data analysis may improve the performance of the specific task and the whole ADAS. Nevertheless, by building communication with vehicle networks like VANETs, missing information caused by high density of traffic flow can be restored or at least improved.

References

[1] SAE International. Sae international standard j3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. pages 1 – 12, 2014.

[2] S. Suchitra, R.K. Satzoda, and T. Srikanthan. Detection & classification of arrow markings on roads using signed edge signatures. pages 796–801. IEEE, 2012.

[3] TI. Advanced driver assistance systems (adas). `http://www.ti.com/lsds/ti/applications/automotive/adas/overview.page`, 2016. [Online Available].

[4] Look, no hands. `http://www.economist.com/node/21560989`, 2012. [Online Available].

[5] Yali Li, Shengjin Wang, Qi Tian, and Xiaoqing Ding. A survey of recent advances in visual feature detection. *Neurocomputing*, 149, Part B:736 – 751, 2015.

[6] Weina Lu, Haifang Wang, and Qingzhu Wang. A synchronous detection of the road boundary and lane marking for intelligent vehicles. volume 1, pages 741–745. IEEE, 2007.

[7] Bao Rong Chang, Hsiu Fen Tsai, and Chung-Ping Young. A fast and effective approach to lane marking and neighboring vehicles detections based on vision/gps sensing together with vehicle-to-vehicle communication. volume 1, pages 3–8. IEEE, 2009.

[8] J. C. H. Leung and G. F. NcLean. Vanishing point matching. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 305–308 vol.2, Sep 1996.

[9] Young-Woo Seo and R.R. Rajkumar. Utilizing instantaneous driving direction for enhancing lane-marking detection. pages 170–175. IEEE, 2014.

[10] Tao Wu and A. Ranganathan. Vehicle localization using road markings. pages 1185–1190. IEEE, 2013.

[11] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239, March 1984.

[12] Yunchong Li, Kezhong He, and Peifa Jia. Road markers recognition based on shape information. pages 117–122. IEEE, 2007.

[13] Masafumi Noda, Tomokazu Takahashi, Daisuke Deguchi, Ichiro Ide, Hiroshi Murase, Yoshiko Kojima, and Takashi Naito. T.: Recognition of road markings from in-vehicle camera images by a generative learning method. In *In: Proc. IAPR Conf. on Machine Vision Applications*, 2009.

[14] M. Schreiber, F. Poggenhans, and C. Stiller. Detecting symbols on road surface for mapping and localization using ocr. pages 597–602. IEEE, 2014.

[15] L. Wendling and S. Tabbone. A new way to detect arrows in line drawings. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:935–941, 2004.

[16] Ontario Goverment. *Ontario Traffic Manual-Pavement, Hazard and Delineation Markings.* 1 edition, 3 2000.

[17] Transportation Association of Canada. *Uniform traffic control devices for Canada.* 1 edition, 1960.

[18] Transportation Association of Canada. *Uniform traffic control devices for Canada.* 5 edition, 2014.

[19] Ontario Goverment. *O. Reg. 306/15: PILOT PROJECT - AUTOMATED VEHICLES.* 2 edition, 1 2016.

[20] David Ticoll. *Driving Changes: Automated Vehicles in Toronto.* 2016.

[21] ALEX DAVIES. Gm and lyft are building a network of self-driving cars. `https://www.wired.com/2016/01/gm-and-lyft-are-building-a-network-of-self-driving-cars/`, 2016. [Online Available].

[22] 33 corporations working on autonomous vehicles. `https://www.cbinsights.com/blog/autonomous-driverless-vehicles-corporations-list/`, 2016. [Online Available].

[23] Ford targets fully autonomous vehicle for ride sharing in 2021; invests in new tech companies, doubles silicon valley team. `https://media.ford.com/content/fordmedia/fna/us/en/news/2016/08/16/ford-targets-fully-autonomous-vehicle-for-ride-sharing-in-2021.html`, 2016. [Online Available].

[24] Autonomous driving. `https://corporate.ford.com/microsites/sustainability-report-2014-15/mobility-autonomous.html`, 2016. [Online Available].

[25] Zachary Shahan. Bosch, daimler, & car2go launching autonomous car parking pilot. `https://cleantechnica.com/2015/07/11/bosch-daimler-car2go-launching-autonomous-car-parking-pilot/`, 2016. [Online Available].

[26] Tech center a-drive: Eur 7.5 million for automated driving. `http://www.kit.edu/kit/english/pi_2016_005_tech-center-a-drive-eur-7-5-million-for-automated-driving.php`, 2016. [Online Available].

[27] Darrell Etherington. Audi setting up a subsidiary dedicated to self-driving car tech. `https://techcrunch.com/2016/07/25/audi-setting-up-a-subsidiary-dedicated-to-self-driving-car-tech/`, 2016. [Online Available].

[28] Technology of volkswagen group. `http://www.volkswagengroupamerica.com/technology.html`, 2016. [Online Available].

[29] Dr. Alexander Hars. First fully autonomous audi expected by 2017. `http://www.driverless-future.com/?cat=5`, 2016. [Online Available].

[30] Mobileye. Artificial vision technology. `http://www.mobileye.com/technology/`, 2016. [Online Available].

[31] Jason Kolsevich. Autonomous vehicle partnerships who are the early players? `https://www.waterstreetpartners.net/blog/autonomous-vehicle-partnerships-an-update`, 2016. [Online Available].

[32] Drive me of volvo. `http://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/intellisafe-autopilot/drive-me/progress-through-partnership`, 2016. [Online Available].

[33] Jordan Golson. Volvo and uber ink deal to develop base vehicles for autonomous cars. `http://www.theverge.com/2016/8/18/12541672/uber-volvo-partnership-autonomous-self-driving-car`, 2016. [Online Available].

[34] Self-driving cars on the motorway. `http://www.psa-peugeot-citroen.com/en/featured-content/connected-car/self-driving-cars`, 2016. [Online Available].

[35] Why psa chose trw as its driver assist systems partner. `http://safety.trw.com/why-psa-chose-trw-as-its-driver-assist-systems-partner/0106/`, 2016. [Online Available].

[36] Mines paristech announces the creation of an international research chair on automated driving. `http://www.psa-peugeot-citroen.com/en/content/mines-paristech-announces-creation-international-research-chair-\automated-driving-0`, 2016. [Online Available].

[37] Jason Kolsevich. Autonomous vehicle partnerships: How tech companies and automakers are collaborating to innovate the future. `https://www.waterstreetpartners.net/blog/autonomous-vehicle-\partnerships-how-tech-companies-and-automakers-are-collaborating-to-\innovate-the-future`, 2016. [Online Available].

[38] Reuters. Bmw has 2 new tech partners for driverless cars. `http://fortune.com/2016/06/30/bmw-mobileye-intel-team-on-driverless-cars/`, 2016. [Online Available].

[39] Renault next two : Our take on the autonomous and connected car. `https://group.renault.com/en/passion-2/innovation/renault-a-born-innovator/autonomous-and-connected-car/`, 2016. [Online Available].

[40] Mert Bal. Can toyotas self-driving lexus save you from accidents? `http://www.engineering.com/ElectronicsDesign/ElectronicsDesignArticles/ArticleID/5160/Can-Toyotas-Self-Driving-Lexus-save-you-from-Accidents.aspx`, 2013. [Online Available].

[41] Roberto Baldwin. Inside honda's ghost town for testing autonomous cars. `https://www.engadget.com/2016/06/02/inside-honda-gomentum-station/`, 2016. [Online Available].

[42] David Undercoffler. Honda merges into autonomous traffic. `http://www.autonews.com/article/20160605/OEM06/306069964/honda-merges-into-autonomous-traffic`, 2016. [Online Available].

[43] Mitsubishi unit adapts military tech for self-driving cars. `http://www.autonews.com/article/20160330/COPY01/303309940/mitsubishi-unit-adapts-military-tech-for-self-driving-cars`, 2016. [Online Available].

[44] DAVID CURRY. Mitsubishi missile attack on self-driving cars. `http://readwrite.com/2016/04/01/mitsubishi-missles-self-driving-cars-tt4/`, 2016. [Online Available].

[45] Hans Greimel. First look: Subaru shows off its next-generation chassis. `http://autoweek.com/article/car-news/subaru-unveils-next-generation-global-platform`, 2016. [Online Available].

[46] Park Jin-hai. Hyundai to test autonomous cars in u.s. `http://www.koreatimes.co.kr/www/news/biz/2016/08/388_193175.html`, 2015. [Online Available].

[47] Kirsten Korosec. Hyundai taps cisco to develop connected car tech. `http://fortune.com/2016/09/12/renesas-buys-intersil/`, 2016. [Online Available].

[48] Benjamin Hunting. First ride: Kias autonomous soul ev. `http://www.autotrader.ca/newsfeatures/20160125/first-ride-kias-autonomous-soul-ev/#3AvhSea7Rf0t43Yc.97`, 2016. [Online Available].

[49] Lynn Walford. Why daimler, tesla, audi & delphi are winning self-driving connected car race in q2. `http://www.autoconnectedcar.com/2015/05/why-daimler-tesla-audi-delphi-are-winning-self-driving-race-in-q2/`, 2015. [Online Available].

[50] Google self-driving car project. `https://www.google.com/selfdrivingcar/how/`, 2016. [Online Available].

[51] Lisa Eadicicco. Ubers first self-driving car is hitting the streets. `http://time.com/4341517/uber-first-self-driving-car/`, 2016. [Online Available].

[52] Dragos Serban. China responds to google's driverless car with the hongqi hq3. `http://www.autoevolution.com/news/china-responds-to-googles-driverless-car-with-the-hongqi-hq3-37850.html`, 2016. [Online Available].

[53] Velodyne Lidar. China: Baidu autobrain self-driving car is tested in beijing. `https://www.sbdautomotive.com/en/china-baidu-autobrain`, 2016. [Online Available].

[54] Goverment of Canada Transport Canada. *Road Safety in Canada*. 1 edition, 3 2011.

[55] K. Guan, B. Ai, M. Liso Nicols, R. Geise, A. Mller, Z. Zhong, and T. Krner. On the influence of scattering from traffic signs in vehicle-to-x communications. *IEEE Transactions on Vehicular Technology*, 65(8):5835–5849, Aug 2016.

[56] Bradley Berman. Lower-cost lidar is key to self-driving future. `http://articles.sae.org/13899/`, 2015. [Online Available].

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

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

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] Steven Shladover, Dongyan Su, and Xiao-Yun Lu. Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transportation Research Record: Journal of the Transportation Research Board*, 2324:63–70, 2012.

[70] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[71] D. Frank. Road markings recognition. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 669–672 vol.2, Sep 1996.

[72] Bao Rong Chang and Hsiu Fen Tsai. Applying visual image and satellite positioning for fast vehicle detection and lane marking recognition. pages 518–521. IEEE, 2009.

[73] NOBUYUKI OTSU. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, Jan 1979.

[74] Haiyan Guan, J. Li, Yongtao Yu, and Cheng Wang. Rapid update of road surface databases using mobile lidar: Road-markings. pages 124–129. IEEE, 2013.

[75] Haitao Ding, Bowei Zou, Konghui Guo, and Cong Chen. Comparison of several lane marking line recognition methods. pages 53–58. IEEE, 2013.

[76] M.J Magee and J.K Aggarwal. Determining vanishing points from perspective images. *Computer Vision, Graphics, and Image Processing*, 26(2):256 – 267, 1984.

[77] Alireza Kheyrollahi and Toby P. Breckon. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications*, 23(1):123–133, 2012.

[78] C. Baghdassarian, H. Lange, H. Sahli, and C. Laurgeau. Recognition of arrows in the environment of road markings. pages 219–224. IEEE, 1994.

[79] Robert Laganire. Compositing a bird's eye view mosaic. In *Conf. Vision Interface, Montreal, Canada*, volume 10, pages 382–387, 2000.

[80] Hua Li, Mingyue Feng, and Xiao Wang. Inverse perspective mapping based urban road markings detection. volume 03, pages 1178–1182. IEEE, 2012.

[81] Zhao Li, Zi xing Cai, Jin Xie, and Xiao ping Ren. Road markings extraction based on threshold segmentation. pages 1924–1928. IEEE, 2012.

[82] Ziqiong Liu, Shengjin Wang, and Xiaoqing Ding. Roi perspective transform based road marking detection and recognition. pages 841–846. IEEE, 2012.

[83] B. Qin, W. Liu, X. Shen, Z.J. Chong, T. Bandyopadhyay, M.H. Ang, E. Frazzoli, and D. Rus. A general framework for road marking detection and analysis. pages 619–625. IEEE, 2013.

[84] Nan Wang, Wei Liu, Chunmin Zhang, Huai Yuan, and Jiren Liu. The detection and recognition of arrow markings recognition based on monocular vision. pages 4380–4386. IEEE, 2009.

[85] I.M. Chira, A. Chibulcutean, and R.G. Danescu. Real-time detection of road markings for driving assistance applications. pages 158–163. IEEE, 2010.

[86] P. Foucher, Y. Sebsadji, J.-P. Tarel, P. Charbonnier, and P. Nicolle. Detection and recognition of urban road markings using images. pages 1747–1752. IEEE, 2011.

[87] Uhang He, hi Chen, Ifeng Pan, and Kai Ni. Using edit distance and junction feature to detect and recognize arrow road marking. pages 2317–2323. IEEE, 2014.

[88] W. Liu, J. Lv, B. Yu, W. Shang, and H. Yuan. Multi-type road marking recognition using adaboost detection and extreme learning machine classification. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 41–46, June 2015.

[89] F. Poggenhans, M. Schreiber, and C. Stiller. A universal approach to detect and classify road surface markings. pages 1915–1921. IEEE, 2015.

[90] Cheng Li, Ivo Creusen, Lykele Hazelhoff, and Peter H. N. de With. Detection and recognition of road markings in panoramic images. volume 9407, pages 940708–940708–10, 2015.

[91] T. Veit, J. P. Tarel, P. Nicolle, and P. Charbonnier. Evaluation of road marking feature extraction. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 174–181, Oct 2008.

[92] M. Mizuno, K. Yamada, T. Nakano, and S. Yamamoto. Robustness of lane mark detection with wide dynamic range vision sensor. pages 171–176. IEEE, 1996.

[93] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.

[94] Chin-Yu Chang and Chang-Hong Lin. An efficient method for lane-mark extraction in complex conditions. pages 330–336. IEEE, 2012.

[95] M. van Ginkel, C.L. Luengo Hendriks, and L.J. van Vliet. A short introduction to the radon and hough transforms and how they relate to each other, 2004.

[96] M. P. N. Burrow, H. T. Evdorides, and M. S. Snaith. Segmentation algorithms for road marking digital image analysis. *Proceedings of the Institution of Civil Engineers - Transport*, 156(1):17–28, 2003.

[97] Y. Ishino and H. Saji. Extraction of road markings from aerial images. pages 2180–2183. IEEE, 2008.

[98] E. Pollard, D. Gruyer, J. P. Tarel, S. S. Ieng, and A. Cord. Lane marking extraction with combination strategy and comparative evaluation on synthetic and camera images. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1741–1746, Oct 2011.

[99] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[100] Stefan Vacek, Constantin Schimmel, and Rdiger Dillmann. Road-marking analysis for autonomous vehicle guidance. In *EMCR*, 2007.

[101] Tao Wu and A. Ranganathan. A practical system for road marking detection and recognition. pages 25–30. IEEE, 2012.

[102] Abdelhamid Mammeri, Azzedine Boukerche, and Zongzhi Tang. A real-time lane marking localization, tracking and communication system. *Computer Communications*, 73, Part A:132 – 143, 2016.

[103] Y. Sebsadji, J.-P. Tarel, P. Foucher, and P. Charbonnier. Robust road marking extraction in urban environments using stereo images. pages 394–400. IEEE, 2010.

[104] Dixiao Cui, Jianru Xue, Shaoyi Du, and Nanning Zheng. Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration. pages 4958–4964. IEEE, 2014.

[105] D.C. Hernandez, A. Filonenko, Dongwook Seo, and Kang-Hyun Jo. Lane marking recognition based on laser scanning. pages 962–965. IEEE, 2015.

[106] L. Hazelhoff, I. Creusen, T. Woudsma, Xinfeng Bao, and P.H.N. de With. Combined generation of road marking and road sign databases applied to consistency checking of pedestrian crossings. pages 439–442. IEEE, 2015.

[107] Range. A review of irregular shapes matching technique. `http://www.cnblogs.com/CVArt/archive/2011/07/03/2096670.html`, 2011. [Online Available].

[108] David G. Kendall. A survey of the statistical theory of shape. *Statist. Sci.*, 4(2):87–99, 05 1989.

[109] G. Maier, S. Pangerl, and A. Schindler. Real-time detection and classification of arrow markings using curve-based prototype fitting. pages 442–447. IEEE, 2011.

[110] P. Charbonnier, F. Diebolt, Y. Guillard, and F. Peyret. Road markings recognition using image processing. pages 912–917. IEEE, 1997.

[111] Sio-Song Ieng, Jean-Philippe Tarel, and R. Labayrade. On the design of a single lane-markings detectors regardless the on-board camera's position. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 564–569, June 2003.

[112] J. Rebut, A. Bensrhair, and G. Toulminet. Image segmentation and pattern recognition for road marking analysis. volume 1, pages 727–732 vol. 1. IEEE, 2004.

[113] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, April 2010.

[114] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

[115] Haiyan Guan, Jonathan Li, Yongtao Yu, Cheng Wang, Michael Chapman, and Bisheng Yang. Using mobile laser scanning data for automated extraction of road markings. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, 87:93 – 107, 2014.

[116] J. K. Suhr and H. G. Jung. Fast symbolic road marking and stop-line detection for vehicle localization. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 186–191, June 2015.

[117] Tairui Chen, Zhilu Chen, Quan Shi, and Xinming Huang. Road marking detection and classification using machine learning algorithms. pages 617–621. IEEE, 2015.

[118] T. Hara and Y. Aoki. A robust road marking detection method with pseudo-top-view images for creating database. pages 1–5. IEEE, 2011.

[119] M. Braga de Paula and C. Rosito Jung. Real-time detection and classification of road lane markings. pages 83–90. IEEE, 2013.

[120] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(1):3133–3181, January 2014.

[121] W. Enkelmann, G. Struck, and J. Geisler. Roma - a system for model-based analysis of road markings. pages 356–360. IEEE, 1995.

[122] C. Kreucher and S. Lakshmanan. Lana: a lane extraction algorithm that uses frequency domain features. *Robotics and Automation, IEEE Transactions on*, 15:343–350, 1999.

[123] Guangqian Lu. A lane detection, tracking and recognition system for smart vehicles. Master's thesis, University of Ottawa, Ottawa, Ontario, 2015.

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

[125] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. British Machine Vision Computing 2002.

[126] H. Sun, C. 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.

[127] 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.

[128] Transport Canada. *High Occupancy Vehicle Lanes in Canada.* 1 edition, 12 2007.

[129] 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.

[130] Azzedine 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.

[131] 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.

[132] 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.

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

[134] 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.

[135] 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.

[136] 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.

[137] 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.