A NEW DRIVER WARNING SYSTEM DEVELOPMENT AND ITS IMPLEMENTATION ON A REAL-TIME BASIS

(YENİ BİR SÜRÜCÜ UYARI SİSTEMİNİN GELİŞTİRİLMESİ VE GERÇEK ZAMANLI OLARAK UYGULANMASI)

by

Burak ÇAYIR, B.S.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

INSTITUTE OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

May 2009

A NEW DRIVER WARNING SYSTEM DEVELOPMENT AND ITS IMPLEMENTATION ON A REAL-TIME BASIS

(YENİ BİR SÜRÜCÜ UYARI SİSTEMİNİN GELİŞTİRİLMESİ VE GERÇEK ZAMANLI OLARAK UYGULANMASI)

by

Burak ÇAYIR, B.S.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Date of Submission	: May 14, 2009
Date of Defense Examination	: May 29, 2009

Supervisor	: Asst. Prof. Dr. Tankut ACARMAN
Committee Members	: Assoc. Prof. Dr. Temel ÖNCAN
	Asst. Prof. Dr. Murat AKIN

ACKNOWLEDGEMENTS

There are many people who deserve acknowledgment for the help I have received while working on this thesis and during my 17 years of study. Unfortunately it's impossible to mention them all.

I would like to express my inmost gratitude to my supervisor Dr. Tankut Acarman for his guidance, patience, advice, insight throughout the research and trust on me.

I am indebted to my parents Hüseyin and Gülay and elder sister Zeynep for all their support and self-sacrifice on my behalf.

I'd like to thank to my girlfriend Ayşen Uludağ for her tolerance and support.

I am also thankful to my friends Erkan Yüksel and Umur Güldiren for their assistance and contribution whenever I need their expertise.

I would also like to express a special thanks to TÜBİTAK(Turkish Scientific and Technical Research Institute) for their scholarship.

I gratefully acknowledge support of the Turkish National Planning Organization through the Drive Safe project and the NEDO grant “International Research Coordination of Driving Behavior Signal Processing Based on Large Scale Real World Database” from Japan.

May 2009
Burak ÇAYIR

TABLE OF CONTENTS

Acknowledgements....................ii

Table of Contents....................iii

List of Symbols....................v

List of Figures....................vi

List of Tables....................ix

Abstract....................x

Résumé....................xi

Özet....................xii

1. Introduction....................1
   1.1. Motivation....................1
   1.2. Review of Literature....................2
   1.3. Driver Assistance Systems....................3
   1.4. Perception and Reaction....................4
   1.5. Human Driver Model....................6
   1.6. Main Deliverable....................7

2. Review of Image Processing Techniques....................9
   2.1. Sobel Edge Detection....................9
   2.2. Dilation....................11
   2.3. Conversion of Image Types and Thresholding....................12
   2.4. Gaussian Smoothing Filter....................13
   2.5. Hough Transform....................14
   2.6. Template Matching....................16
   2.7. Blob Detection....................17
   2.8. Optical Flow....................18

3. Driver Warning System Development....................21
   3.1. Introduction....................21

3.2. Lane Departure Warning..21
3.3. Forward Collision Warning..37
3.4. Blind Spot Warning..42
4. Implementation of the System on a Real-Time Basis...48
4.1. Configuration Aspects and Mechanical Parts of the System.............................48
4.2. Blind Spot Warning Module Aspects...49
4.3. Lane Departure Warning Module Aspects...50
4.4. CAN Bus Support..52
4.5. Warning Devices..53
4.6. Experiments and Results...55
5. Conclusion...61
References..62
Appendix A..65
Appendix B..66
Appendix C..67
Biographical Sketch...68

LIST OF SYMBOLS

DAS	: Driver Assistance Systems
LDW	: Lane Departure Warning
FCW	: Forward Collision Warning
BSW	: Blind Spot Warning
TTC	: Time to Collision
ROI	: Region of Interest

LIST OF FIGURES

Figure 1.1 : Principle of Operation of Adaptive Cruise Control......................................4

Figure 1.2 : Stopping the Slide...5

Figure 2.1 : Sobel Convolution Kernels...9

Figure 2.2 : Result of Sobel Edge Detection...11

Figure 2.3 : Dilation of Edges..12

Figure 2.4 : Same image as RGB, Grayscale and Binary Respectively..........................13

Figure 2.5 : Effect of Gaussian Smoothing Filter...14

Figure 2.6 : Hough Transformation..15

Figure 2.7 : Result of Hough Transform on a Road Segment Image..............................16

Figure 2.8 : Template Matching Example...17

Figure 2.9 : External and Internal Contours on an Image...19

Figure 3.1 : RGB Image Captured from the Forward Camera..22

Figure 3.2 : Gray Level Conversion Result of an RGB Image.......................................22

Figure 3.3 : Sobel Edge Detection Result...22

Figure 3.4 : Thresholded Image with parameter 100...23

Figure 3.5 : Filtering Process...23

Figure 3.6 : Result of Filtering Process..24

Figure 3.7 : Image After Morphological Open..24

Figure 3.8 : Blobs Found in the Image...25

Figure 3.9 : Borders of Segments in an Image...26

Figure 3.10 : Results of Hough Transform in Image Segments......................................27

Figure 3.11 : Logically OR Operation for Blobbed and Hough Transformed Images....27

Figure 3.12 : Image Obtained with Possible Lane Candidates.......................................28

Figure 3.13 : Clustering the Points According to Distance...29

Figure 3.14 : Polynomial Evaluation Areas on the Image..31

Figure 3.15 : Fitted Polynomials for Lane Borders...32

Figure 3.16 : Angles for Arc Drawing..33
Figure 3.17 : Lane Borders on a Curved Road...33
Figure 3.18 : Shifting the Curve..34
Figure 3.19 : Calculation of the Lane Departure..35
Figure 3.20 : Result of Lane Detection..35
Figure 3.21 : Flowchart of Lane Departure Warning Subsystem..................................36
Figure 3.22 : Hough Transform vs. Proposed Method..36
Figure 3.23 : Image Captured for Forward Vehicle Detection.......................................39
Figure 3.24 : Binary Image for Forward Vehicle Detection...39
Figure 3.25 : Shadow Image Template...39
Figure 3.26 : Region of Interest to Search the Template..40
Figure 3.27 : Vehicle Found on the Image..40
Figure 3.28 : Flowchart of the Forward Collision Warning Subsystem.........................41
Figure 3.29 : Blind Spot Diagram of a Vehicle..42
Figure 3.30 : Regions of Interests on an Image for Blind Spot Warning.......................43
Figure 3.31 : Optical Flow Result..44
Figure 3.32 : Blind Spot Warning on an Empty Road..44
Figure 3.33 : Blind Spot Warning with a Far Vehicle...45
Figure 3.34 : Blind Spot Warning with an Approaching Vehicle..................................45
Figure 3.35 : Blind Spot Warning with a Near Vehicle..46
Figure 3.36 : Blind Spot Warning with a Very Near Vehicle...46
Figure 3.37 : Blind Spot Warning Module Flowchart...47
Figure 4.1 : Camera Installation on Vehicle..48
Figure 4.2 : Finite State Machine Modeling of Blind Spot Detection Algorithm...........50
Figure 4.3 : Lane Departure Position...52
Figure 4.4 : Bit Locations on Parallel Port..54
Figure 4.5 : Circuit of a Board of Equipment...54
Figure 4.6 : Physical Equipments..55
Figure 4.7 : Data Acquisition Test Vehicle..56
Figure 4.8 : Camera Installation on Test Vehicle...56
Figure 4.9 : Data Storage and Processing in the Left Back Seat....................................57
Figure 4.10 : Forward Collision Warning Result...58

Figure 4.11 : Blind Spot Warning Result..58
Figure 4.12 : Screenshot of LDW with GPS and CAN Information..............................59
Figure 4.13 : Lane Detection Results...59
Figure 4.14 : General Overview of the System..60

LIST OF TABLES

Table 4.1 : Performance of the System..60

ABSTRACT

Most studies so far on Advanced Driver Warning Systems have assumed perfect knowledge of driving conditions, driver intents which require expensive sensor suite, data communication and processing. In this thesis, a fairly restrictive driver warning system supporting “blind spot”, “lane departure” detection and “safe following distance” monitoring is studied. A vision-based active safety system is presented to prevent possible traffic accidents during lane change maneuvering and lane following tasks. The active system considers that the driver is always in-the-loop and in case of lane departure detection or lane change unintentionally into the path of the coming vehicle, visual and audio signal is generated to warn the driver.

Proposed system brings a new approach to blind spot detection and lane recognition. Blob Detection and Hough Transform are used together for a robust detection of lanes. White pixel intensity method is proposed to be aware of the dangers in blind spot. Forward vehicles are easily detected by template matching algorithm.

This new system does not only bring new approaches but also a lower cost basis comparing with the other after market products thanks to its three cameras and an onboard computer.

RESUME

La plupart des études faites jusqu'alors sur les systèmes avancés d'alerte du conducteur ont pris comme postulat la connaissance parfaite des conditions de conduite, les intentions du conducteur qui exigent la présence d'une série de capteurs onéreux, de transmission de données et traitement de ces données. Dans cette thèse, on présente un système d'alerte du conducteur assez restrictif qui inclut les changements de voie durant la conduite, les angles morts, et la distance de sécurité de suivi. Un système de sûreté fondé sur la vision active est présenté pour empêcher les accidents de la circulation possibles pendant la manœuvre de changement de voie durant la conduite.

Le système actif considère que le conducteur est toujours responsable et en cas de la détection de changement de voie durant la conduite volontaire ou non alors qu'un véhicule approche, un signal visuel et audio est produit pour avertir le conducteur. Le système proposé apporte une nouvelle approche à la détection des angles morts et à l'identification de la chaussée. La détection de Blobs et la transformée de Hough sont couplées pour une solide détection des chaussées. On propose la méthode d'intensité des pixels blancs pour se rendre compte des dangers dans les angles morts. Des véhicules suivants sont facilement détectés par l'algorithme de template matching.

Ce nouveau système apporte non seulement de nouvelles approches mais également une base bien moins coûteuse que les autres produits de pièces détachées avec ses trois caméras et un ordinateur de bord.

ÖZET

Bugüne kadar, ileri sürücü güvenlik sistemleri üzerinde yapılan araştırmalar, sürüş koşullarının ve ancak çok pahalı algılayıcılarla yüksek veri alışverişi ve işlemesi sonucu anlaşılabilen sürücü eğilimlerinin çok iyi bilindiğini varsaymışlardır. Bu tezde, şerit ihlallerini tespit eden, kör nokta tespiti yapabilen ve güvenli sürüş mesafesini takip edebilen bir sürücü uyarı sistemi üzerinde çalışılmıştır. Şerit takibi ve şerit değiştirme manevraları esnasında olası trafik kazalarını engellemek amacıyla görme tabanlı aktif uyarı sistemi sunulmuştur. Etkin sistem, sürücünün her zaman aracı kontrol ettiği varsayılarak, şerit değişikliği yapması veya istemdışı olarak karşıdan gelen aracın şeridine girmesi durumunda, sürücüyü uyarmak için sesli ve görsel olarak uyarılar üretmektedir.

Önerilen sistem, şerit tanıma ve kör nokta tespitine yeni yaklaşımlar getirmektedir. Blob tespiti ve Hough dönüşümü, daha güvenilir şerit tanıma amacıyla bir arada kullanılmaktadır. Beyaz piksel yoğunluğu yöntemi ile kör noktalardaki tehlikelerden haberdar olunabilmekte ve şablon karşılaştırma algoritmasıyla da öndeki aracın durumu izlenebilmektedir.

Bu yeni sistem sadece yeni yaklaşımlar getirmemiş aynı zamanda üç adet kamera ve bir adet gömülü bilgisayarıyla piyasadaki diğer satış sonrası ürünlerden daha düşük bir maliyete sahip olmuştur.

1. INTRODUCTION

1.1. Motivation

One of the major causes of the accidents on the road is determined by the perception limits of the driver. In order to increase perception limits and at the same time to help decrease reaction limits, different type sensor suites are proposed to percept the surrounding vehicles, stationary or moving obstacles in the moving traffic. The data coming from sensors and the driver's responses and analysis & evaluation of the current situation based on the relevant constraints, the future position of the vehicle with respect to the left and right lane and the time to collision to the vehicle in front or vehicle the vehicle approaching from the next lane may be predicted.

According to the data of the General Directorate of Security, 749.456 traffic accidents happened in Turkey in 2007. In these accidents 3459 people were killed and 149.140 wounded [1]. Human factor has the biggest role in accidents with 94% [2]. Almost 80% of accidents include driver inattention in the 3 seconds preceding the accident [3]. 60% of the road accidents are due to unintentional lane departures [4].

According to the Insurance Institute for Highway Safety status report, 3 million crashes per year could be mitigated or prevented by 5 new technologies [5]:

- Forward Collision Warning
- Lane Departure Warning
- Blind Spot Detection
- Emergency Brake Assistance
- Adaptive Headlights

1.2. Review of Literature

“Road edge departure without prior vehicle maneuver” and “vehicles changing lanes in the same direction” is the very frequent crash scenario causing high damage [5]. Therefore low cost lane departure, blind spot detection and time to collision warning system may be helpful to assist to the inattentive driver. Monitoring and warning systems are already available in the market. Mobileye system provides lane departure and front collision warning with one single camera [6].

Determination of the relative position between the cars, localization of the road on a basis of vision has been solved by many different methodologies. The essential problem related to the inverse optics, mapping transformation between the two dimensional image and three dimensional road scene has been solved by using flat road model assumption [7].

When the road is not flat, the stereo vision has been proposed to compute the height of lane markings with respect to the vertical axis [8]. To reconstruct the road curvatures, a B-Snake model is used for vision based lane detection and tracking [9]. Ridgeness has been used as low level image descriptor and an estimation method to fit the parabolas on the lane markings has been proposed in [10]. Vision based blind spot detection system has been proposed by using optical flow and data clustering techniques to detect the cars [11].

Edge detection, support vector machine learning and template matching has been fused to detect the cars and implemented on the image processing library Camillia [12]. An adaptive template matching algorithm has been considered for blind spot detection [13]. Radar supported Iterus AutoVue Vehicle Sensors are sold as different modules for lane departure, blind spot and forward collision warning. Another radar supported system GreenRoad provides lane changing assistance.

1.3. Driver Assistance Systems

Driver assistance systems (DAS) help drivers to drive safely. They are capable of understand the risks in a driving process and warn the driver generally through a human computer interface. They assist to the drivers during maneuvering tasks in challenging driving situations and also make the driving more comfortable and less costly. These systems are classified according to their customer relevant developmental aims [14]:

1. Economy DAS
2. Comfort DAS
3. Safety DAS

Economy DAS are used for an optimum cost driving process. An example of these systems is the automatic transmission with adaptive control. These transmissions enable using the best possible gear for every situation, so that fuel consumption reduces. DaimlerChrysler's Promote Chauffeur is another example for Economy DAS. It keeps the little distance in between two cars following each other and causes reduced aerodynamic drag and operating costs as well.

Comfort DAS aim to provide a comfortable driving for drivers. One of the examples for Comfort DAS are the Head-up displays in which the informations, navigations and warnings are mirrored onto the windscreen so that the driver can see them all without turning his head or averting his eyes from the traffic situation. Parking Assistants which help the driver to park safely and Adaptive Cruise Control Systems that keep a constant distance and relative speed between the forward vehicles (Figure 1.1) are another examples for Comfort DAS.

Safety DAS are designed to make driving safer. Some representatives of this kind of DAS are Adaptive Light Control that increases the illumination in the driver's vision range, Lane Departure Warning System that recognizes the road lanes and creates warnings while departing the lanes, ESP that keeps the car stabilized at the curves (Figure 1.2), Automatic Emergency Brake System that minimizes the speed in case of unpreventable accidents, Daimler Chrysler's Pre-Safe System that closes the windows, activates the belt pretensioners and brings the seats into their correct positions.

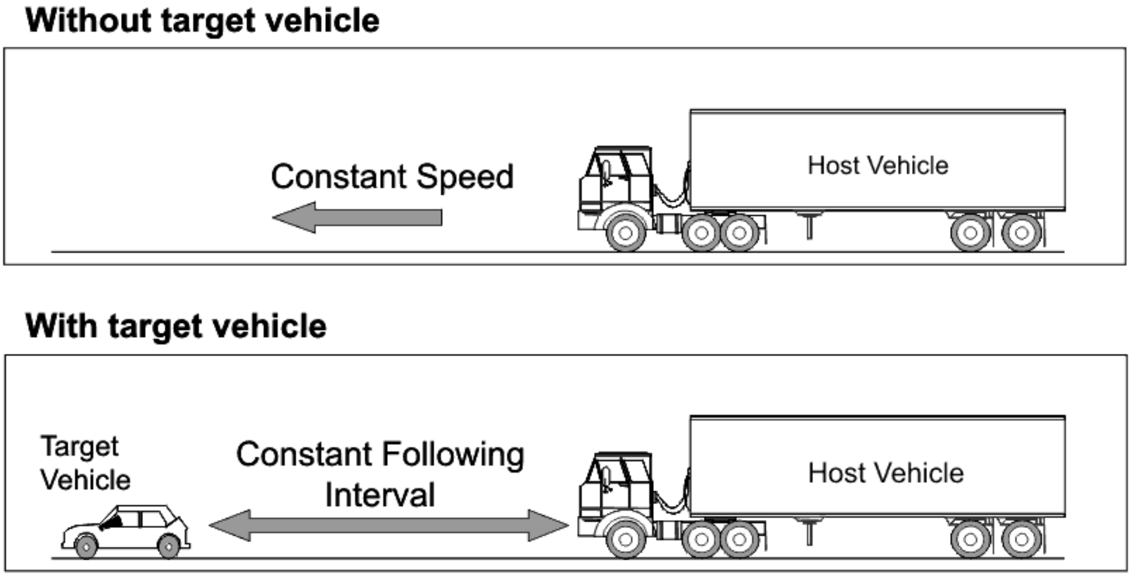


Figure 1.1: Principle of Operation of Adaptive Cruise Control [15]

1.4. Perception and Reaction

Reaction time for a perceived situation can be decomposed into some components. Sum of the time spent for each of these components is the reaction time. These components are [16]:

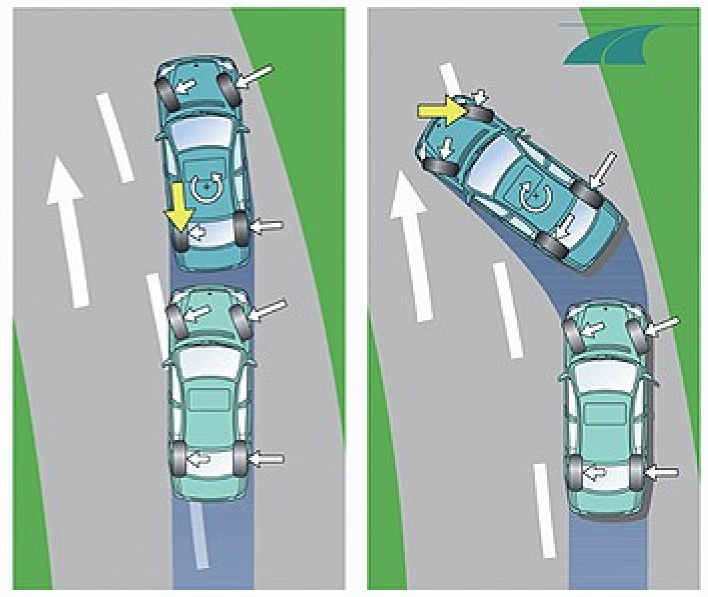


Figure 1.2 : Stopping the Slide: Around a left-hand bend, ESP systems can correct oversteer (front-end slide), far left, by applying the left-rear brake. Oversteer (rear-end slide), left, can be corrected by applying the front-right brake. [17]

I. Mental Processing Time

It is the time that the drivers spend to perceive the signal of a case and to decide the reaction for it. It can be decomposed into some components.

• **Sensation** : Time needed to notice the object, e.g. “An object is coming.”

• **Perception/Recognition** : Time needed to recognize the meaning of the sensation, e.g. “This object is a horse.”

• **Situational Awareness** : The time it takes to interpret the situation and understand what will happen if no action is taken, e.g. “If I don't reduce my speed, I will hit the horse.”

• **Response Selection and Programming** : Time needed to decide which action among the alternatives to take, e.g. “Instead of braking, I must steer left”. We can also call the mental processing time as the perception time.

II. Movement Time

The time it takes to move the muscles to react after deciding which reaction to show, e.g. To turn the wheel.

III. Device Response Time

A reaction can not give its results immediately. To accomplish a reaction, the vehicle needs time too. This time is called device response time, e.g. A vehicle does not stop immediately after braking.

1.5. Human Driver Model

The driver's perception and reaction time plays a significant role in all problems of active safety. The perception-reaction time of a driver is the time taken to detect the presence of an object; to identify that the object represents a potential or actual hazard; to decide the action to be taken in response to the hazard and to react or respond by starting to take that action. An introduction to this subject is made in first chapter. A simple human driver model is used to represent the reactions of the driver.

$$\frac{\delta_s}{\delta_d} = e^{-\tau s} \frac{1}{T_I s + 1} \tag{1.1}$$

where δ_s is the driver model steering wheel angle, δ_d is the desired steer angle, τ is the magnitude of pure time delay and T_I is the lead time constant. These driver model parameters based on average driver behavior are in the range

- $0.4 \le \tau \le 0.5$ seconds
- $0.11 \le T_I \le 0.29$ seconds

for lane tracking with the full-size car. In this study, the perception-reaction time is used to determine the time threshold to be waited by the presented active system before generating the audio/visual warning about the possible hazard.

Once the hazard is predicted, about one second of driver's reaction time is allowed and warning is removed if during this time the driver intends to react to the possible hazard by activating left, right turn indicator switch, braking or steering input [18] [19].

1.6. Main Deliverable

For a safer driving experience, different driver assistance systems satisfying several needs are produced and used. They are able to warn and monitor drivers after detecting some main reasons for an accident like lane departure, forward collision and blind spots. None of the commercial driver assistance systems include all these warning types in the same package. Moreover, these systems are very costly both for manufacturers and clients.

The main objective of this study is to develop vision-based warning system that may help to inform the driver about the potential hazards. Forward-looking camera assures lane departure in the left and right direction while providing the distance between the subject and lead vehicle traveling in the same direction. Sensing range of the front looking camera is 100 meters. In this sensing range, lane and front vehicle position is generated and compared with the displacement, velocity and the steering of the subject vehicle in both of the longitudinal and lateral direction to generate the warnings in advance about possible lane departure and possible front collision hazards. Blind spot detection is also assured by using two web cameras installed on the left and right mirrors. The vehicles approaching to the subject vehicle in the left and right adjacent lanes are sensed and when the subject vehicle's driver intents to change lane, audio and visual warning is generated about possible hazard related to lane changing in the path of the coming vehicle.

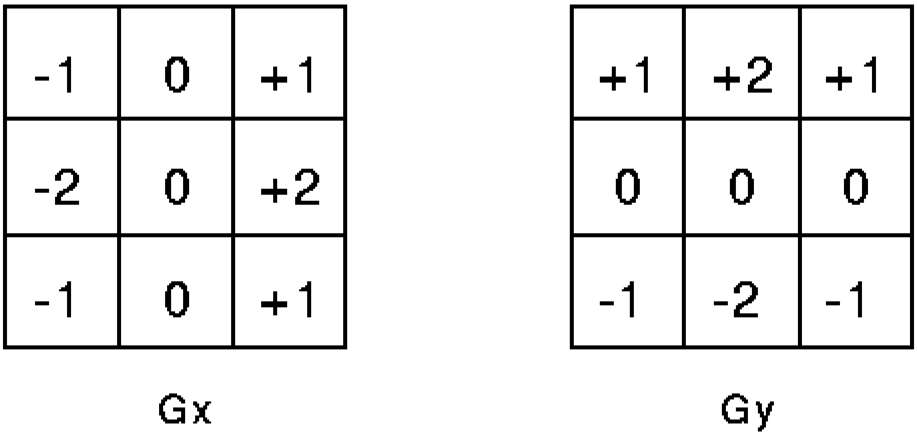
The sensing range of the visual based cameras are chosen such that the warning may be generated in advance and the driver may react to it before the hazards occur. The perception-reaction time of the average attentive driver is chosen to be one seconds. Therefore, sensing, computation and warning generation is achieved two seconds in advance with respect to the possibility of the hazard occurrence while letting the driver react to the warning.

2. REVIEW OF IMAGE PROCESSING TECHNIQUES

2.1. Sobel Edge Detection

The goal of the edge detection is to locate the points of a digital image which corresponds to a brutal change of luminous intensity. In general, the result of an edge detection process is a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings as well curves that correspond to discontinuities in surface orientation. So that image is cleared away from unimportant data.

One way to detect edges is to use Sobel operator. Sobel Edge Detection algorithm does not only calculate the magnitude of the edges but also their direction. Sobel operator uses two different 3x3 convolution kernels to convolve with the original image, one for horizontal and one for vertical changes (Figure 2.1).



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

Gx

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

Gy

Figure 2.1 : Sobel Convolution Kernels

Let the A denote the original image.

$$Gh = Gx * A \tag{2.1}$$

$$Gv = Gy * A \tag{2.2}$$

where the "*" is a two dimensional convolution operation, the combined gradient magnitude is computed as:

$$G = \sqrt{Gh^2 + Gv^2} \tag{2.3}$$

One can also calculate the gradient's direction using the formula given by Equation 2.4, [20]:

$$\Theta = \arctan\left(\frac{Gh}{Gv}\right) \tag{2.4}$$

Applying this algorithm to an image results with a new image with detected edges (Figure 2.2).

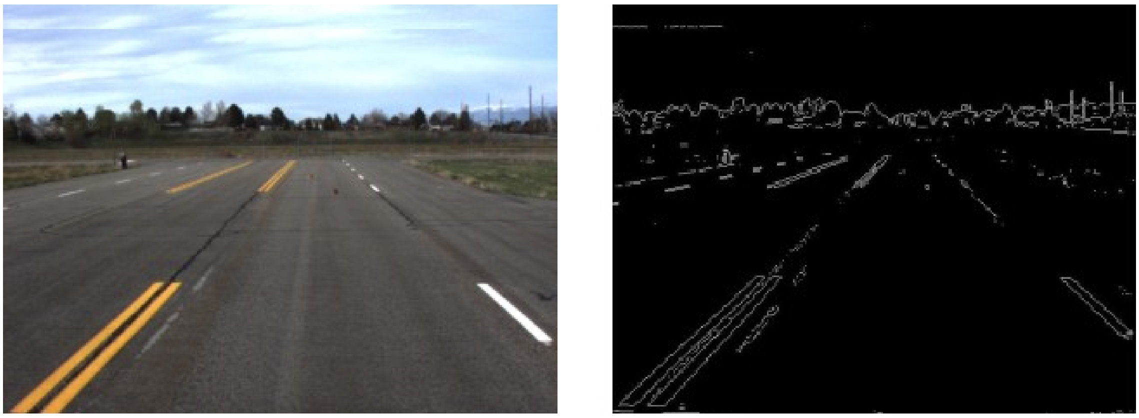


Figure 2.2 : Result of Sobel Edge Detection

2.2. Dilation

Dilation operator can be both applied to binary and grayscale images to gradually enlarge the boundaries of regions of foreground pixels. If it is applied to an image after edge detection operation, it will be easier to see the edges since they will grow in size.

In the dilation operation of grayscale images, they are considered as functions which map the Euclidean Space to real numbers. The second is a (usually small) set of coordinate points known as a structuring element (kernel). It is this structuring element that determines the precise effect of the dilation on the input image. Grayscale structuring elements are also functions of the same format, called structuring functions.

Denoting an image by *f(x)* and the structuring function by *b(x)*, the grayscale dilation of *f* by *b* is given by the formula below where "sup" denotes the supremum [21]:

$$(f \oplus b)(x) = \sup_{y \in E}[f(y) + b(x - y)] \tag{2.5}$$

Result of a dilation to an image with detected edges is shown in Figure 2.3.

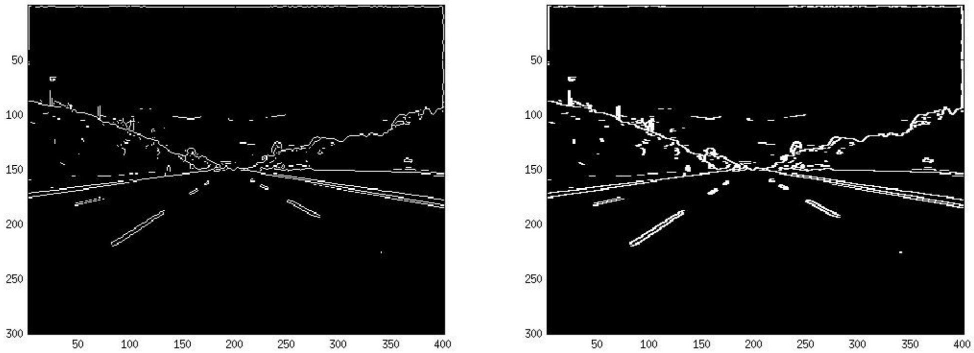


Figure 2.3 : Dilation of Edges

2.3. Conversion of Image Types and Thresholding

A RGB image is stored as a m-by-n-by-3 data which defines red, green, blue components for each pixel, where the image size is mxn. A grayscale image has only the intensity information and it has a color palette that consists of different shades of gray.

To convert a RGB image to grayscale, a uniform weighted average of the values for each color component is computed. Let r, g and b denote the values for color components in a pixel, red, green and blue respectively. Then the grayscale value gray for that pixel is calculated as:

$$\text{gray} = \tfrac{1}{3}\,(r + g + b) \qquad (2.6)$$

To convert a grayscale image to a RGB image, grayscale pixel value of the image is assigned to red, green and blue components of corresponding pixel in the new image. That is the values of r, g and b will be equal to value of gray.

Thresholding is used to convert a grayscale image to a binary image that is composed of only black and white pixels. If a pixel value in the grayscale image is less than a threshold, that pixel will get a value of zero in the new binary image and value of one if it is greater.

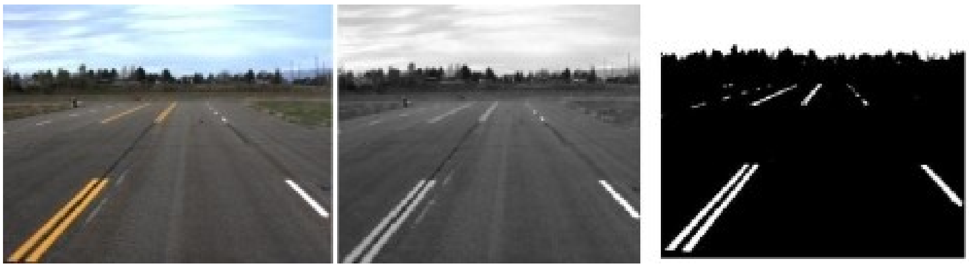


Figure 2.4 : Same image as RGB, Grayscale and Binary Respectively

2.4. Gaussian Smoothing Filter

The Gaussian smoothing operator is a two dimensional convolution operator which is used to blur images and remove details and noises. The Gaussian distribution in two dimensions has the form :

$$G(x)=\frac{1}{\sqrt{2\pi}*\sigma}e^{-\frac{x^2}{2\sigma^2}} \qquad (2.7)$$

where σ denotes the standard deviation and *x* and *y* denotes the horizontal and vertical distances from the origin of the image. When this formula is applied to the 2D image frame, it produces a surface whose contours are concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is applied to the original image. Each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases [22].

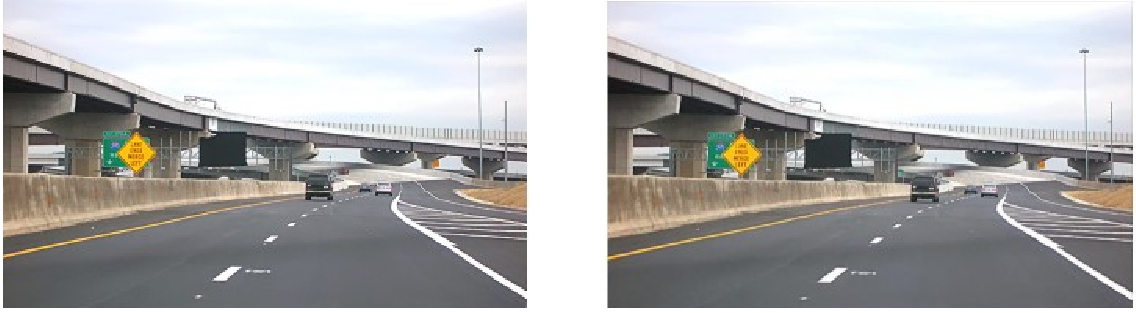


Figure 2.5 : Effect of Gaussian Smoothing Filter : Gaussian filter applied to first image, results the second image.

2.5. Hough Transform

Hough Transform is used to recognize the shapes in a digital image. Recognition of lines in an image is the simplest application that can be done with Hough Transform. There are infinite numbers of lines passing through one point and the only difference between these lines is the angle i.e. the orientation. The aim of this transform is to determine which of the lines are fitting into a given scheme and to find the points which belong to the same line potentially. A representation of this transform is shown in Figure 2.6.

In Hough Transform, each line is a vector having parametric coordinates θ and ρ where θ is the angle and ρ is the length of the segment starting from the origin which is perpendicular to the line of angle θ. By transforming all the possible lines which connect one point to another i.e. calculating the value of ρ for each θ, we obtain a unique sinusoid in (ρ, θ) plane which is called the Hough Space, if $\theta \in [0, \pi]$ and $\rho \in R$, or if $\theta \in [0, 2\pi]$ and $\rho \geq 0$.

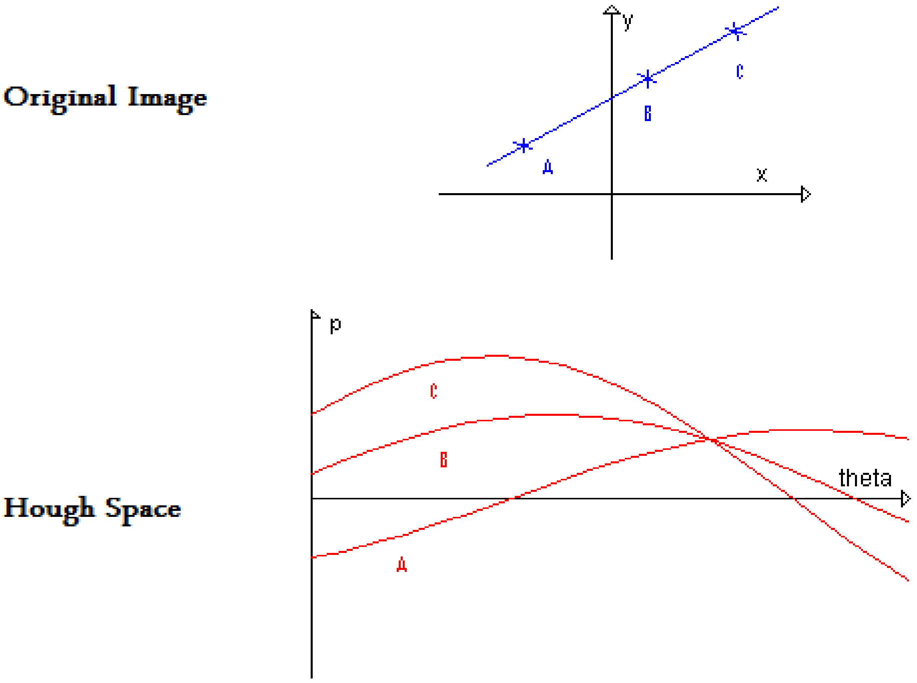


Figure 2.6 : Hough Transformation

The relation between ρ and θ is shown by the formula given below [23]:

$$\rho = x\cos\theta + y\sin\theta \tag{2.8}$$

If the curves associated with two or more points intercept each other, the location of this interception in the Hough Space, corresponds to the parameters of the line which connects these points. In the image frame, these parameters are the angle and the distance between the image center and the line.

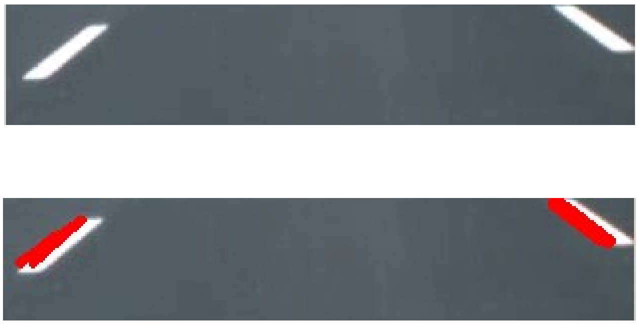


Figure 2.7: Result of Hough Transform on a Road Segment Image

2.6. Template Matching

Given a template image, it is possible to find small parts of a source image that match the template image by using Template Matching algorithm. The idea behind it is that to compare the neighborhood of each pixel of the source image to a template. The best match which has the largest strength, the n largest strength or all matches that exceed a threshold value are chosen.

To compare the neighborhood of each pixel to a template is done by shifting the template to each point in the source image and dot product of each neighborhood with the template. If source image, template, result, width and height of template are denoted by I, T, R, x' and y' respectively, the formula to compute the result is:

$$R(x,y)=\sum_{x',y'}[T(x',y').I(x+x',y+y')] \tag{2.9}$$

where x' = [0, …, width - 1] and y' = [0, …, height - 1].

Searching over the whole result array for the maximum elements will yield the matched part of the source image to a template (Figure 2.8).

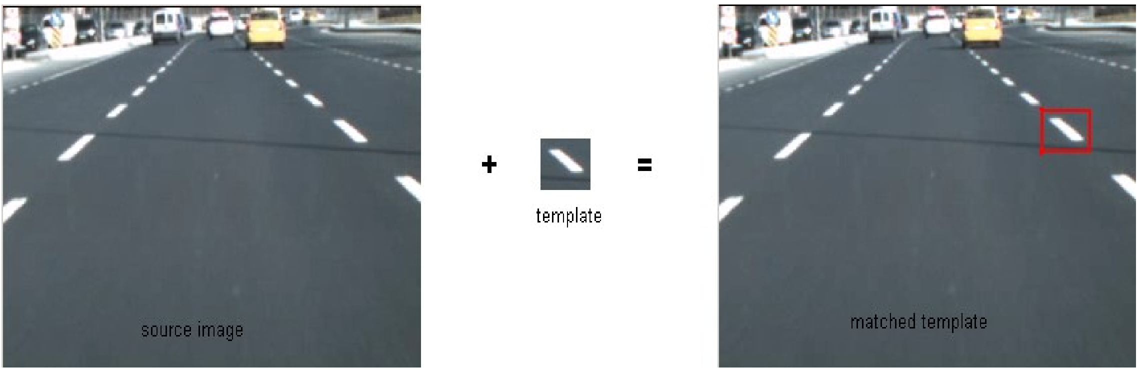


Figure 2.8: Template Matching Example

2.7. Blob Detection

The algorithm used for Blob Detection was proposed by Chang et al. [24]. It is based on detecting external and internal contours. This part explains briefly how blob detection works.

For a binary image I, the blob search is done as follows. The image I is accompanied with another image L which stores the label information. A dummy row consisting of white pixels is added into I. Each column in each row is searched for a black pixel.

The first encountered black pixel if it has a white pixel above is considered as a member of an external contour. Then contour tracing method is applied to find and label all points of that external contour. If the preceding pixel is labeled, current pixel is unlabeled and there is a white pixel below, it is considered as a new internal contour and labeled as the preceding pixel. Otherwise it is another point of external counter.

Contour tracing is applied once again for the internal contour (Figure 2.9). Surrounding pixels of an external contour are labeled with a minus sign, labeled black pixels are labeled with a different integer. Labeled or unlabeled black pixels and white pixels have different labels to indicate their status. Contour tracing is done by searching the 8 surrounding pixels for the same label as the label of the current pixel until arrival to the starting point.

2.8. Optical Flow

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [25].

The optical flow methods try to calculate the motion between two image frames which are taken at times t and t + δt at every voxel position. These methods are differential since they are based on local Taylor series approximations of the image signal, that is: they use partial derivatives with respect to the spatial and temporal coordinates.

For a 2D+t dimensional case (3D or n-D cases are similar) a voxel at location (x,y,t) with intensity I(x,y,t) will have moved by δx, δy and δt between the two image frames,

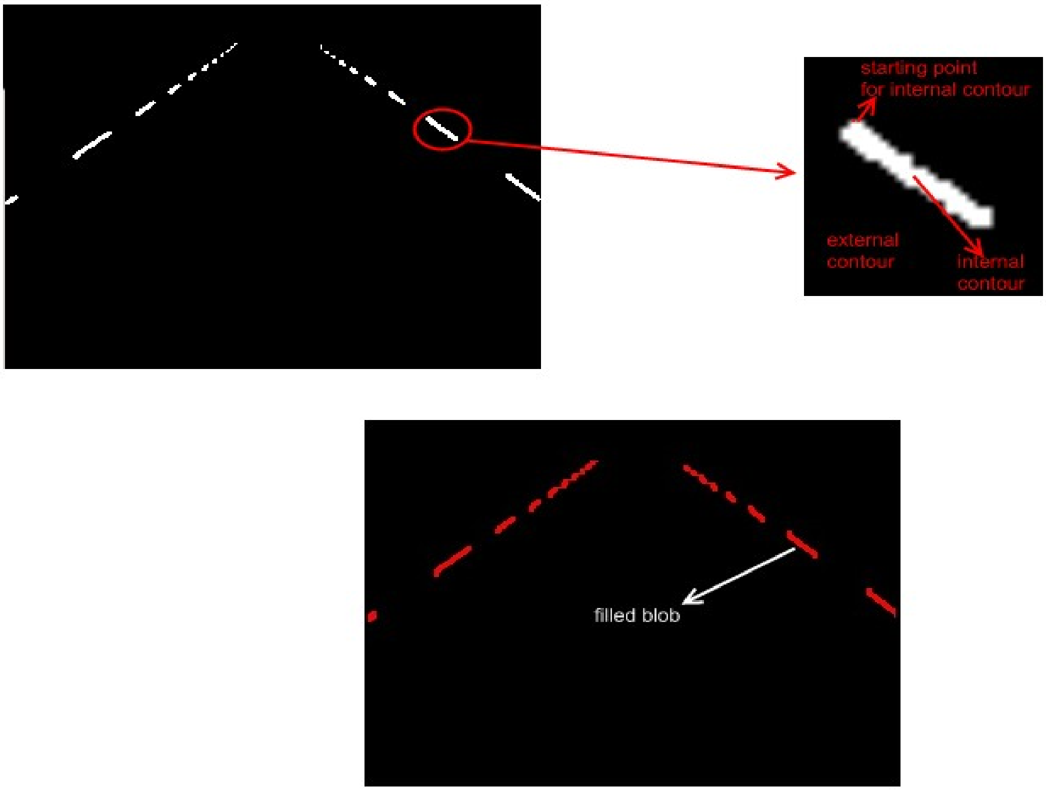


Figure 2.9: External and Internal Contours on an Image

and the following image constraint equation can be given:

$$I(x,y,t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{2.10}$$

Assuming the movement to be small, the image constraint at *I*(*x*,*y*,*t*) with Taylor series can be developed to get:

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + H.O.T \tag{2.11}$$

where V_x,V_y are the *x* and *y* components of the velocity or optical flow of *I*(*x*,*y*,*t*), I_x, I_y

and I_t can be written for the derivatives in the following.

$$IxVx + IyVy = - It \qquad (2.12)$$

This is an equation in two unknowns and cannot be solved as such. This is known as the aperture problem of the optical flow algorithms. To find the optical flow another set of equations is needed, given by some additional constraint.

Lucas-Kanade method is used to provide this additional constraint. In computer vision, the Lucas–Kanade method is a two-frame differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade. It introduces an additional term to the optical flow by assuming the flow to be constant in a local neighborhood around the central pixel under consideration at any given time [26].

The additional constraint needed for the estimation of the flow field is introduced in this method by assuming that the flow (Vx,Vy) is constant in a small window of size mxm with m> 1, which is centered at pixel x,y and numbering the pixels within as 1...n, $n = m^2$, an equation can be found:

$$A\vec{v} = -b \qquad (2.13)$$

where A is the pairs of I_x, I_y , $\vec{\upsilon}$ is the velocity vectors and b is the I_t series.

Solving this equation results that the optical flow can be found by calculating the derivatives of the image in along all dimensions.

3. DRIVER WARNING SYSTEM DEVELOPMENT

3.1. Introduction

There is an obvious need for a complete driver warning system which provides the main warning functions for inattentive lane departures, blind spots and possible forward collisions. In order to satisfy this need, development of a new real time driver warning platform is mentioned in this chapter. System is divided into subsystems and developed algorithms are explained, respectively.

3.2. Lane Departure Warning

This subsystem is able to detect road lane markers, show their borders and warn the driver when the vehicle trajectory exceeds the borders. The algorithms applied for each RGB image captured from the forward camera is explained in this part of chapter.

Lane detection is based on the image processing techniques applied to the edges in an image frame. So, the first thing is to convert the RGB image (Figure 3.1) to gray level (Figure 3.2) for being able to use Sobel Edge Detection algorithm to find the edges (Figure 3.3).

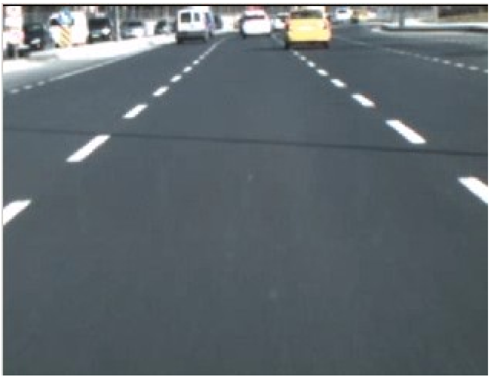


Figure 3.1: RGB Image Captured from the Forward Camera

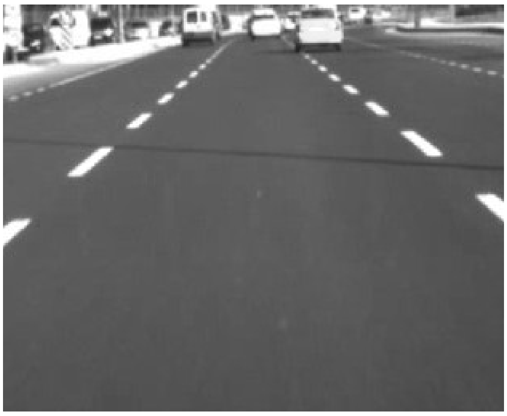


Figure 3.2: Gray Level Conversion Result of an RGB Image

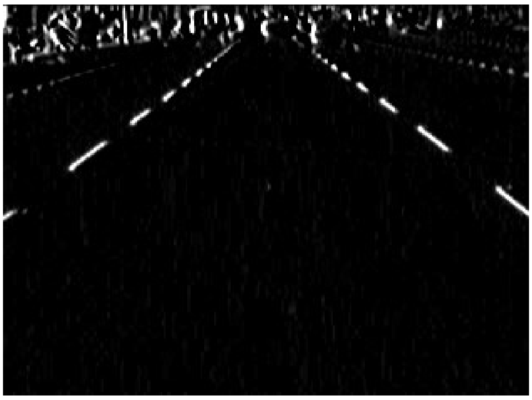


Figure 3.3: Sobel Edge Detection Result

Obtained image is converted into binary image using thresholding operation with parameter 100 (Figure 3.4).

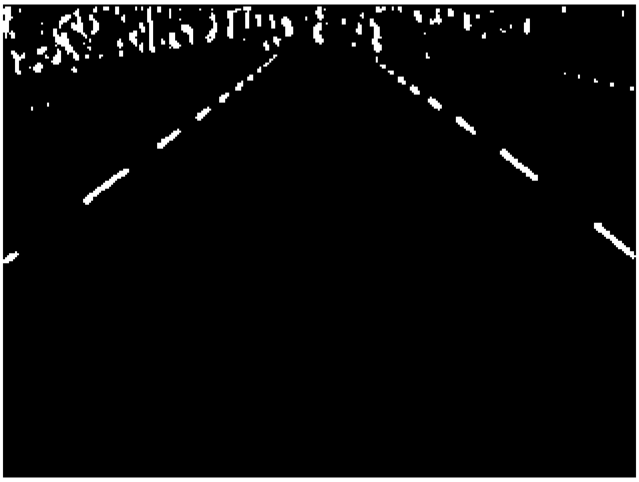


Figure 3.4: Thresholded Image with parameter 100

To remove noises, each column in each row is searched for two or more neighbor white points in the same row. Such group of white points are replaced by one single white point in the middle of the group (Figure 3.5) (Figure 3.6).

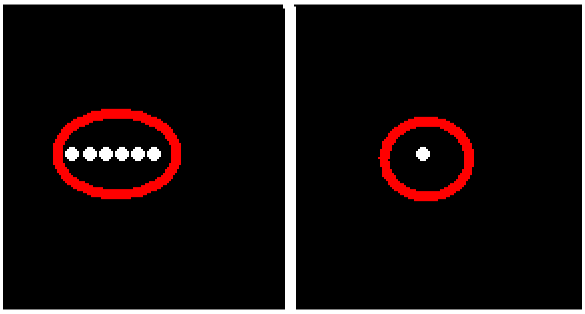


Figure 3.5: Filtering Process : Before filtering (left), after filtering (right)

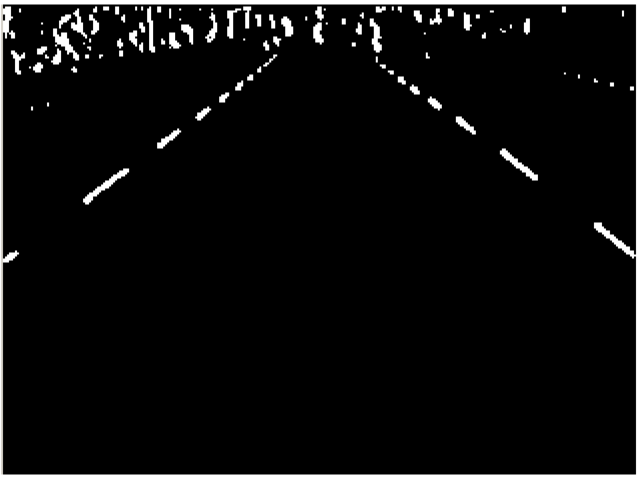


Figure 3.6: Result of Filtering Process

Then a morphological open operation is applied by dilating and eroding the image (Figure 3.7). After elimination of the noises, the image becomes available for lane detection.

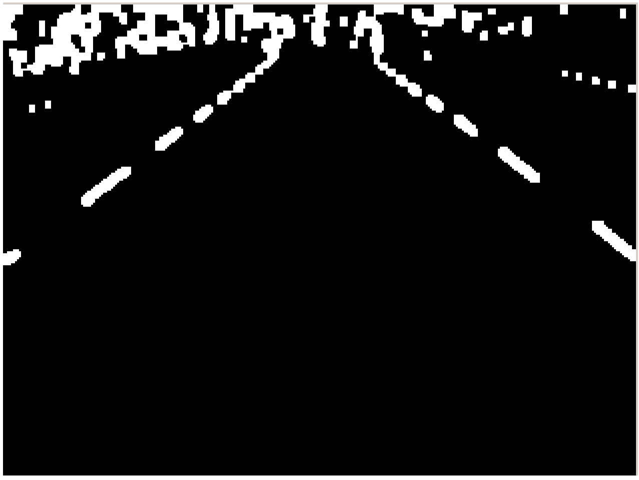


Figure 3.7: Image After Morphological Open

Proposed approach brings two image processing algorithms together: Blob Detection and Hough Transform.

Firstly, a blob detection is done in the region of interest which excludes the upper one ninth and one quarter from both right and left of the prepared image. Blobs having an area between 15 and 1000 pixel square are selected and marked in red while others are deleted (Figure 3.8).

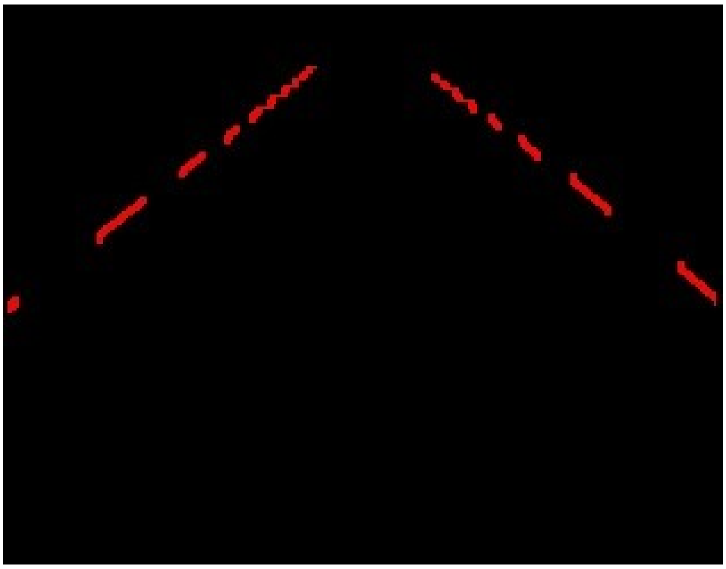


Figure 3.8: Blobs Found in the Image

Secondly, gray level conversion, edge detection, dilation, gauss smoothing filter and thresholding are applied to another copy of the captured image. Obtained image is divided into 5 segments proportional to 1, 2, 3, 4 and 5 respectively (Figure 3.9). Segmentation allows us to divide non linear lanes into smaller linear lanes so that the hough transform can detect curves.

For each segment, probabilistic hough transform is applied. Hough transform parameters are subject to change according to the size of the segment. Because the length of lines and the gap between consecutive lines vary in each segment. Detected lines are marked in red and filtered by the absolute value of their slopes (Figure 3.10).

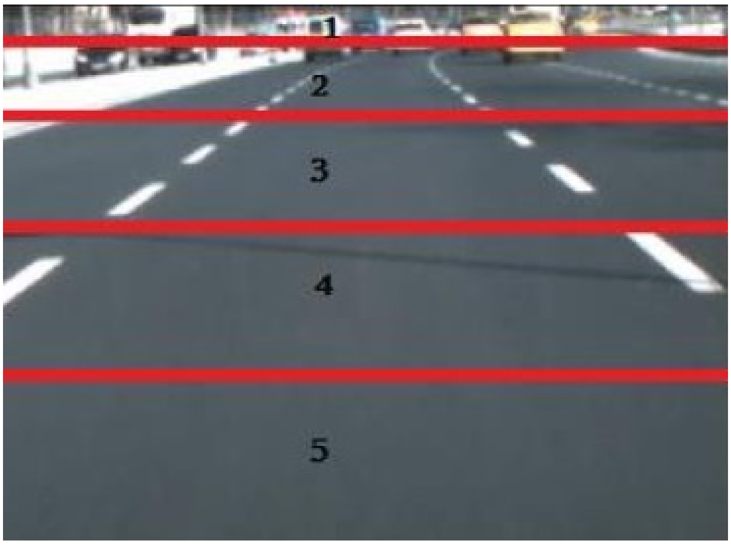


Figure 3.9: Borders of Segments in an Image

The lines from the forth and fifth segment are copied into an empty image with a black background. An appropriate line is assumed to have an absolute value of slope between 0.3 and 0.9.

Two images obtained from hough transform and blob detection are logically combined to create a new single image (Figure 3.11). This yields a robust detection of lanes. Hough transform is once again applied to new image so that the lines are detected.
Lines located on the right side of middle point of image are marked in blue and the ones on the left side in red (Figure 3.12).

Then the obtained image with possible lane candidates is searched for blue pixels from middle towards right and for red pixels towards left. The first encountered point in each row for each direction is pushed into stacks dedicated to left and right points. So that, points on the left and on the right are stored in two different stacks. Points found marked in white in Figure 3.12.

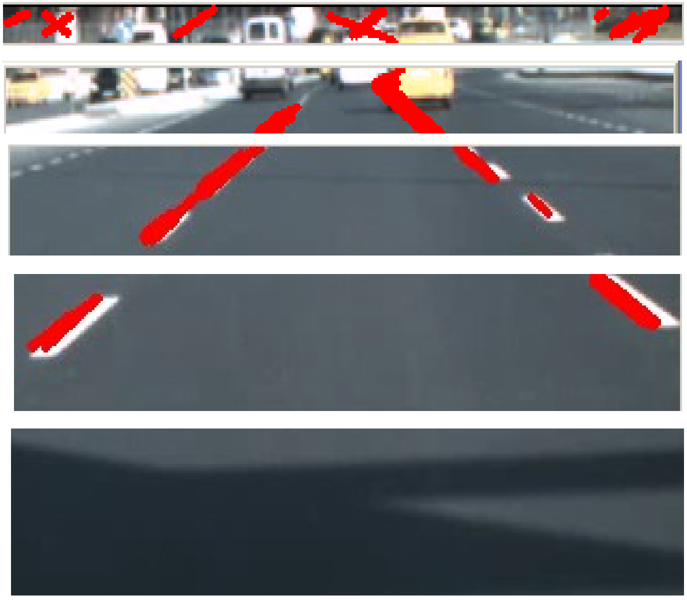


Figure 3.10: Results of Hough Transform in Image Segments

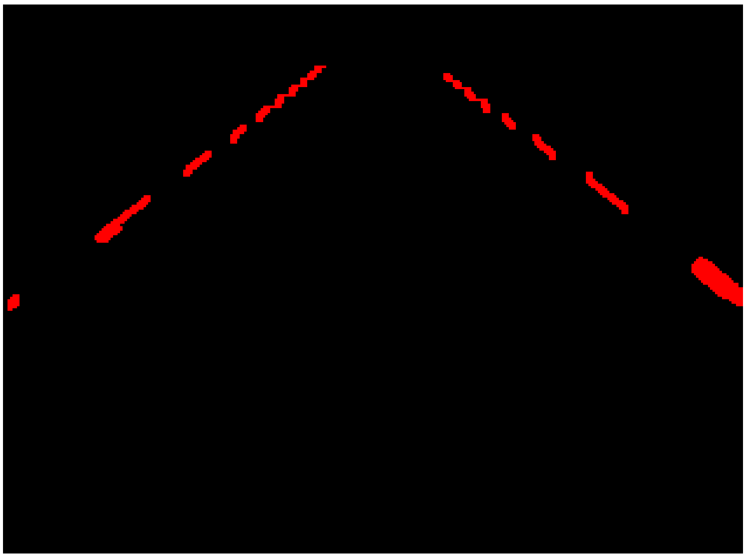


Figure 3.11: Logically OR Operation for Blobbed and Hough Transformed Images

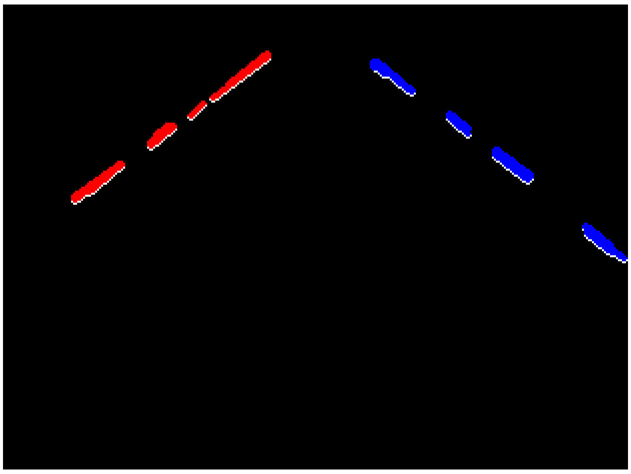


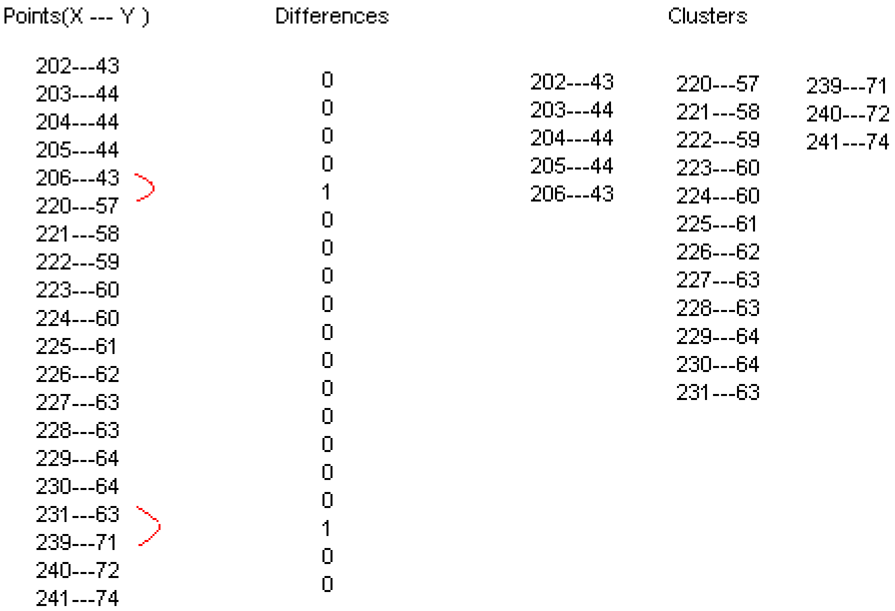
Figure 3.12: Image Obtained with Possible Lane Candidates

For each stack, stored points are separated into clusters by calculating their coordinate differences by the formula below:

$$\Delta=((x(k)-(x(k-1)^{2}))+(y(k)-y(k-1)^{2}))^{(1/2)} \tag{3.1}$$

If the difference between two consecutive stack elements *k* and *k-1* denoted by *Δ* is greater than a threshold value, 1 is pushed into a different stack for differences and 0 otherwise. So the index value of a 1 indicates the index value of the border of a cluster in clusters' stack (Figure 3.13). Therefore, the new point is considered to be an element of a new cluster. To prevent excessive number of cluster, each cluster is enforced to have a minimum number of consecutive points.

Two other regions of interest is determined as upper one tenth and one sixth from both right and left. Same search for red and blue pixels is done and two new stacks are created to store the points of the curves on the left and right.



```
Points(X --- Y )      Differences              Clusters

   202---43
   203---44               0          202---43     220---57    239---71
   204---44               0          203---44     221---58    240---72
   205---44               0          204---44     222---59    241---74
   206---43               0          205---44     223---60
   220---57               1          206---43     224---60
   221---58               0                       225---61
   222---59               0                       226---62
   223---60               0                       227---63
   224---60               0                       228---63
   225---61               0                       229---64
   226---62               0                       230---64
   227---63               0                       231---63
   228---63               0
   229---64               0
   230---64               0
   231---63               0
   239---71               1
   240---72               0
   241---74               0
```

Figure 3.13: Clustering the Points According to Distance

All those points are used to generate second order polynomials passing through the lane borders. To find the coefficients of polynomials, three points are needed. Once point sets under form of clusters are obtained, the number of clusters are calculated. If three or more clusters exist in a stack, then clusters are checked with each other. This check is done to avoid having false clusters. Checking starts with a initialization. If the coefficients of the polynomial created by average value of the first part of first cluster, average value of whole first cluster and average value of second cluster are close to the coefficients created by the average value of the first part of first cluster, average value of first cluster and average value of third cluster, all three clusters are pushed into final stack. If not, clusters having negative second coefficient are pushed into final stack for right side and having positive second coefficient into final stack for left. After initialization, the current points in the final stacks are considered as single clusters and its average on first part, its average and the average of the next cluster gives the coefficients recursively so that the new cluster is added into stack if it creates an appropriate line with the old clusters. Controlling the clusters allows to eliminate erroneous line segment candidates.

For the left and right straight lines, three points from the final stacks are selected. These are the first, middle and last elements of the stacks. Coefficients for lane border are calculated by matrix operations. To find the coefficients of a second order polynomial given three points as (x, y) pairs, the matrix operation is explained below.

Let X, C and Y matrices be,

$$X=\begin{bmatrix} x_0^2 & x_0 & 1 \\ x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \end{bmatrix} \qquad C=\begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} \qquad Y=\begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \tag{3.2}$$

These three matrices satisfy the Equation 3.3:

$$X\,C = Y \qquad ==> \qquad C = \mathrm{inv}(X)\,Y \tag{3.3}$$

where inv() denotes the matrix inverse.

By solving this equation after finding the inverse of matrix A, resulting B matrix keeps the coefficients for the polynomials passing through the given points. Then these polynomials are in the form:

$$y=c_0 x^2+c_1 x+c_2 \tag{3.4}$$

The coefficients found are checked again. All coefficients must assure the inequality below to maintain the stability:

$$0.5 * Cip < Cic < 2 * Cip \tag{3.5}$$

where i indicates the order of coefficient, p and c indicates previous and current frames respectively.

Also the second coefficient must be positive for left sider lanes and negative for the right siders. The third coefficient must be positive and less than 1500 for both sides. Successful coefficients are used to plot the lane borders and pushed into a stack which stores the successful coefficients of last five frames. If coefficients can not pass the control or there are no polynomials available, then the average value of coefficients from the last five frames are used.

Generated polynomial for left straight line is evaluated from 0 to x value of the last element of the stack of left sider points for left lane and generated polynomial for right straight line is evaluated from the rightmost corner's x value to the last element of the stack of right sider points for right lane (Figure 3.14). Resulting (x, y) pairs are marked in green on the image (Figure 3.15). Each time the length of the line is the mean value of the length in previous and current frames to maintain stability.

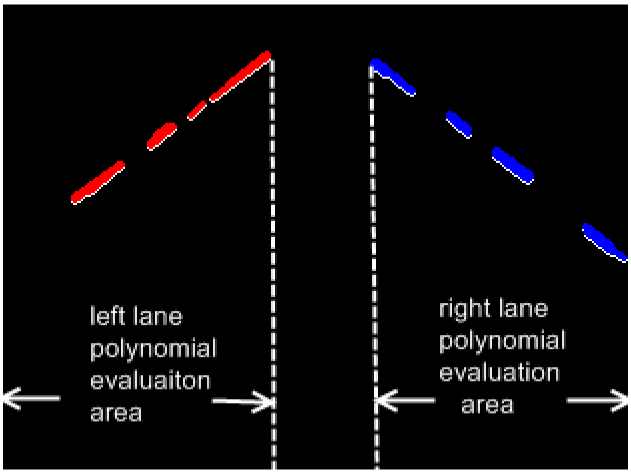


Figure 3.14: Polynomial Evaluation Areas on the Image

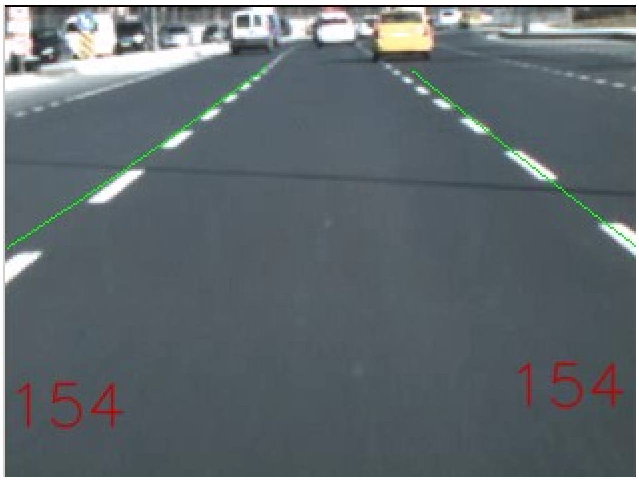


Figure 3.15: Fitted Polynomials for Lane Borders

For curved roads, arc drawing approach is proposed instead of polynomial approach. The curve on the right side is plotted by using the midpoint of the stack of right sider curve and the last plotted point on the right. A virtual radius is drawn from the rightmost bottom corner to the mid point of the stack of right sider curve. Angle between this radius and bottom of image is calculated by first calculating the tangent of the radius. Then arc tangent is used to find the angle. Same process is applied to find the angle of the line drawn from the last plotted point of right straight line. End point of the radius redrawn for each angle value is marked in red on the image along the twice of difference between these two angles with an increment of 0.1 degrees. The arc drawn fits the curve on the right side (Figure 3.16).

The curve on the left side is plotted by using the same way. This time points of the left side stacks are used to draw the arc.

If one of the right or left curve is not available, then the copy of the available arc is shifted to the opposite direction. To calculate the shifting distance, a horizontal line passing through the middle of the image is checked to find in which x values it intercepts the right and left polynomials. The difference between x values are multiplied by the first curve point in the direction that has an available curve and then divided by the half of the height of the image. That is the similarity of triangles are used to calculate the width between the two lines near curves. The curves found are drawn for next 5 frames unless a new curve is available (Figure 3.17) (Figure 3.18).

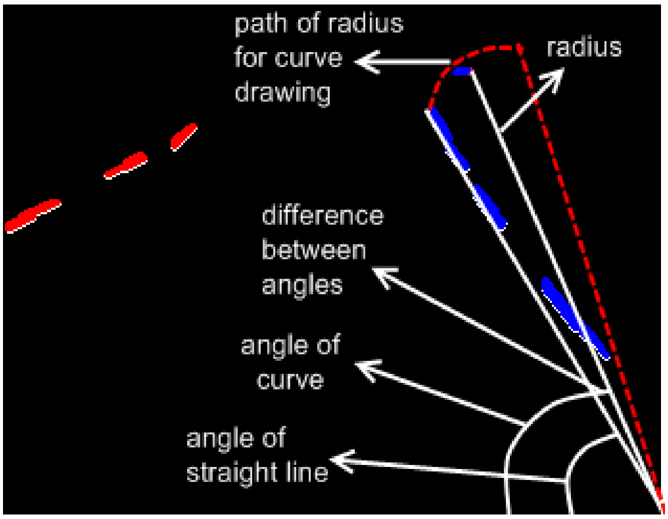


Figure 3.16: Angles for Arc Drawing

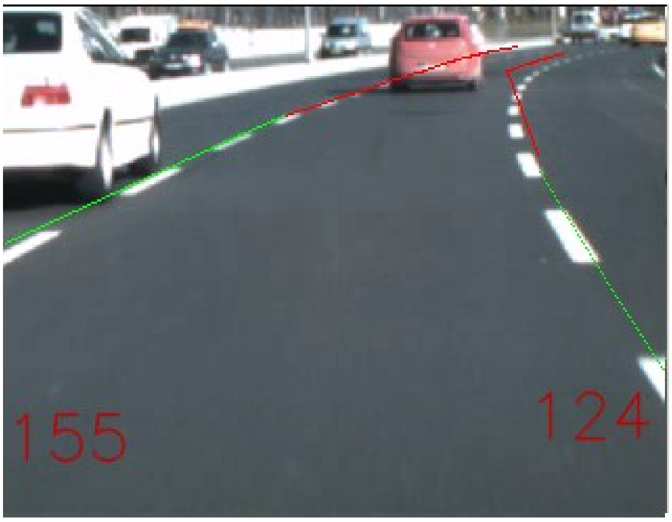


Figure 3.17: Lane Borders on a Curved Road

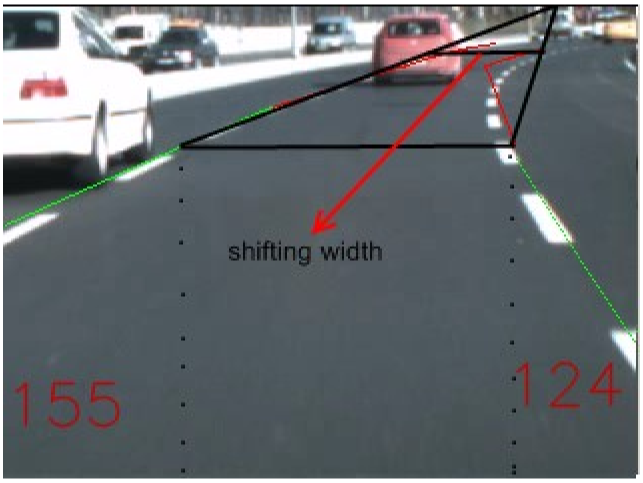


Figure 3.18: Shifting the Curve

After detecting the lanes, another process is applied to understand whether vehicle moves out of its lane.

The interception points of the left and right polynomials and the horizontal line passing through the middle of the image are subtracted from the half of the width of the image. Absolute values of these subtractions yield the distances to lane crossing for both directions (Figure 3.19) (Figure 3.20).

If the ratio of these two subtractions is more than 2, this indicates a lane departure to the direction where the subtraction has the smaller value. But to be sure, keeping track of the last 5 cases can avoid failure. An array of indicators stores the last 5 alerts in binary form on whether there is a departure or not. If at least 4 of the last 5 alerts are positive that is the average of the elements in array is greater than 0.8 then the necessary signals are sent to warn the driver.

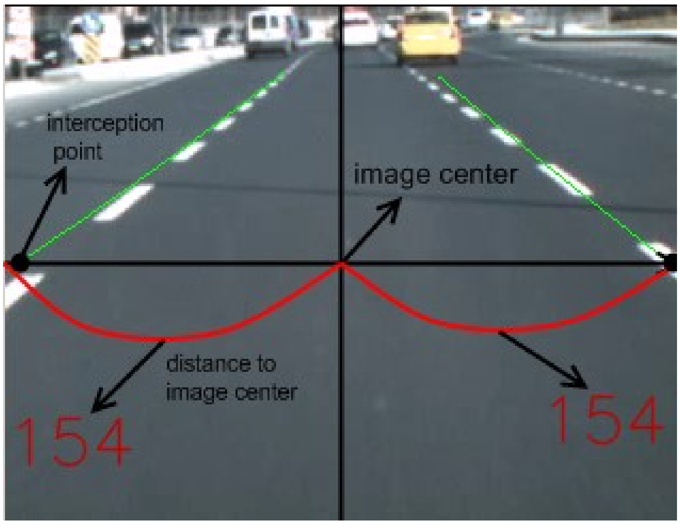


Figure 3.19: Calculation of the Lane Departure

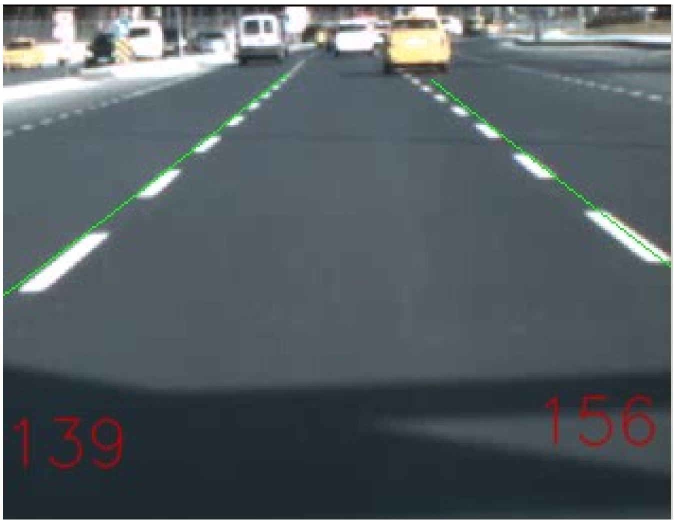


Figure 3.20: Result of Lane Detection

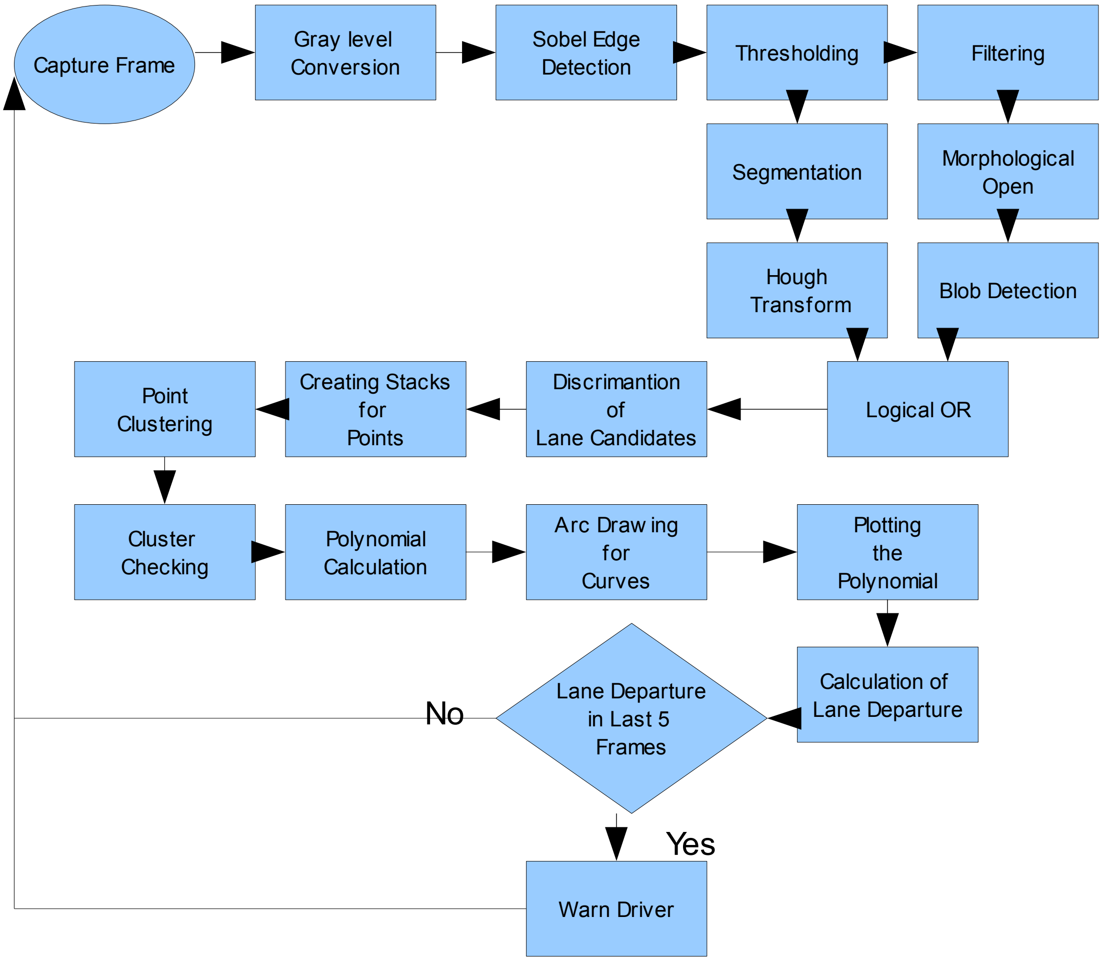


Figure 3.21: Flowchart of Lane Departure Warning Subsystem

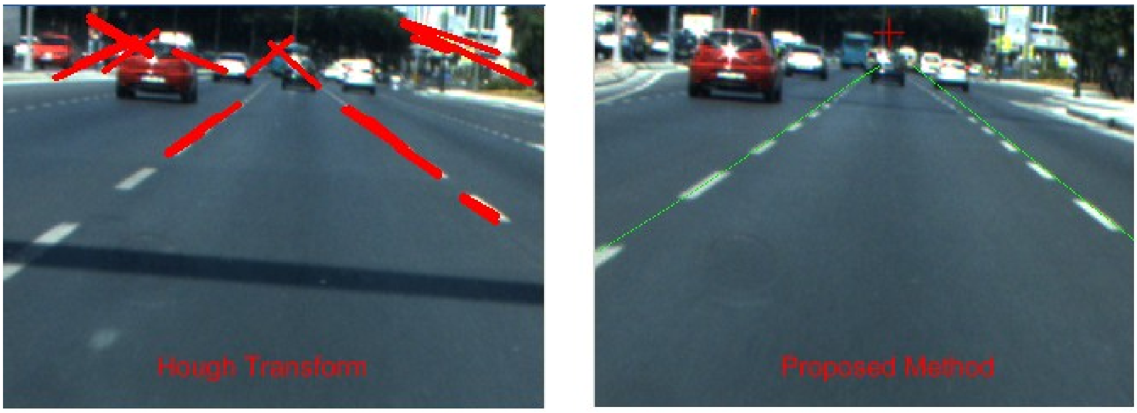


Figure 3.22: Hough Transform vs. Proposed Method

3.3. Forward Collision Warning

This module allows driver to see the distance from the forward vehicle and to be warned if this distance is below a threshold value. For this module to be run properly, the first step is to detect the forward vehicle. System uses the shadow under the forward vehicle to detect it. During daytime and even nighttime, there is always a shadow under the vehicle. Since this shadow has a low brightness, in a binary image it is a white region even with a low threshold value. Because of the road surface is always black in a well thresholded binary image, any object on the surface creates an edge and is seen as a white region.

While forward view camera capturing images, for each captured image, system first converts the image into grayscale (Figure 3.23). Then, Sobel Edge Detection and Dilation processes are applied. The dilated image converted to binary image. System uses 160 as the default threshold value (Figure 3.24). A predefined binary shadow template image is searched by using Template Matching algorithm (Figure 3.25) in a region of interest which covers the headway of the vehicle on this binary image (Figure 3.26). This and the other image processing algorithms used in this module are mentioned in Chapter 2. After finding the best match with template, the vehicle is bounded with a rectangle on display screen.

The next step is to calculate the distance from the forward vehicle. The distance between the center point of the bounding rectangle and the middle bottom point of the whole frame, is calculated in pixels (Figure 3.27).

To convert the distance from pixel to meters following formula is applied:

$$DM = IW \times OP / IP \qquad (3.6)$$

where DM is the distance in meters, IW is depth of image, OP is the distance in pixels and IP is the height of the image in pixels. Depth of image is the real distance in meters from the camera to the last point the camera sees further.

Then the difference between current distance and the distance in the previous frame is divided into the relative speed of the vehicles which yields the TTC, i.e. Time to Collision. TTC indicates the time left for a possible collision if the relative speed is constant. The relative speed of the vehicles is calculated as pixels per second by the formula:

$$(Dc - Dp) \times F \tag{3.7}$$

where the Dc, Dp and F denote pixel distance in current frame, pixel distance in previous frame and frame rate of the forward view camera respectively.

For the CAN supported system, distance of the vehicle is divided by the relative speed of the vehicles by obtaining the speed information from the CAN and forward vehicle speed from its displacement to calculate TTC.

If the TTC and the pixel distance are both less than threshold values, which are 2 seconds for TTC and 2 meters for pixel distance, a visual and audio warning is given by the system.

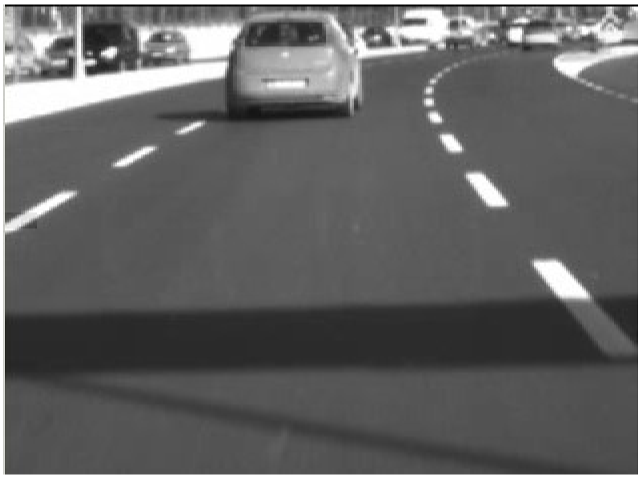


Figure 3.23: Image Captured for Forward Vehicle Detection

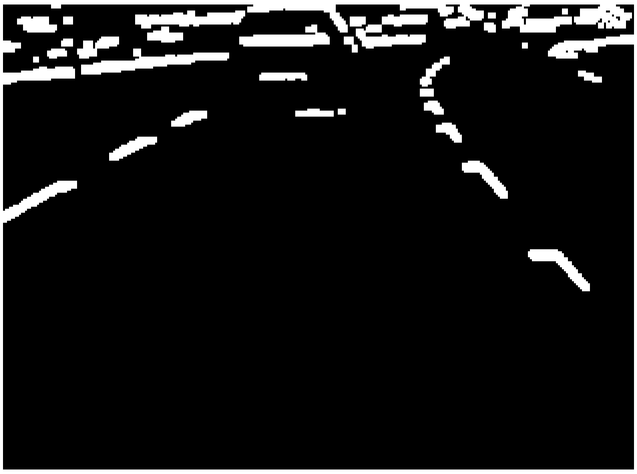


Figure 3.24: Binary Image for Forward Vehicle Detection

Figure 3.25: Shadow Image Template

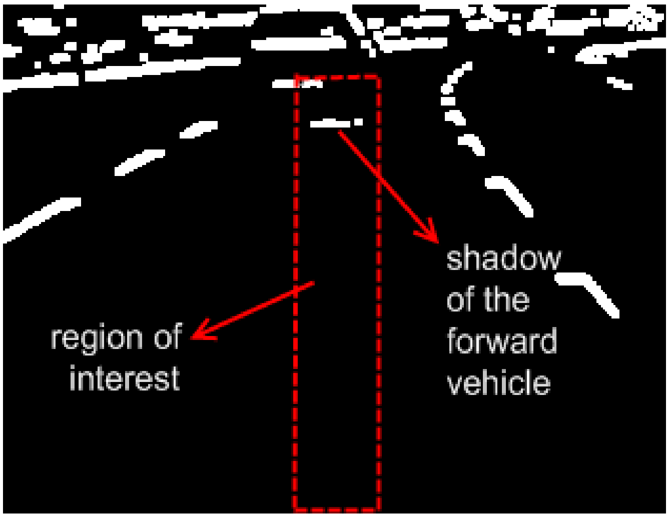


Figure 3.26: Region of Interest to Search the Template

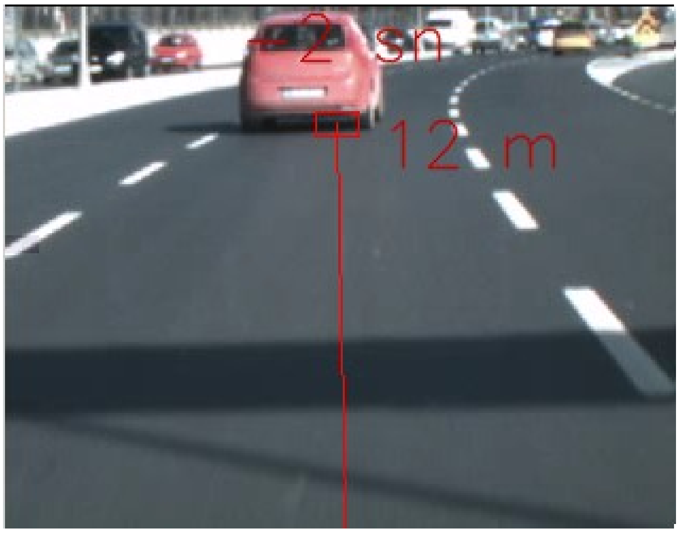


Figure 3.27: Vehicle Found on the Image

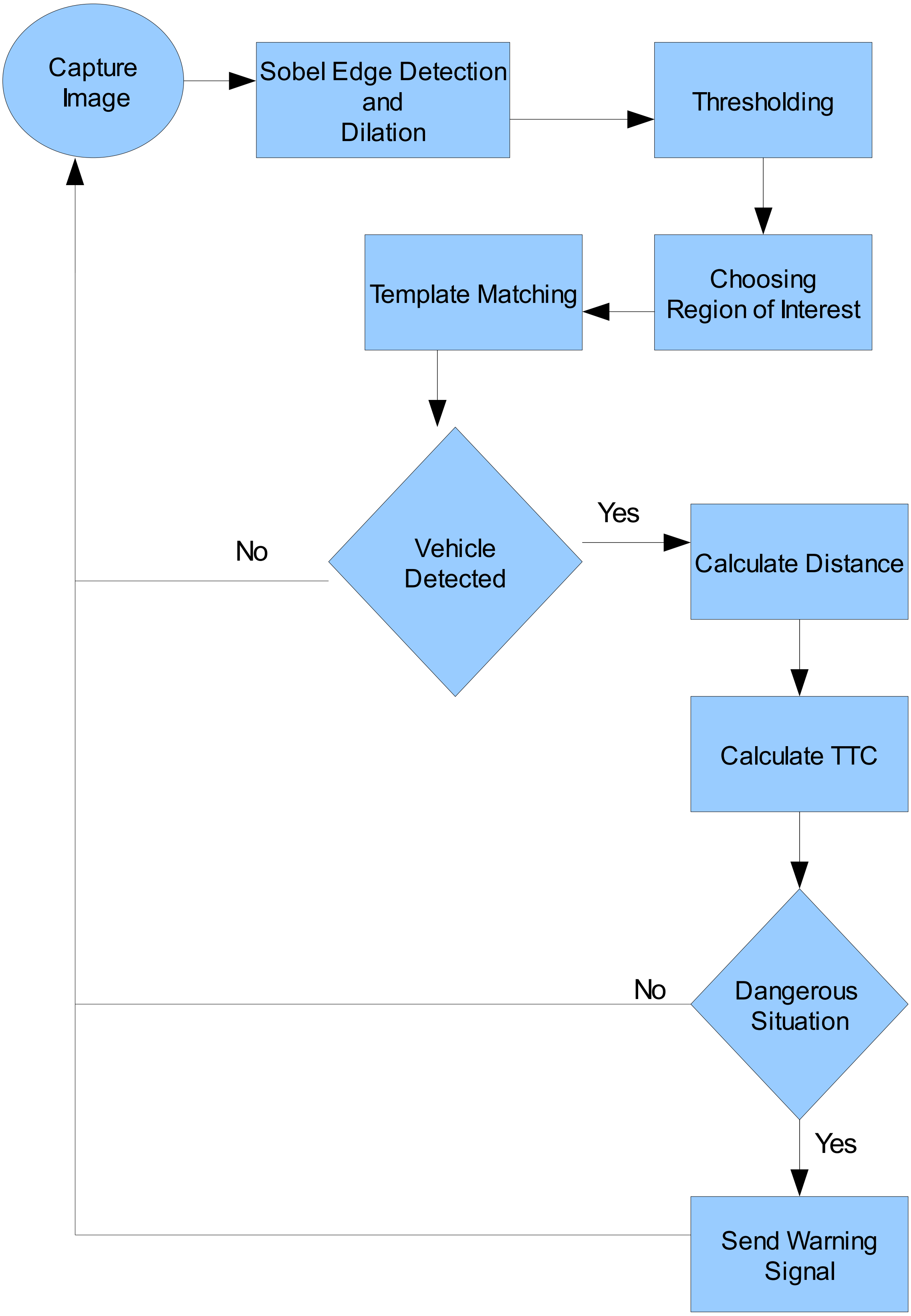


Figure 3.28: Flowchart of the Forward Collision Warning Subsystem

3.4. Blind Spot Warning

The size and location constraints of the mirrors in vehicles cause blind zones in drivers' view. This obscured areas surrounding the driver are called blind spots (Figure 3.29).

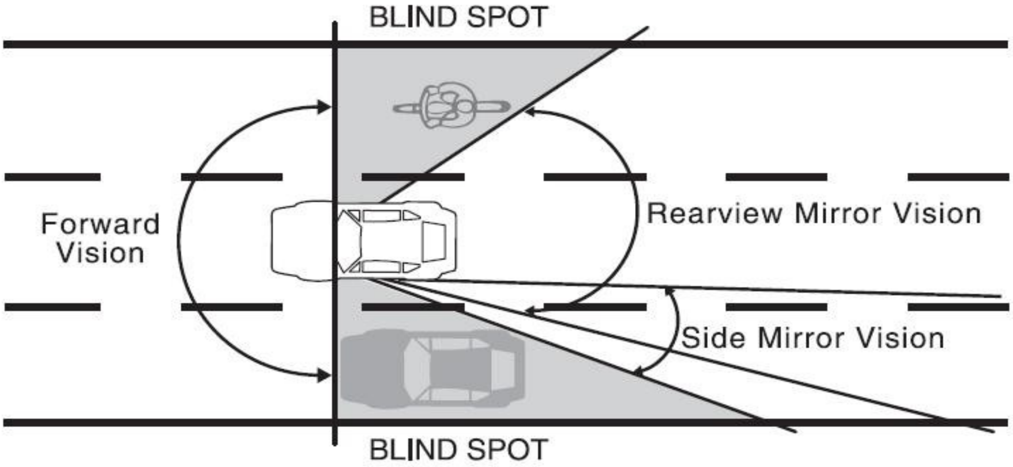


Figure 3.29: Blind Spot Diagram of a Vehicle

Blind Spot Warning Module starts running after installed side view cameras. For each captured image of both sides, RGB to Grayscale Conversion, Sobel Edge Detection and Dilation are applied. Then, the resulting image is divided into four regions of interest (Figure 3.30).

In the last obtained grayscale image, the road surface has low pixel values and the objects on it create edges and have higher pixel values. When a vehicle comes into a region of interest in the image, the average pixel value increases remarkably in that region. If this increase is greater than the threshold value which is specific for the corresponding region then the nearness of the vehicle coming from back is determined from the region number.

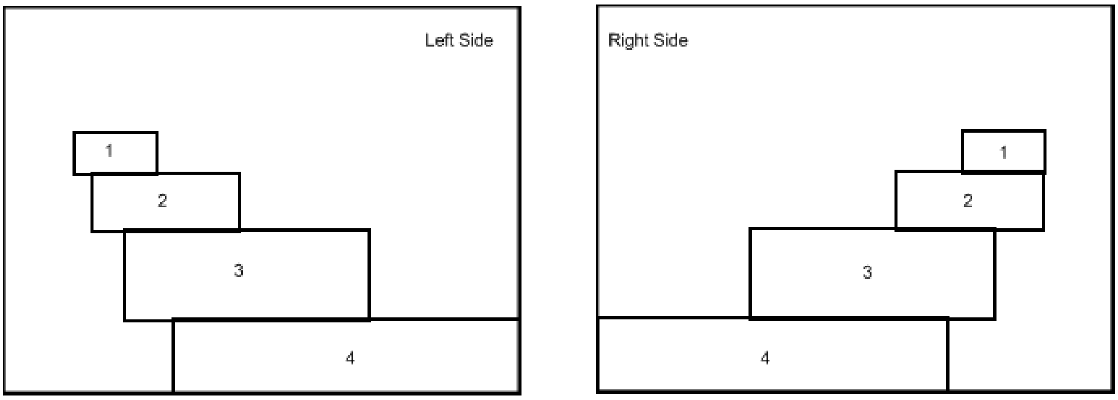


Figure 3.30: Regions of Interests on an Image for Blind Spot Warning

Another image processing technique called Optical Flow is applied to ensure that the detected objects are moving onto the ego vehicle. If the optical flow results moving objects in the same region of interest where the vehicle is detected, the warning is given according to the nearest vehicle (Figure 3.31).

Qualitative values far, approaching, near and very near are assigned to the regions 1, 2, 3, 4 respectively and indicate the location of the vehicle. The regions of interest are not equally sized because of the visual perspective. The closer objects cover more area in the image.

The degree of nearness of vehicle is displayed as a warning. When there is a vehicle in near or very near regions, if a steering is done, a sound alert is given. Steering information is obtained from CAN bus.

Following figures show the process for 5 different cases: empty road, far, approaching, near and very near vehicle. Example images are at size 240x320. Region 1 is a rectangle of size 215x47 which starts at point 105, 193. Region 2 is a rectangle of size 150x55 which starts at point 75, 138. Region 3 is a rectangle of size 90x35 which starts at point 55, 103. Region 4 is a rectangle of size 50x25 which starts at point 44, 78. Threshold values for regions 1, 2, 3, 4 are 70, 20, 15 and 10 respectively (Figure 3.32) (Figure 3.33) (Figure 3.34) (Figure 3.35).

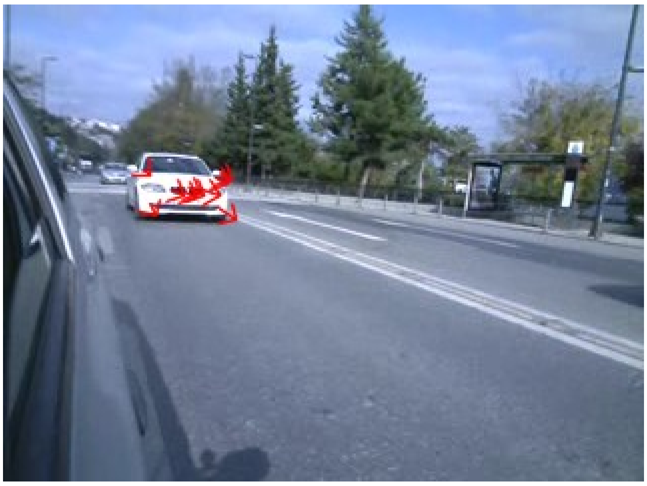


Figure 3.31: Optical Flow Result

Implementation issues are explained in the next chapter.

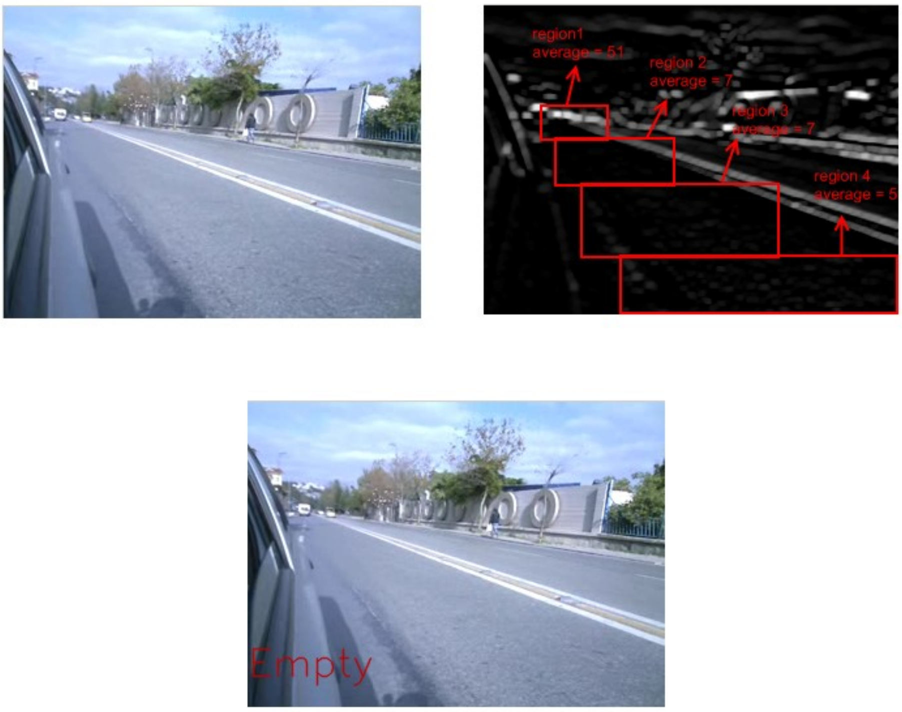


Figure 3.32: Blind Spot Warning on an Empty Road

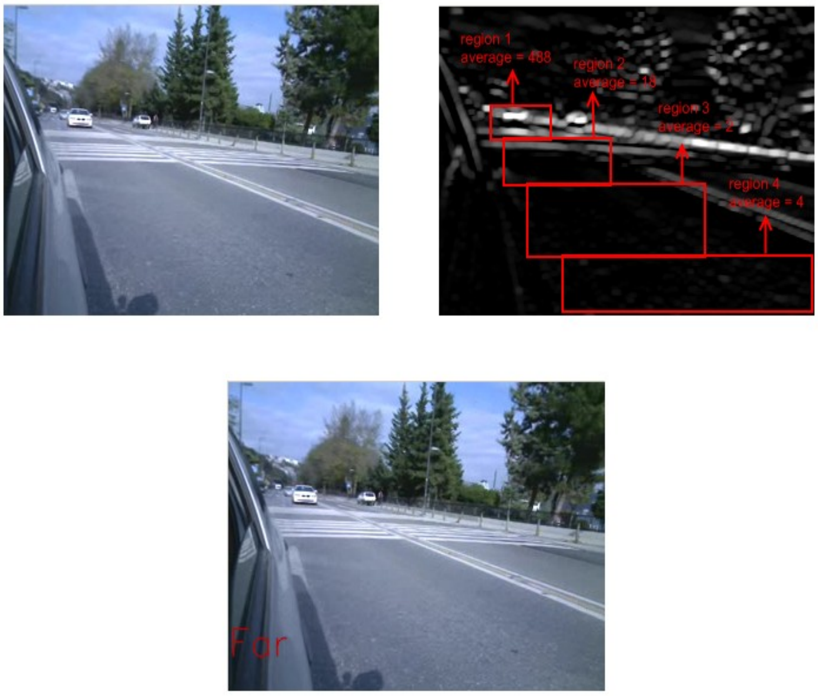


Figure 3.33: Blind Spot Warning with a Far Vehicle

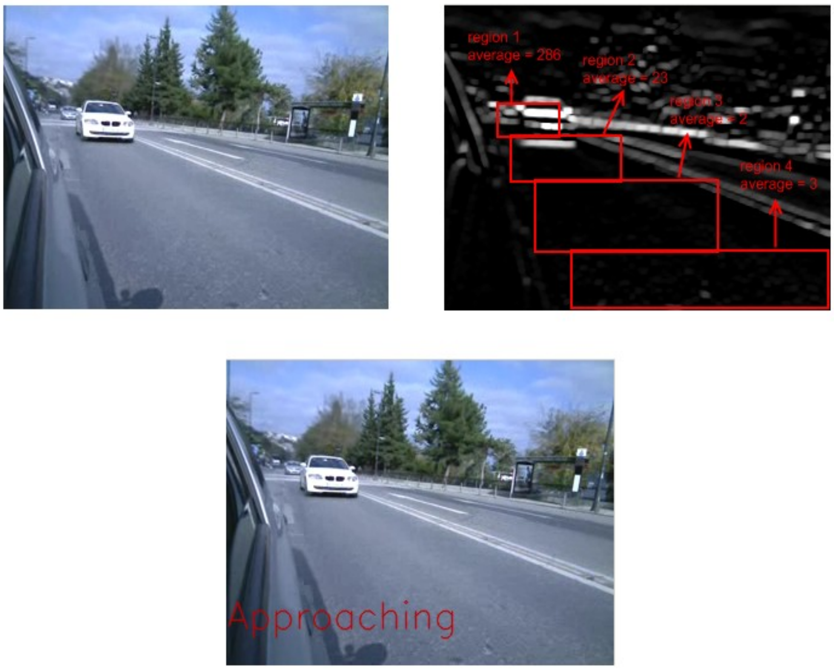


Figure 3.34: Blind Spot Warning with an Approaching Vehicle

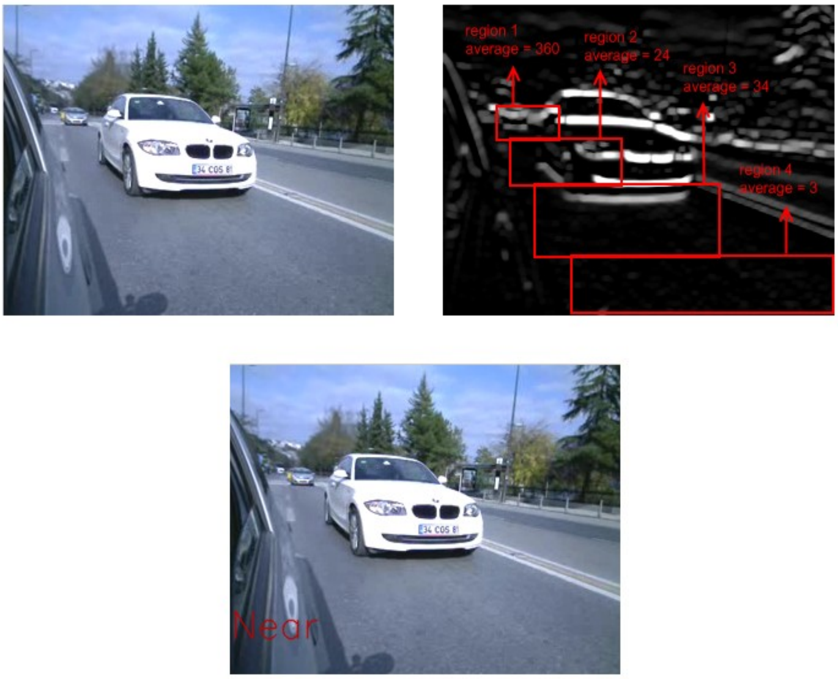


Figure 3.35: Blind Spot Warning with a Near Vehicle

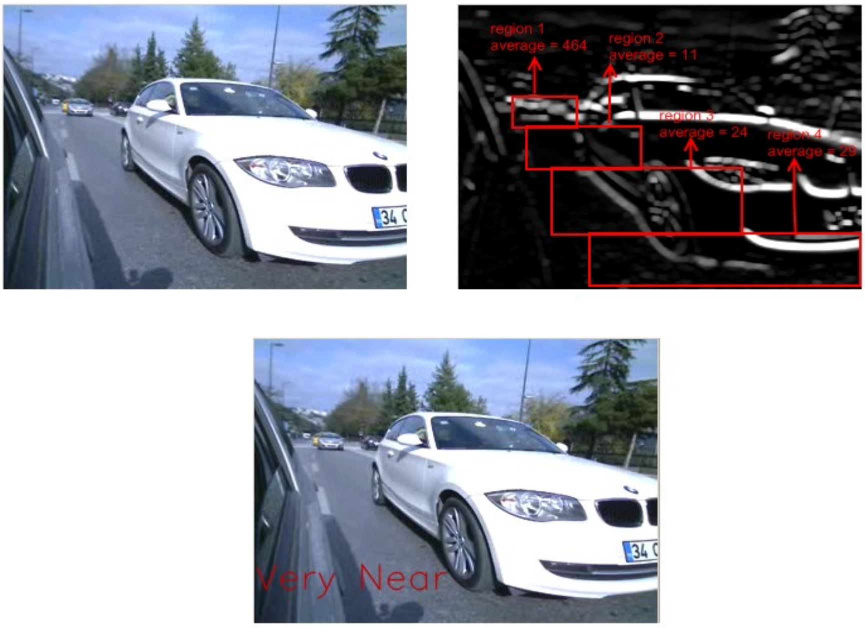


Figure 3.36: Blind Spot Warning with a Very Near Vehicle

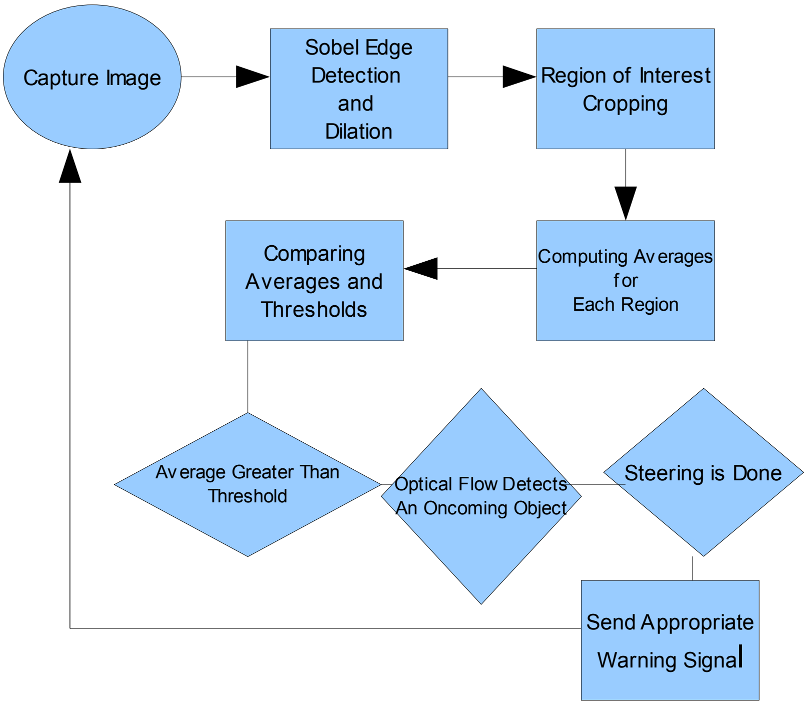

Capture Image
Sobel Edge Detection and Dilation
Region of Interest Cropping
Computing Averages for Each Region
Comparing Averages and Thresholds
Average Greater Than Threshold
Optical Flow Detects An Oncoming Object
Steering is Done
Send Appropriate Warning Signal

Figure 3.37: Blind Spot Warning Module Flowchart

4. IMPLEMENTATION OF THE SYSTEM ON A REAL TIME BASIS

4.1. Configuration Aspects and Mechanical Parts of the System

The proposed low cost driver warning system is constituted by the sensor suite of one camera looking forward to percept vehicles, lane borders and detect possible lane departure in a sensing range of 100 meters, two web cameras placed into the side mirrors detecting the oncoming vehicles in the adjacent lanes in a sensing range of 20 meters and an onboard computer (Figure 4.1).

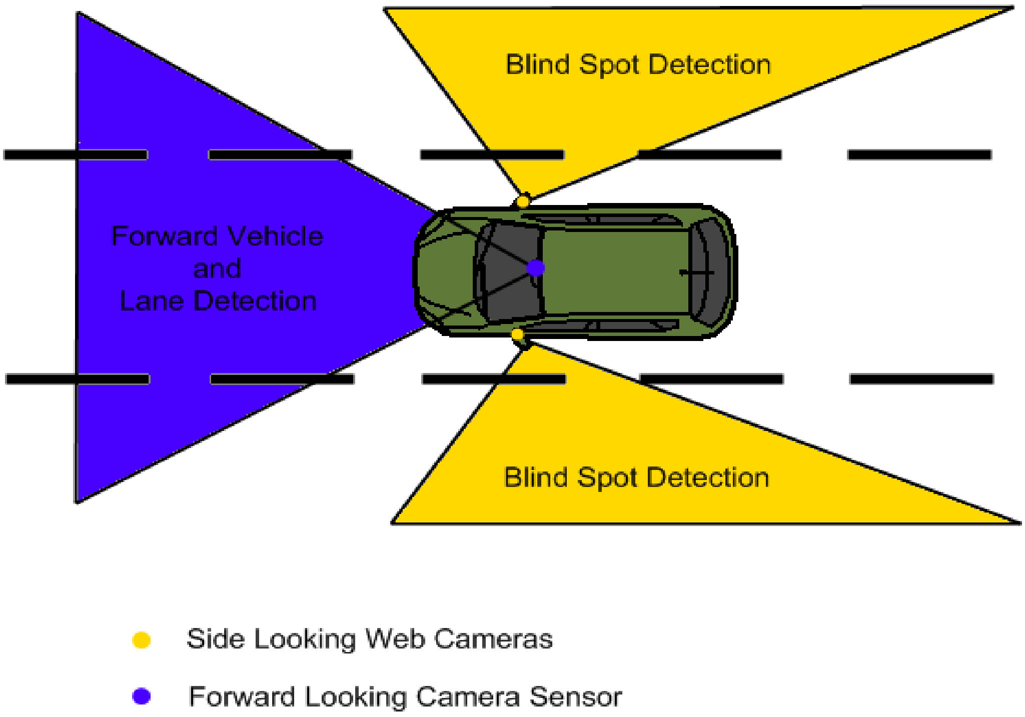


Figure 4.1: Camera Installation on Vehicle

4.2. Blind Spot Warning Module Aspects

As illustrated in Fig. 3.29, the regions of interest create a logic array of two bits denoted by “Region I” when the vehicle is detected in the far and it takes the value of “00”, when the vehicle is approaching, it is in the “Region II” denoted by “01” and when the coming vehicle is in the possible crash zone with respect to the subject vehicle, the driver’s lane change intent by indicating lane change switch or his lane change maneuver attempt by turning the steering wheel activates the audio and visual warning in Region III having the binary value of “10”. The possible lane change may cause an accident in Region III and Region IV “11” informs about the vehicle’s possibility in the collision zone (Figure 4.2). Blind spot binary code may help towards very simple relative speed knowledge and false alarm avoidance when the subject vehicle is overtaking or passing the stationary or moving vehicles in the left side. The sensing range and the warning timing threshold is chosen on a basis of the perception-reaction time of the driver. The sensing range for the web cameras and the short range radar is 20 meters. When the relative distance between the subject and the oncoming vehicle is 20 meters and when the the relative speed between the vehicles are less than 10 meters/sec, the approaching vehicle may be safely monitored in the time interval of 2 seconds.

A possible lane change attempt scenario may be such that a subject vehicle cruising at 72 km/h (20 m/s) intents to change into the right lane where another vehicle on that lane is approaching with 108 km/h (30 m/s). The relative speed is 10 m/s letting two seconds monitoring time windows to warn the driver in-the-loop letting him to percept the approaching vehicle in the first half of the monitoring time windows.

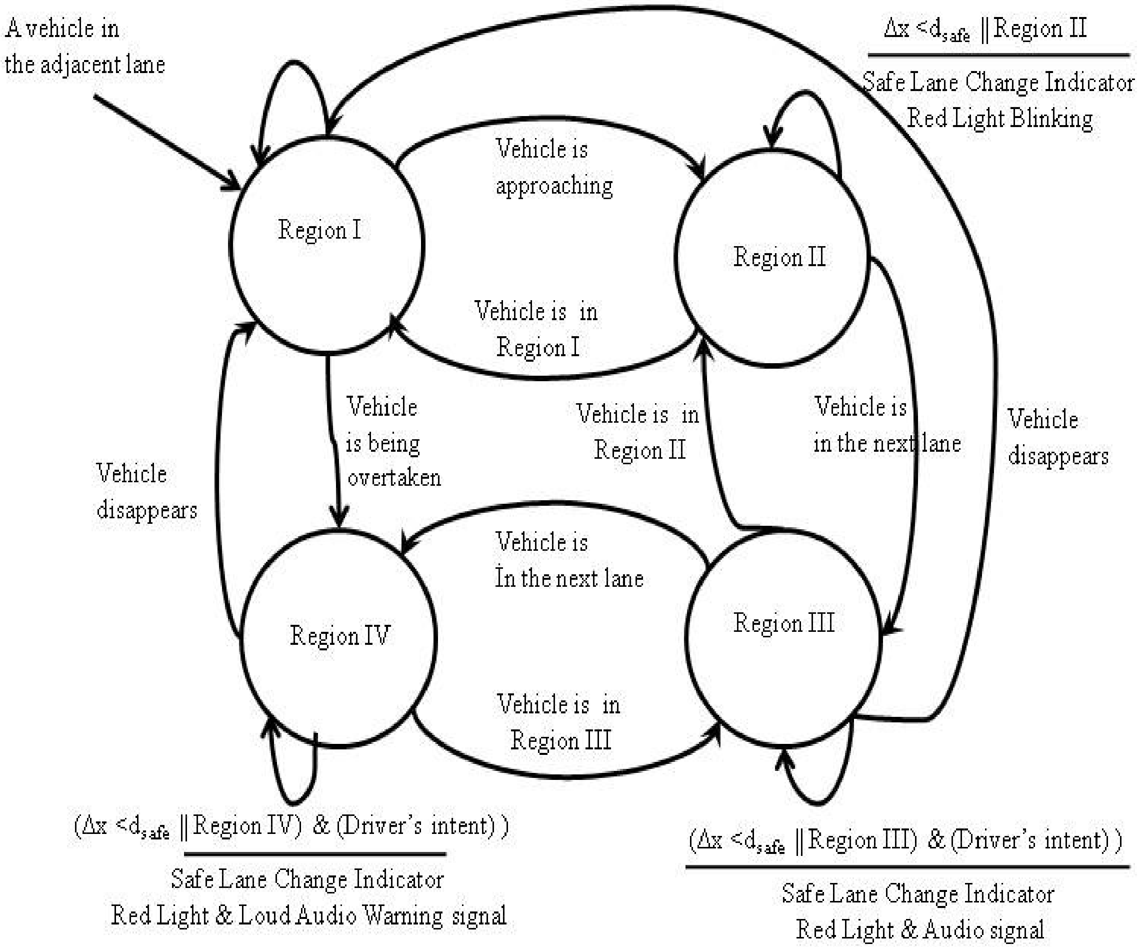


Figure 4.2: Finite State Machine Modeling of Blind Spot Detection Algorithm

4.3. Lane Departure Warning Module Aspects

The presented warning system monitors lane keeping performed by the driver and checks for potential lane departure situations. Thus it requires knowledge of time to cross the lane border. Time to lane crossing is the time required for a vehicle to run off the boundary assuming constant steering. Time to lane crossing is derived by the following equations:

Define the yaw angle:

$$\theta(t) = \int_0^t r(\tau)d\tau \tag{4.1}$$

where ***r*** denotes the yaw angle read from the CAN of the vehicle. The lateral displacement is formulated [27]:

$$y(t) \cong \int_0^t V(t)\cos(\theta(t))dt + \int_0^t U(t)\sin(\theta(t))dt \tag{4.2}$$

where ***V(t)*** is the lateral velocity and ***U(t)*** longitudinal velocity and it is assumed to be constant for lane departure scenario. If is sufficiently small for all ***t***, then the following approximations are valid: $cos\theta(t) \cong 1$, $sin\theta(t) \cong \theta(t)$. Consequently,

$$y(t) \cong \int_0^t V(t)dt + \int_0^t U(t)dt \tag{4.3}$$

The predicted lateral position of the vehicle at time $t + \Delta t$ can be obtained by:

$$\hat{y}(t + \Delta t) = y(t) + V(t)\Delta t + U(t)\theta(t)\Delta t \tag{4.4}$$

then time to lane crossing is defined as the time before the vehicle's lane position exceeds the lane boundary,

$$\hat{\tau} = \min_{\Delta t}\left\{ \Delta t : \left| \hat{y}(t + \Delta t) \right| > \frac{d_{road} - d_{vehc}}{2} \right\} \tag{4.5}$$

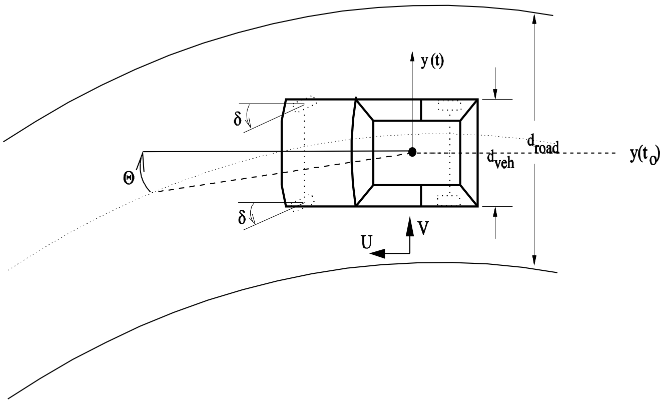


Figure 4.3: Lane Departure Position

where d_{road} is the lane width assumed to be constant and d_{veh} is the width of the vehicle.

4.4. CAN Bus Support

Controller Area Network (CAN) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other within a vehicle without a host computer [28]. Each subsystem in vehicle is referred Electronic Control Unit (ECU). Engine, transmission, brakes, airbags, etc. have all their own ECUs. The communication between these ECUs are provided by the CAN bus. It is possible to get the information from each ECU thanks to the CAN analyzer cards plugged into a computer. Another connection from CAN analyzer card is needed to get the information from the CAN bus output. Data from the central CAN bus are the series of 8 bytes. The meaning of the byte segments is subject to change according to the manufacturer of the vehicle. To reach the 8 bytes segment of an information, one will need the CAN IDs which are different hexadecimal values for different types of information from each ECUs (e.g. 0x302 for brakes, 0x2C3 for vehicle speed, etc.) Information is usually stored as bit segments in one of the 8 bytes. In proposed system, yaw rate, vehicle speed, left and right turning indicators and steering angle informations are obtained from the CAN bus.

4.5. Warning Devices

Warning equipments play an essential role in the implementation phase. In order to prevent driver to make inattentive lane departures and to make him awake, a warning equipment should generate both audio and visual signals. An equipment consisting of led lights and beepers was designed for experimental purposes. Equipment is connected to parallel port of the onboard computer. Equipment has two boards for right and left departure warnings, two boards for right and left blind spot warnings and one board for forward collision warning. Each blind spot warning equipment consists of a beeper and four series of led lights one for each region and last two series are red. In each region, a new led series are on. Eventually, all the leds are on when there is a very near vehicle. Each lane departure warning equipment has two series of led lights and a beeper. The forward collision warning equipment has a beeper and two series of red led lights. 1 byte of data are used to send appropriate signals to equipments. First bit of data is for FCW, second bit is for right lane departure, third bit is for left lane departure, forth bit is for the first region of blind spot, fifth bit is for the second region and sixth and seventh bits are for third and forth regions of blind spot. Last bit is for the selection of the side for blind spot detection (Figure 4.4)

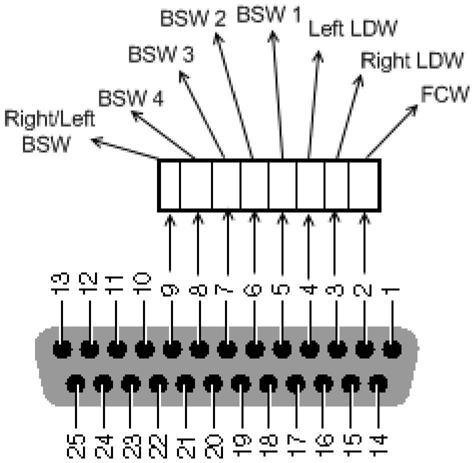


Figure 4.4: Bit Locations on Parallel Port

For example a bit sequence “01000010” means that a very near vehicle is detected on the left side and driver moves out of its lane to the right. When a left steering is done in this case, beep signals will be continuous. A bit sequence “100100001” means that a vehicle is approaching from the right side and also there is a risk of forward collision.

Each board consists of a BC337 transistor, led light and resistors of 1K ohms (Figure 4.5) (Figure 4.6).

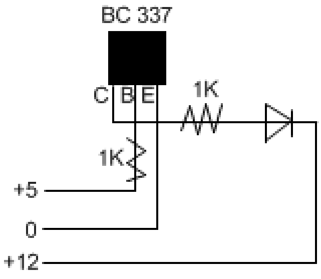
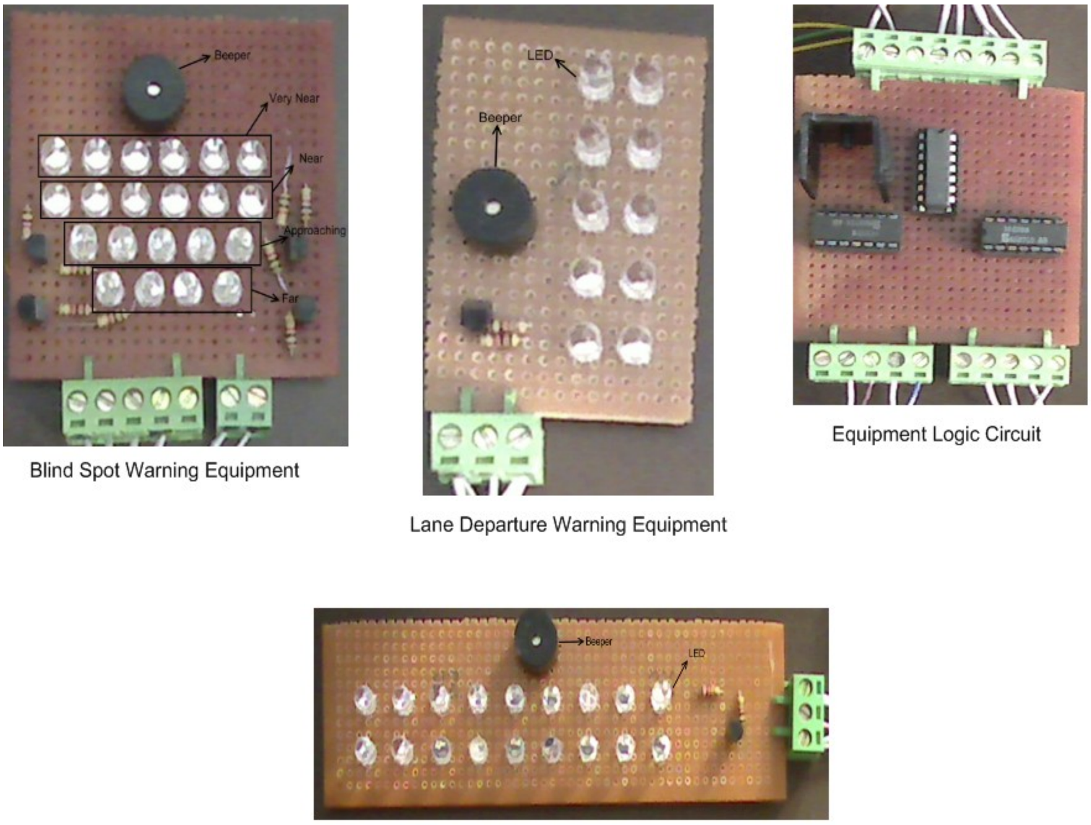


Figure 4.5: Circuit of a Board of Equipment



Blind Spot Warning Equipment

Lane Departure Warning Equipment

Equipment Logic Circuit

Forward Collision Warning Equipment

Figure 4.6: Physical Equipments

4.6. Experiments and Results

The driver warning system is tested on the test-bed vehicle provided by the project called “Drive Safe” in Turkey. The test-bed vehicle was developed and the project aims to sense the human driver’s intents, physical and mental condition, driving environment, traffic flow and the vehicle motion through multimodal sensor systems and to develop effective methods to identify the driver’s conditions (Figure 4.7). The camera sensors are installed (Figure 4.8) and the steering wheel angle, vehicle velocity, yaw rate, the left, right switch indicators are read from CAN bus of the vehicle signal system.

To store and process the data on a real-time basis, workspace is setup in left back seat space (Figure 4.9).

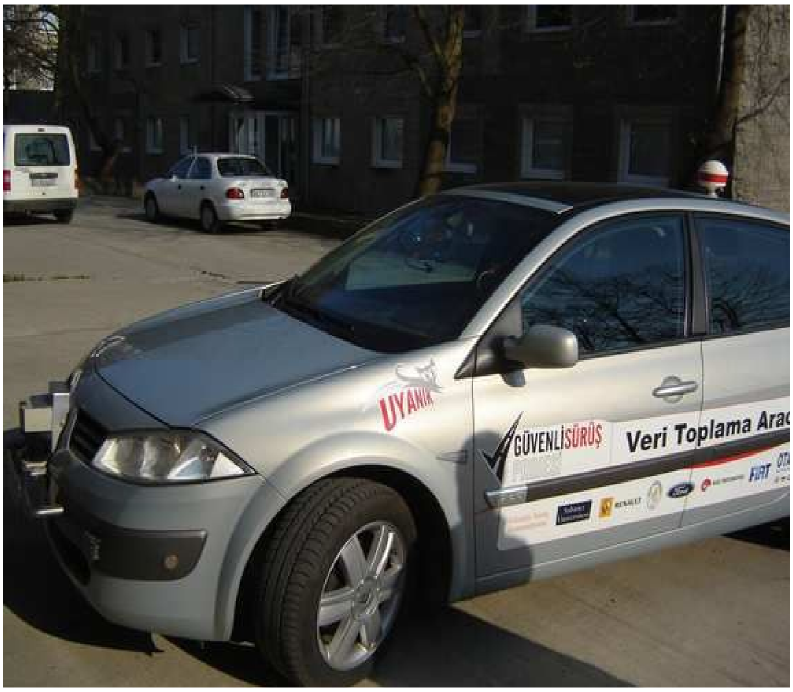


Figure 4.7: Data Acquisition Test Vehicle

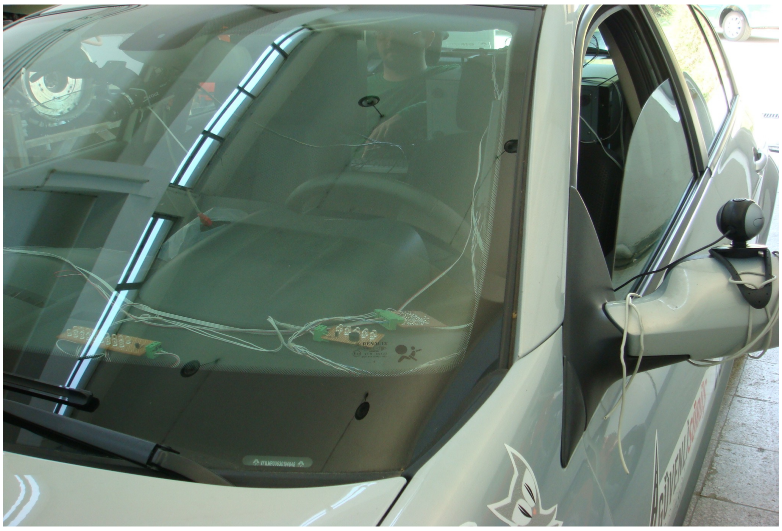


Figure 4.8: Camera Installation on Test Vehicle

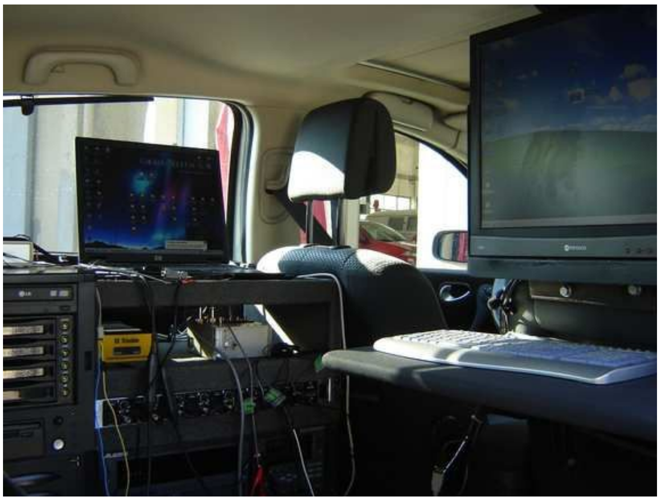


Figure 4.9: Data Storage and Processing in the Left Back Seat

Side view cameras for blind spot warning module are two Logitech QuickCam S5500 webcams with UltraBrite technology. The optical sensor used for forward image data acquisition is Basler A601FC color camera with 640 x 480 pixel resolution and the focal length of 16 mm. Frame rates of 10 and 5 are used for side views and forward view respectively. The onboard computer is a Quad Core with 3GB of RAM and a CAN analyzer card.

GPS support was also added during the tests. So that, the tracks of the faults are kept with global position information. These data are used by another project to show the faults of the driver on a map.

The warning system has been tested in freeway traffic. Left and right lane markers are detected successfully and the center of gravity of the subject vehicle with respect to the lane borders are computed. In the moving traffic, the relative distance between the vehicle in front is computed by searching the predefined template image representing the shadow. Results from the test drives are shown (Figure 4.10) (Figure 4.11) (Figure 4.12) (Figure 4.13).

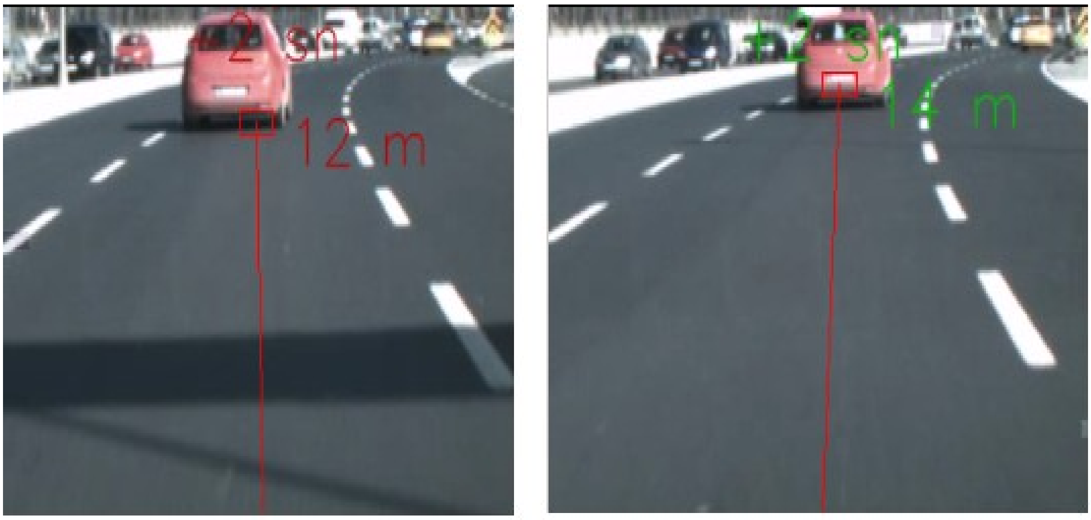


Figure 4.10: Forward Collision Warning Result

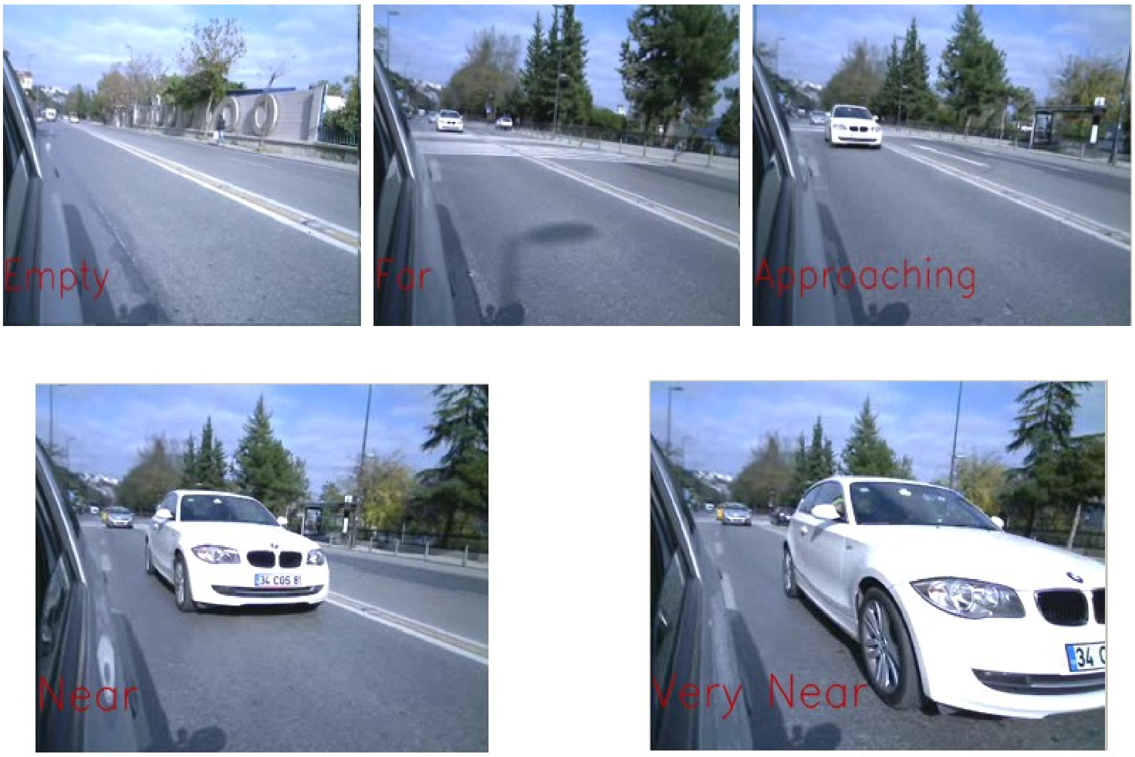
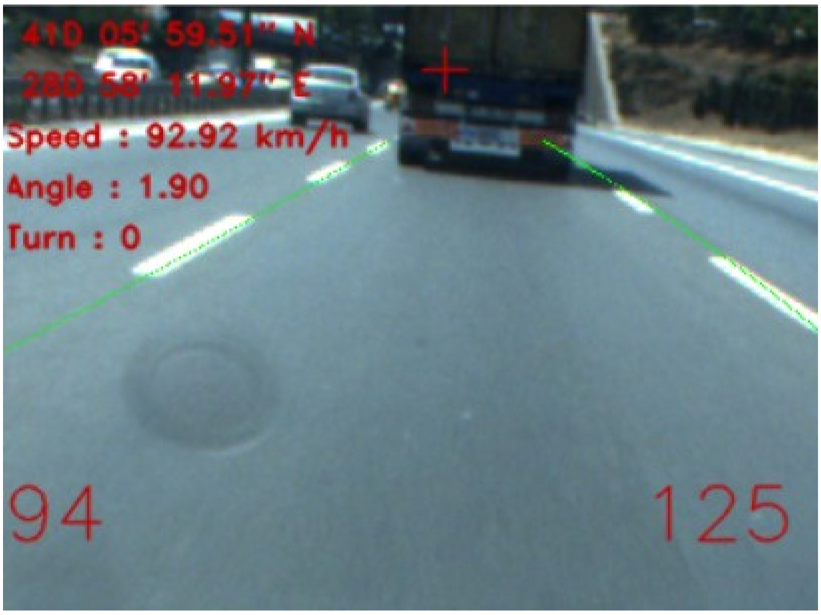


Figure 4.11: Blind Spot Warning Result



41D 05' 59.51" N
28D 58' 11.97" E
Speed : 92.92 km/h
Angle : 1.90
Turn : 0

Figure 4.12: Screenshot of LDW with GPS and CAN Information

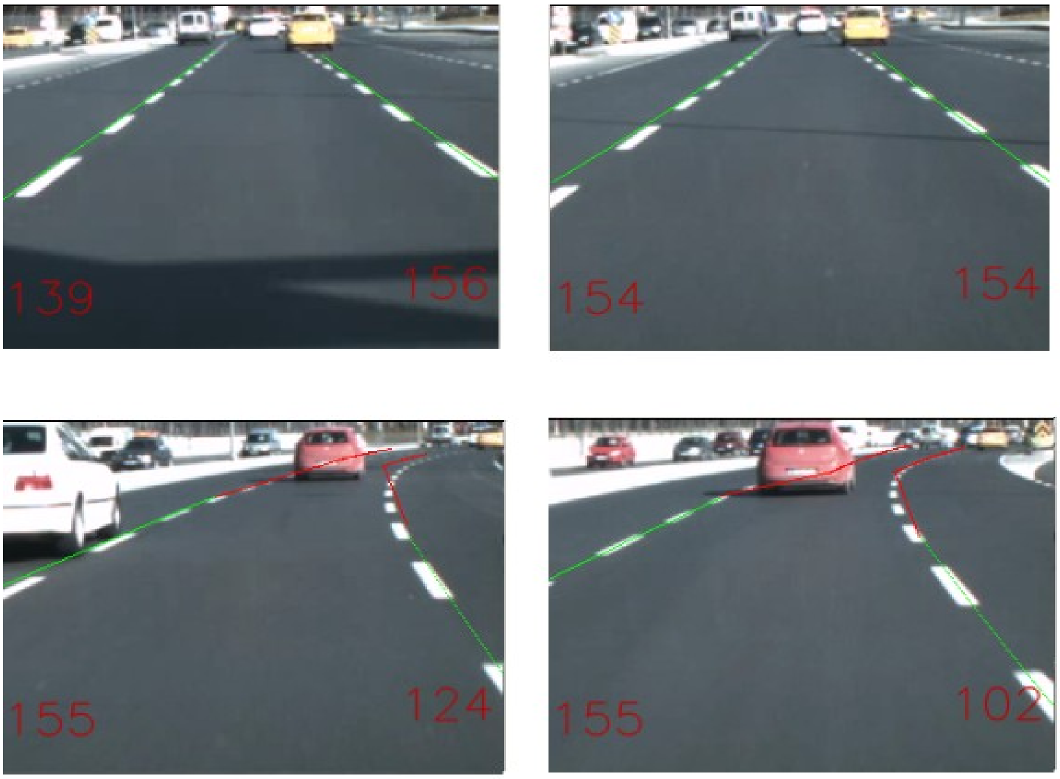


Figure 4.13: Lane Detection Results

Table 4.1: Performance of the System

APPLICATION NAME	**COMPUTER SPECS**	**PROCESSING TIME**
Blind Spot Warning	Core 2 Quad 2.83 GHz, 3 GB RAM	63 milliseconds/frame
Forward Collision Warning	Core 2 Quad 2.83 GHz, 3 GB RAM	86 milliseconds/frame
Lane Departure Warning	Core 2 Quad 2.83 GHz, 3 GB RAM	115 milliseconds/frame

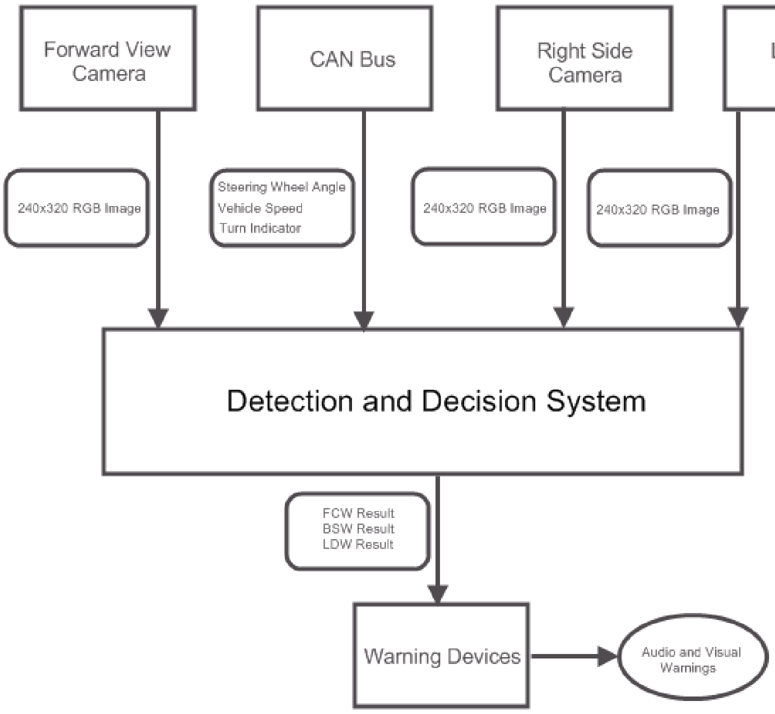


Figure 4.14: General Overview of the System

5. CONCLUSION

This thesis has presented the development and a real time implementation of a new driver warning system. Although there are various driver assistance systems in the market, these systems do not provide a complete assistance, because LDW, FCW and BSW systems are not sold as a single package. Also the cost of these systems are not affordable for most of the customers. Some of them use radars, lidars, GPS and GPRS and this makes the costs higher for both customers and manufacturers. The new platform has a low cost due to its equipments. It only uses three cameras and an onboard computers. If the high prices of the driver warning systems in the market are taken into account, this new system costs at least half of the others. A new blind spot warning algorithm and a new approach for lane detection have also been presented.

Systems in the market are equipped with radars, cameras or infrared ports. Generally, vehicle detection algorithms are used to detect the vehicles in the blind spot. But the new system detects the rapprochement of a vehicle into the blind zone and also divides the rear view into regions of interest to obtain location information for the vehicle coming from behind. LDW and FCW modules uses the same frames from the same camera and this reduces the running time since the image capturing is one of the most time spending part.

This new platform is an infrastructure for future researches. More properties can be added like robuster curve detection, more accurate image processing algorithms for edge and blob detection, driver evaluation module and Kalman filter based accuracy for the lane border detection.

REFERENCES

[1] Turkish General Directorate of Security, "*Accident Statistics of Last 10 Years*", http://www.trafik.gov.tr, (2008).

[2] Hendricks, D.L., Fell, J.C., Freedman, M., "*The Relative Frequency of Unsafe Driving Acts in Serious Traffic Crashes*", U.S. National Highway Traffic Safety Administration, (2001).

[3] VirginiaTech Transportation Institute, "*100-Car Naturalistic Driving Study*", http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/100CarPhase1Report.pdf, (2002).

[4] U.S. Federal Highway Administration, "*Road Safety Fact Sheet*", http://safety.fhwa.dot.gov/facts/road_factsheet.htm, (2008).

[5] Insurance Institute for Highway Safety, "*Status Report*", 43(3), (2008).

[6] Gat, I., Benady, M., Shashua, A., "*A monocular vision advance warning system for the automotive aftermarket*", SAE transactions, vol.114, 7, 403-410, (2005).

[7] Redmill, K.A., "*A simple vision system for lane keeping*", Proceedings of Intelligent Transportation Systems, Boston, MA, (1997).

[8] Bertozzi, M., Broggi, A., "*GOLD: a paralel real-time stereo vision system for generic obstacle and lane detection*", IEEE Transactions on Image Processing, 7(1), 62-81, (1998).

[9] Wang, Y., Teoh, E.K., Shen, D., "*Lane detection and tracking using B-Snake*", Image and Vision Computing, 22, 269-280, (2004).

[10] Lopez, A., Canero, C., Serrat, J., Saludes, J., Lumbreras, F., Graf, T., "*Detection of lane markings based on ridgeness and RANSAC*", Proceedings of Intelligent Transportation Systems, 254-259, (2005).

[11] Sotelo, M.A., Barriga, J., Fernández, D., Parra, I., Naranjo, J.E., Marrón, M., Alvarez, S., Gavilan, M., "*Vision-based blind spot detection using optical flow*", Computer Aided Systems Theory – EUROCAST, 4739, 1113-1118, (2007).

[12] Blanc, N., Steux, B., Hinz, T., "*LaRASideCam: a fast and robust vision based blind spot detection system*", Proceedings of the 2007 IEEE Intelligent Vehicles Symposium Istanbul, Turkey, (2007).

[13] Krips, M., Velten, J., Kummert, A., Teuner, A., "*AdTM tracking for blind spot collision avoidance*", IVS, 544-548, (2004).

[14] Kucukay, F., Bergholz, J., "*Driver Assistant Systems*", http://www.osd.org.tr/9.pdf, (2004).

[15] United States Department of Transportation, "*Description of the IVI Technologies and the FOT*", http://www.itsdocs.fhwa.dot.gov/JPODOCS/REPTS_TE/14352_files/2.0description.htm, (2003).

[16] Green, M., "*How Long Does It Take To Stop? Methodological Analysis of Driver Perception-Brake Times*", Transportation Human Factors, 2, 195 - 216, (2000).

[17] Hagon, T., "*ESP standard on new Commodore*", http://www.drive.com.au/Editorial/ArticleDetail.aspx?ArticleID=17765, (2006).

[18] Acarman, T., Pan, Y., Özgüner, Ü., "*A control authority transition system for collision and accident avoidance*", The Vehicle System Dynamics Journal, 39(2), 149-187, (2003).

[19] Renski, A., "*The Driver Model and Identification of its Parameters*", SAE Paper 980011, Warrendale, PA, (1998).

[20] Wikipedia, "*Sobel Operator*", http://en.wikipedia.org/wiki/Sobel_operator, (2008).

[21] Wikipedia, "*Dilation(Morphology)*", http://en.wikipedia.org/wiki/Dilation_(morphology), (2008).

[22] Wikipedia, "*Gaussian Blur*", http://en.wikipedia.org/wiki/Gaussian_blur, (2008).

[23] Duda, R.O., Hart, P.E., "*Use of the Hough Transformation to Detect Lines and Curves in Pictures*", Comm. ACM, 15, 11–15, (1972).

[24] Chang, F., Chen, C., Lu, C., "*A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique*", Computer Vision and Image Understanding, 93(2), 206-220, (2004).

[25] Wikipedia, "*Optical Flow*", http://en.wikipedia.org/wiki/Optical_flow, (2009).

[26] Wikipedia, "*Lucas–Kanade method*", http://en.wikipedia.org/wiki/Lucas-Kanade_Optical_Flow_Method, (2009).

[27] Hatipoğlu, C., Özgüner, Ü., Künyelioğlu, O., "*Advanced automatic lateral control schemes for vehicles on highways*", IFAC World Congress, 477–482, San Francisco, USA, (1996).

[28] Wikipedia, "*Controller-area Network*", http://en.wikipedia.org/wiki/Controller -area_network, (2009).

APPENDIX A: OPENCV and MATLAB

All the codes were written in Microsoft Visual Studio 2008 powered by Intel OpenCV library and MATLAB. A good reference about installing OpenCV with Visual Studio 2008 can be found at Laganière's site:
http://www.site.uottawa.ca/~laganier/tutorial/opencv+directshow/cvision.htm

OpenCV comes with a big documentation file and some sample codes. More information can be found at Intel's website: http://www.intel.com.

There is also a blob detection library written for OpenCV which can be found at:
http://opencv.willowgarage.com/wiki/cvBlobsLib

Almost all functions in OpenCV have equivalent functions in MATLAB Image Processing Toolbox. Detailed information about MATLAB can be found at:
http://www.mathworks.com.

APPENDIX B: SIDE VIEW CAMERA SPECIFICATIONS

Technical Specifications of Logitech QuickCAM S5500 WebCAM

- True 1.3-megapixel sensor with RightLight™ technology
- Live capture: Up to 1280 x 960 pixels
- Still image capture: 4-megapixels* (software enhanced)
- Built-in microphone with RightSound™ technology
- Privacy shade for audio and video mute
- Live video up to 30 frames per second (with recommended system)
- Hi-Speed USB 2.0
- (6-foot, 1.82-meter) USB cable
- Adjustable clip/base fits any monitor or notebook
- Fixed focus

APPENDIX C: FORWARD VIEW CAMERA SPECIFICATIONS

Technical Specifications of Basler A601FC Camera

Table C.1: Technical Specifications of Basler A601FC Camera

Sensor Size (H x V Pixels)	656 x 491 (mono), 656 x 490 (color)
Sensor Type	Progressive Scan CMOS
Pixel Size	9.9 µm x 9.9 µm
Max. Frame Rate at Full Resolution	60 frames/s (fixed)
Video Output Format	Mono: 8 bits/pixel, Color: YUV4:2:2 at half frame rate or raw data
Color / Mono	Color or Mono
Video Output Type	IEEE 1394
Synchronization	Via external trigger or the 1394 bus
Exposure Control	Programmable via the 1394 bus
Power Requirements	+8 to +36 VDC (12 VDC nominal) max. ~ 2 W
Lens Mount	C-mount
Housing Size (L x W x H)	67.3 mm x 44 mm x 29 mm
Weight	max. 100 g (typical)
Conformity	CE, FCC

BIOGRAPHICAL SKETCH

Burak Çayır was born in Çorlu in 1986. He completed his primary education in Istanbul and Edremit. He started to high school in Edremit Özcan College with scholarship in 1999 and was graduated in 2003. He took his B.Sc. degree from Izmir University of Economics Computer Engineering Department with scholarship and as a high honour student in 2007. He is currently a Master of Science student in Computer Engineering in Galatasaray University.