

Estimation of Vehicle Lateral Position Using the Optical Flow Method

A Thesis  
SUBMITTED TO THE FACULTY OF  
UNIVERSITY OF MINNESOTA  
BY

Rini Shrestha

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
MASTER OF SCIENCE

Dr. Jiann-Shiou Yang

May 2013

## **Acknowledgements**

First, I would like to thank my advisor, Dr. Jiann-Shiou Yang, for providing me an opportunity to work under his guidance on his research study. Furthermore, I would like to express my gratitude to the members of my thesis examination committee, Dr. Imran Hayee and Dr. Ryan G. Rosandich, for their valuable time and assistance. I am grateful to Benjamin Nelson for helping me during the experiments. I would also like to thank the entire electrical engineering faculty and my colleagues for their support during my graduate study.

## **Abstract**

With the increasing number of vehicles on the road the number of accidents have also been increasing. Development of various techniques such as lane departure warning systems that helps drivers to assist in driving can help reduce the number of accidents significantly. In this thesis, we attempt to develop such lane departure warning system by estimating the vehicle's lateral position. The lateral position of vehicle can be known if the heading angle of the vehicle can be determined. Therefore, this study focuses on determining heading angle and works toward development of the lane departure warning system based on image processing techniques. An in-vehicle camera is used to capture the images of the road in real time. The system then uses homography on this front - view images of the road to remove the perspective effect and transform the images such that the obtained resulting images represent as if they were observed from the top. The histogram equalization is also applied to the images to increase the global contrast by spreading out the most frequent intensity values. Shi and Tomasi corner detection technique has been used to find significant features (corners) in the images and Lucas - Kanade optical flow to track those corners in following images. The heading angle, and thus the lateral displacement, is determined by relating these tracked corners. As part of the study, a number of road tests were conducted on different roads of Duluth, MN and the findings based on the road tests are discussed.

# Table of Contents

<b>LIST OF TABLES .....</b>	<b>V</b>
<b>LIST OF FIGURES .....</b>	<b>VI</b>
<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1 OVERVIEW .....	1
1.2 BACKGROUND.....	2
1.3 MOTIVATION.....	6
1.4 SCOPE .....	10
<b>2. OVERVIEW OF THE LANE DEPARTURE WARNING SYSTEM .....</b>	<b>11</b>
2.1 LDW SYSTEM.....	11
2.2 LANE MARKER DETECTION .....	14
2.2.1 Region of Interest (ROI) .....	14
2.2.2 Gaussian Filtering .....	16
2.2.3 Hough Transform for Lane Detection.....	17
2.2.4 Calculate distance to lane boundary .....	20
2.3 POINT TRACKING .....	20
2.3.1 ROI.....	20
2.3.2 Feature Selection.....	21
2.3.3 Feature Tracking .....	22
2.3.4 Heading Angle and Lateral Displacement Computation .....	22
<b>3. DESCRIPTION OF THE POINT TRACKING APPROACH .....</b>	<b>23</b>

3.1	FEATURE SELECTION .....	23
3.2	THE LUCAS-KANADE OPTICAL FLOW BASED TRACKING .....	26
3.3	VEHICLE'S LATERAL CHARACTERISTICS .....	32
<b>4.</b>	<b>EXPERIMENTS AND RESULTS .....</b>	<b>35</b>
4.1	CAMERA .....	35
4.2	HISTOGRAM EQUALIZATION .....	38
4.3	FRONT TO TOP-VIEW CONVERSION .....	38
4.4	REGION OF INTEREST .....	39
4.5	ROAD TEST .....	40
4.6	RESULTS .....	46
<b>5.</b>	<b>CONCLUSION .....</b>	<b>51</b>
5.1	SUMMARY .....	51
5.1	ISSUES AND RECOMMENDATIONS .....	52
<b>6.</b>	<b>REFERENCES.....</b>	<b>53</b>
<b>7.</b>	<b>APPENDIX.....</b>	<b>58</b>
7.1	RESULT OF THE TEST SHOWING THE POINTS TRACKED AND ANGLE BETWEEN THEM FOR IMAGE FRAMES 1 TO 10 .....	58

## List of Tables

Table 1.1: Car manufacturer and their LDW system implementation year .....	4
Table 1.2: Effectiveness of LDW systems .....	8
Table 4.1: Result of the test showing the points tracked and angle between them.....	42
Table 4.2: Average angle calculated from features .....	44
Table 4.3: Values of heading angle and lateral displacement.....	46

## List of Figures

Figure 1.1: LDW systems (a) camera-based (b) infrared sensor based [3] [4].....	3
Figure 1.2: Modes of transportation for commute in the U.S. [14] .....	7
Figure 2.1: Lane departure warning system flowchart .....	13
Figure 2.2: Lane marker detection block diagram .....	14
Figure 2.3: : Top - view image and ROI (a) A full top - view image (b) ROI of the top - view image .....	15
Figure 2.4: ROI filtering (a) before filtering (b) after filtering.....	16
Figure 2.5: (a) Three data points with $r$ and $\theta$ for different lines passing through them (b) Hough parameter graph plot showing the intersection of curves representing the line in which all three data points lie on [23].....	19
Figure 2.6: Point tracking block diagram .....	21
Figure 3.1 Representation of optical flow equation [20] .....	29
Figure 3.2: Heading angle estimation .....	32
Figure 4.1: Microsoft LifeCam Cinema 720p HD webcam used for the experiments .....	35
Figure 4.2: Camera installed in the vehicle with the help of a suction cup .....	36
Figure 4.3: (a) Scene from the wide angle camera lens with distorted fish eye effect (b) Scene after removal of the fish eye effect through calibration .....	37
Figure 4.4: (a) View of the from camera (b) View after histogram equalization .....	38
Figure 4.5: (a) Front-view image (b) Conversion of front-view to top-view using homography .....	39
Figure 4.6: Region of interest .....	40
Figure 4.7: Feature displacement.....	40

Figure 4.8: Front view images from frames 1 to 10 .....	41
Figure 4.9: Front - view images from frames 81 to 90 .....	42
Figure 4.10: Heading angle and lateral displacement with 50 mph speed.....	47
Figure 4.11: Heading angle and lateral displacement with threshold > 15.....	48
Figure 4.12: Heading angle and lateral displacement at 65mph with threshold > 5.....	49
Figure 4.13: Heading angle and lateral displacement at 65mph with threshold > 15.....	50



# 1. Introduction

## 1.1 Overview

In today's world of technology, modern vehicles come with many luxurious and safety features and options. The high-tech features that were once optional and used by few have now become a necessary luxury for most. These features are not only popular for their convenience but also for their contribution to safety of vehicle and its riders. From automatic parking systems to in-dash navigation systems, there is a wide range of advanced systems that are integrated in vehicles which can be of great help while driving. Driving on roads can be challenging sometimes. The population of vehicles on roads has been ever- increasing and so is the probability of accidents. Hence, caution should be taken while one is behind the wheels. One of these advanced but necessary systems that could assist drivers while driving and possibly reduce the rate of accidents on roads is Road/Lane Departure Warning Systems.

Lane Departure Warning (LDW) systems are one of the most desired attributes in the vehicle safety system. This is due to the fact that they play a crucial role in preventing drivers from unintentional lane crossing and assisting them to avoid potential road mishaps. Although rumble strips are currently present along the edge of the roads to alert drivers when deviation occurs, chances are that they might not always be available. For example, some state regulation requires a gap of 12 feet for every 40 to 60 feet of rumble-strip [1]. Therefore, this thesis is to develop a solution in form of a LDW system that

would assist drivers in cases where either rumble strips are not available or there exists a wide gap between each rumble strip.

The objective of this thesis is to present the research and development work on the estimation of vehicle lateral characteristic that is a basis for the development of a vision based LDW system that acts as a virtual rumble strip and warns drivers when the vehicle starts to move out of its lane unintentionally. The objective of this research can be best defined by the following elevator test statement.

"For vehicle drivers who want to prevent any driving related mishaps and accidents from occurring, the Onboard Virtual Rumble Strip System is an automated road departure warning system that warns drivers if the vehicle begins to move out of its lane unintentionally. An audio warning is given when the car comes very close to the edge of the lane marker and no turn signal has yet been activated. This can significantly reduce accidents on the road caused by the lack of attention by the driver. Unlike other products which are based just on tracking the lane markers, our system works well in cases where lane markers may not be available.

## **1.2 Background**

LDW system monitors the position of vehicle relative to its lane boundaries. So in order to operate, it requires the lateral distance information of the vehicle from the lane boundaries to make correct decision and alert drivers when they are about to drift away from their lane.

Basically, two types of LDW systems can be found in market these days. These systems have been categorized on basis of the type of sensor they employ. The most common one used by many car manufacturers is the vision based system in which the sensor is a camera that is mounted on the windshield (usually) of the car. The other type uses infrared sensors consisting of an infrared light emitting diode (LED) and a photo cell. Mounted under the front bumper of the car, this second kind of sensor triggers the warning system based on the variations in reflections from the beams by the LED on the road. Although the above mentioned systems are of commercial types, other LDW systems involving different sensors have also been reported. They include technologies like laser scanners, high-resolution radar, combination of Global Positioning System (GPS) receivers and video cameras, to name a few [2].



(a)

(b)

**Figure 1.1: LDW systems (a) camera-based (b) infrared sensor based [3] [4]**

The first ever produced LDW system was developed by a US company, Iteris in 2000 and was implemented on commercial trucks [5,6]. In the following year, Nissan employed the warning system in its consumer vehicle. Table 1.1 lists the car manufacturers and the year they implemented this safety system in chronological order.

**Table 1.1: Car manufacturer and their LDW system implementation year [7]**

<b>Manufacturer</b>	<b>Year</b>
Nissan	2001
Toyota	2002
Honda	2003
Audi, BMW	2007
GM	2008
Mercedes	2009
Ford, Kia	2010

Over the last decade, the field of real-time computer vision and image processing has thrived. Intensive research has been continuously taking place in this field to provide for better vision based driver assistance systems.

One of the early works include the “time to lane crossing” (TLC) metric. It is a measure of the time remaining before a vehicle on a given trajectory will depart the road, which was first proposed by Godthelp [8]. The assumption of TLC is that the vehicle speed and steering angle does not change over the length of the predicted path. In [9], Pomerleau simplified the vehicle and road models such that it only uses information available from the computer vision system to estimate the vehicle's position and road curvature. LeBlanc

et. al. [10], for increasing the accuracy in vehicle path prediction, used complicated vehicle model. The lane crossing is predicted by comparing the predicted vehicle path and computer - vision generated image of the upcoming road.

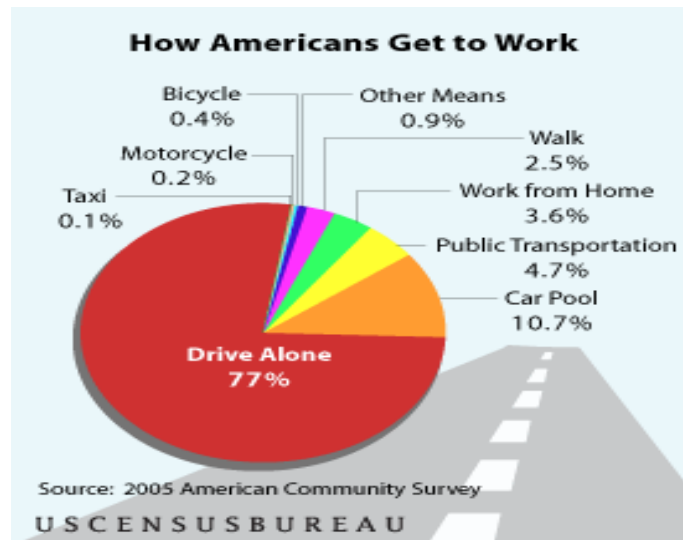
Teshima, et. al. [11] proposed a method in which a camera, installed on the vehicle, is used to acquire a series of front view images. These images are converted to top-view images by computing their respective homographies, which is computed by using the angle of vehicle's direction. The two top-view images from the consecutive frames are then compared and the Sum of Absolute Difference (SAD) is calculated. The same process is repeated for the entire hypothesis of the angle of direction. The hypothetical angle which gives the minimum SAD is considered as the heading angle of the vehicle. Once the heading angle is calculated the lateral position can be determined by accumulating the *sine* of the angle per frame. The problem with this method is that it requires a lot of computation while determining the heading angle which may not be practical for real time implementation.

Jung and Kelber [12] proposed a lane departure warning system based on a linear-parabolic model. In this method, image of the road acquired by the camera is divided into two regions – near field and far field. A linear function is used in near field, whereas in far field a quadratic function is used. The orientation of both lane boundaries in the near field is used to provide a measure of deviation from the center of the lane. However, this method involves a high degree of mathematical analysis as well making it inappropriate for real time operation.

### **1.3 Motivation**

Automobiles are an important part of our lives. A necessity in our daily lives - they have become an effective means of transportation for commute to work, and for many other purposes providing us with an easy lifestyle. However, the desire to lead an easy way of life raises many other issues. One such example is increasing demand of vehicles leading to a higher growth in population of vehicles on the road. The growth trend in number of automobiles brings along with it a higher probability of road accidents. As of 2010, one billion vehicles were in operation worldwide [13]. According to a study reported by the Department of Transportation, in the United States alone there were 250 million registered passenger vehicles. About 87.7% of commuters in the U.S drive to work with 77% driving alone in 2005 as per American Community Survey [14].

Studies show road accidents as one of the leading causes of death across the world accounting for 2.2% of overall death. National Highway Traffic Safety Administration (NHTSA) claims that at least one accident takes place every minute. According to the World Health Organization (WHO), each day an estimate of nearly 3,500 people die due to vehicle traffic crashes across the world which sums up to annual death of about 1.2 millions. In the United States, an average of 40,000 deaths is reported every year [15].



**Figure 1.2: Modes of transportation for commute in the U.S. [14]**

Majority of road accidents results from driver's inattention or distraction, driving under the influence of alcohol and drugs, reckless driving, and sleepiness/drowsiness while driving. Statistics show that drunken driving accounts for 40% of the fatal casualties in the U.S. Similarly, more than 33% accidents are caused due to reckless driving [16].

Increasing trend of road crashes has instilled growing concerns among driving population. While all the factors contributing to such hazards cannot be completely mitigated, people have become aware and are taking necessary precautionary steps. LDW systems, for the same reason, are becoming popular these days. The act of warning the driver and giving enough time to take preventive actions when the vehicle starts to drift from its lane is believed to alleviate the number of accidents. Analysis of the nature of accidents reported by the police showed that these types of systems have ability to avert or reduce the risk of 7,529 fatal and 37,000 injury accidents in the U.S. on an annual

basis [17]. Table 1.2 discusses briefly the effectiveness of LDW systems in different countries as identified from various studies.

**Table 1.2: Effectiveness of LDW systems [18]**

<u>Effectiveness</u>	<u>Source</u>
10% reduction in passenger car road departure crashes 30% reduction in heavy truck road departure crashes	Effectiveness estimated through mathematical modeling by Pomerleau et al., in the USA. Accidents due to system failure caused by adverse environmental conditions, missing lane boundaries, and low speed were not considered for estimation. Target Population excluded 75% of the accidents where alcohol intoxication was involved.
12% of fatalities, 9% of severe, 5% slight	Bosch (2005) in COWI (2006) estimated casualty savings in Germany.
Avoidance, 25% head on collisions, 25% left roadway collisions 25% side collisions Mitigation, 25% head on collisions, 15% left roadway collisions	Abele et al. (2005) estimated effectiveness based on changes to reaction time. Information related to head-on and left roadway collisions is clear, however the effectiveness of LDW for mitigating injuries in side collisions is not clear



24% reduction in singular accidents	Schermers (2000) cited in Malone et al. 2006. US Study
20% for cars	Alkim et al. (2007) based on FOT pilot using 19 Volkswagen Passats for 4 month period in one region of the Netherlands. Assumed representative of the Netherlands as a whole.

LDW systems may not be 100 percent effective. Different factors come into play in hindering these systems from being fully robust. However, even though in a small scale the benefits that they provide certainly cannot be neglected. Thus, research and development in this area has drawn more attention over the past few years.

Moreover, in comparison to other sensor technologies used, the camera-based systems have gained much popularity. Data analyses of images from the camera using image processing tools have a wide potential in such applications. Vision-based systems offer a number of advantages such as longer field of view, useful information extraction of surrounding environment. General working principle of the camera-based systems is to track lane markers on the road to see if the vehicle is straying outside the lane. But there is no guarantee that the lane markers will always be present on the road. Various environmental factors such as rain and heat might wear out the lane markings. Also, the presence of snow on the road might affect the visibility of road marking. Hence, a system is required which makes use of lane marking when available, but does work when lane markings are not available for short duration, and such is the goal of this study.

## **1.4 Scope**

In the following chapters, we describe in detail the estimation of vehicle's lateral position which acts as a base for the design and development of an on-board virtual rumble strip lane departure warning system. In Chapter 2, we explain the overview of the research. Chapter 3 discusses about the details of the optical flow point tracking system. In Chapter 4, we present and discuss data collected through experiments and road tests. Finally, in Chapter 5 we conclude and provide recommendations.

## 2. Overview of the Lane Departure Warning System

### 2.1 LDW System

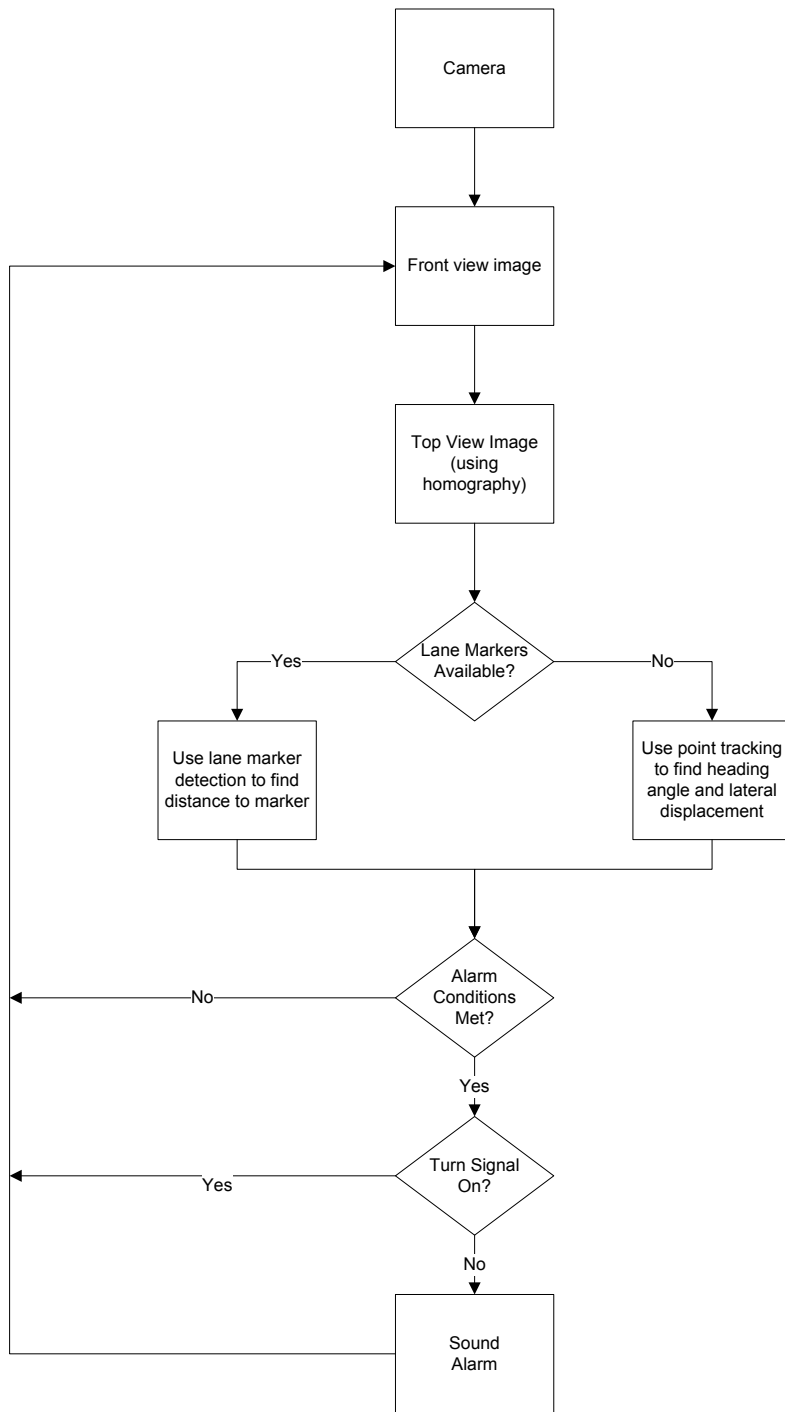
Before moving on to the details, a brief description of the system will be given in this chapter. This section will explain different parts of the system and discuss their relationship with each other. Figure 2.1 shows a flowchart of how the system works to warn the driver. A camera is mounted on the windshield inside the vehicle with the help of a suction cup. The camera is chosen such that it can do a live streaming of video as continuous sequence of road images will be needed. Camera calibration might be needed if a wide angle lens is used that imposes different types of distortions such as “fish-eye” effect. The calibration process helps to nullify the effects caused by the wide lens. Details of the camera used for this study are given in Chapter 4.

A series of front-view images of the road is taken by the camera. Image warping is performed on the front-view images to convert them into top-view images. This is done because when a scene is viewed through a camera lens, the points on the plane are subjected to a perspective transform. Hence, for better analysis of images and its features it is important that the perspective effect be removed so that the information content is homogeneously distributed among all the pixels. The transformation is achieved using homography for which the parameters for the homography matrix is calculated beforehand.

In the field of computer vision, any two images of the same planar surface in space are related by a homography. Homography refers to the concept of mapping of points and lines on projective plane. According to Hartley and Zisserman [19], "homography is an invertible mapping from projective plane to itself such that three points lie on the same line if and only if their mapped points are collinear". Homographies can be estimated between images through correspondence points and the geometrical relations. In our study, with the help of OpenCV [20] use multiple views of an object (chessboard, in our case) to compute individual rotation and translations of each view to determine the homography matrix.

Also for the purpose of the enhancement of image the histogram equalization is also included in the image processing. That is, the brightness distribution is equalized thereby increasing the contrast of the image. This is conducted so that there is sufficient texture variations to detect good features.

After the conversion, two consecutive top-view images are compared and analyzed to extract useful information. Images are checked to see if the lane markers are available. Depending upon whether they are present or not, one of the two algorithms is chosen (i.e., the lane marker detection and point tracking) to estimate the lateral displacement of the vehicle. The two algorithms are discussed in later sections of this chapter.

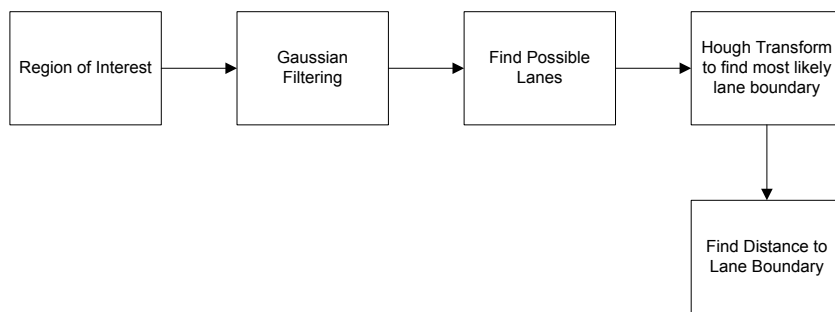


**Figure 2.1: Lane departure warning system flowchart**

Once the distance from the edge of the road to the vehicle is calculated, a system check is performed to see whether the criteria for warning has been met. Moreover, if the turn signal has not been activated the system would sound the alarm; otherwise the system will repeat the procedure of taking front view images and comparing the top-view images.

## 2.2 Lane Marker Detection

Figure 2.2 shows the block diagram of the lane marker detection algorithm.

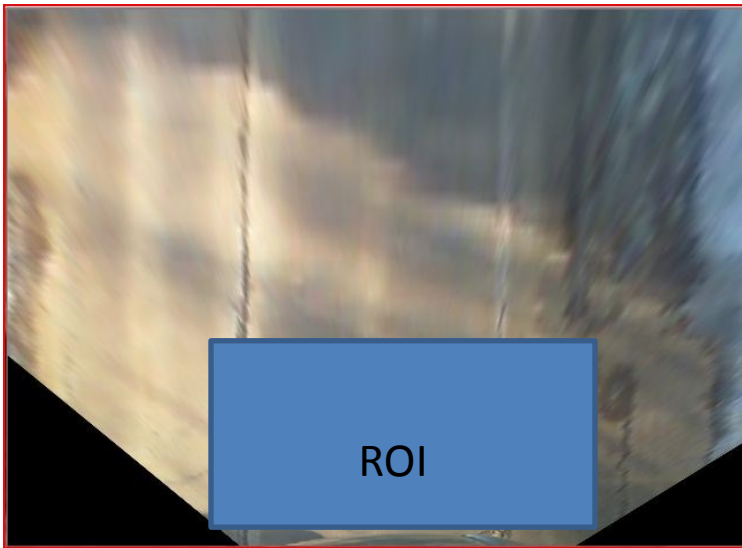


**Figure 2.2: Lane marker detection block diagram**

### 2.2.1 Region of Interest (ROI)

The data information contained in a digital image is huge. Given that the size of the image is large, if the whole image is to be processed it would consume a lot of time. In addition, all the information that an image holds may not be of use for the purpose of lane detection. Hence, instead of analyzing the image as a whole, only a sub region of the image that is of interest is selected. The analysis of specified region only would greatly speeds up the system operation and, thus, decreases its processing time.

For the lane detection purpose, a rectangular ROI of 300-pixel by 165-pixel is chosen from a 640-pixel by 480-pixel top-view image. It is located around the center of the car with a width of about 15 feet and is just wide enough to fit the width of a standard lane on a highway (about 12 feet). The ROI starts at the hood of the vehicle and goes up to about 10 feet in front of the vehicle.



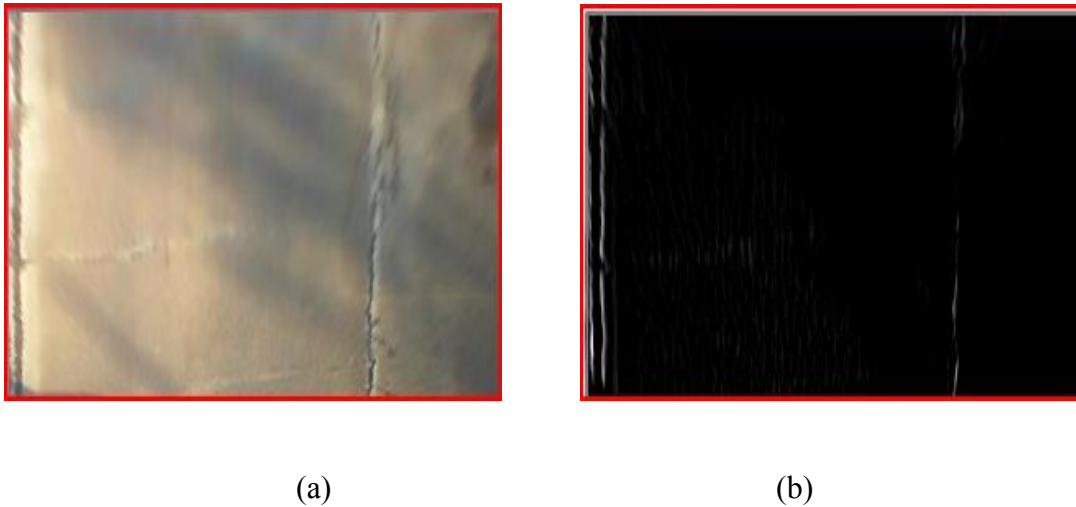
(a)

(b)

**Figure 2.3: : Top - view image and ROI (a) A full top - view image (b) ROI of the top - view image**

### 2.2.2 Gaussian Filtering

After selection of the ROI, it is converted from a color image to a grayscale image before filtering. Filtering is performed in order to find possible bright vertical lines (lane markers) on a dark background, which is the road. In addition, it also helps to remove the effects of noise and imperfections in an image. This is done by Gaussian smoothing which takes the weighted average of neighboring pixels and sets the pixel to that value [21].



**Figure 2.4: ROI filtering (a) before filtering (b) after filtering**

For the purpose of smoothing a two - dimensional convolution, a Gaussian kernel is used as an operator to apply the transformation to each pixel of the image. A Gaussian kernel refers to the normal distribution curve that is used to take average of pixels for smoothing. To detect lines in the vertical direction on image, a Gaussian smoothing is



applied in the vertical direction, while the derivative of Gaussian smoothing is used in the horizontal direction [22]. This process helps to single out those vertical areas where the change in contrast is high. The Gaussian filtering has a high response to lane markers which helps identify the pixels corresponding to the possible lane markers. Separable Gaussian kernel is used because of its efficient implementation and it is faster than using a non separable kernel.

### **2.2.3 Hough Transform for Lane Detection**

The output of the filtering process is a collection of pixels pertaining to the possible edge of the lane. Finding a geometric object in those set of pixels is what we are concerned about. In our case, the object is known to be line segments, that is, lines with vertical orientation to be more specific. Hence, these pixels need to be grouped together in order to find the vertical lines in the image. Moreover, all the vertical lines found in the image may not be of only lane markers. Edge points of other unnecessary features may also pass through the filtering process thereby providing us with vertical lines in stage. The vertical lines related to unwanted features should be omitted. The technique used to identify the vertical lines in the filtered image and keep hold of the most likely edge of the lane is the Hough transform [23].

An edge pixel can be represented as  $(x, y)$  coordinates in geometric space. The number of lines passing through this point is indefinite. Any straight line can be defined as

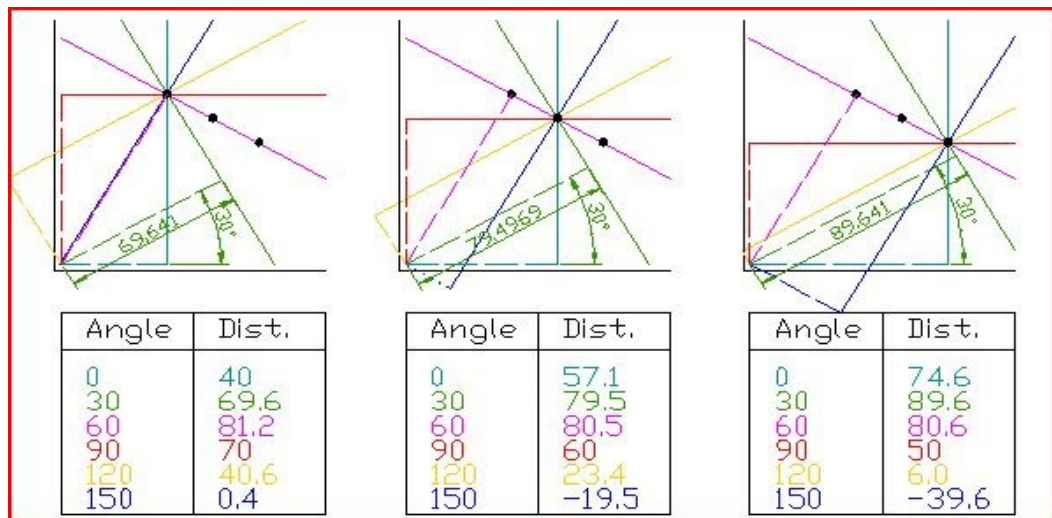
$$y = mx + c$$

where  $m$  is the slope of the line and  $c$  its  $y$ - intercept. However, for vertical lines the slope is undefined. Hence, they can be represented in the form of polar coordinates as

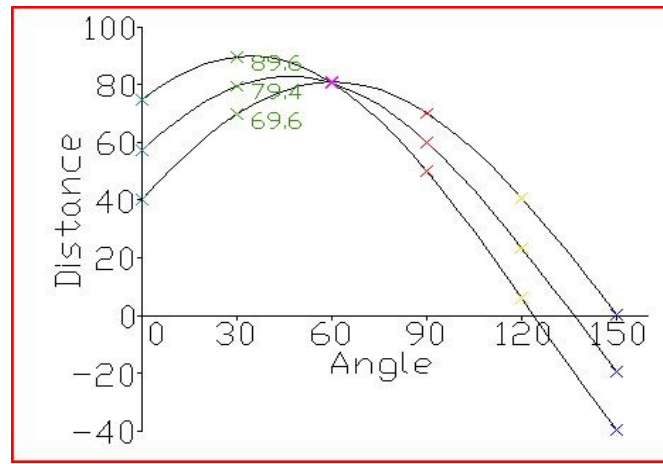
$$r = x \cos\theta + y \sin\theta$$

where  $r$  is the perpendicular distance from the line to the origin and  $\theta$  is the angle made by  $r$  with the  $x$ -axis.

Each straight line can be represented as a unique point  $(r, \theta)$  in the Hough's parameter space. If  $(r, \theta)$  were to be plotted for different possible lines passing through  $(x, y)$  Cartesian points, it yields a curve in parameter space. Cartesian points that are collinear in the Cartesian space can be easily depicted in the Hough parameter space because their curves intersect at a common  $(r, \theta)$ . Figure 2.4 demonstrates the same concept, where three Cartesian data points lying on a straight line and their corresponding Hough parameter graph plots are shown.



(a)



(b)

**Figure 2.5: (a) Three data points with  $r$  and  $\theta$  for different lines passing through them (b) Hough parameter graph plot showing the intersection of curves representing the line in which all three data points lie on [23].**

The Hough transform uses a voting scheme to detect the probable edge of the lane. A two dimensional array – accumulator – is used in case of the linear Hough transform to correspond to the two parameters, i.e.  $(r, \theta)$  in the Hough's parameter space. Whenever a curve is intersected, the Hough's parameters are obtained for the intersection point. The value of the accumulator bin representing this unique parameter pair is increased. After all points have been processed, the algorithm looks for the accumulator bin with the highest value, that is, to find the local maxima [24]. A threshold can also be applied to find the possible extracted line, which in our case is the lane. A suitable threshold value for our algorithm to detect the lane was found to be 25. These steps are first performed on

the left half of the image to find the left lane boundary and then on the right half to locate the right lane boundary.

#### **2.2.4 Calculate distance to lane boundary**

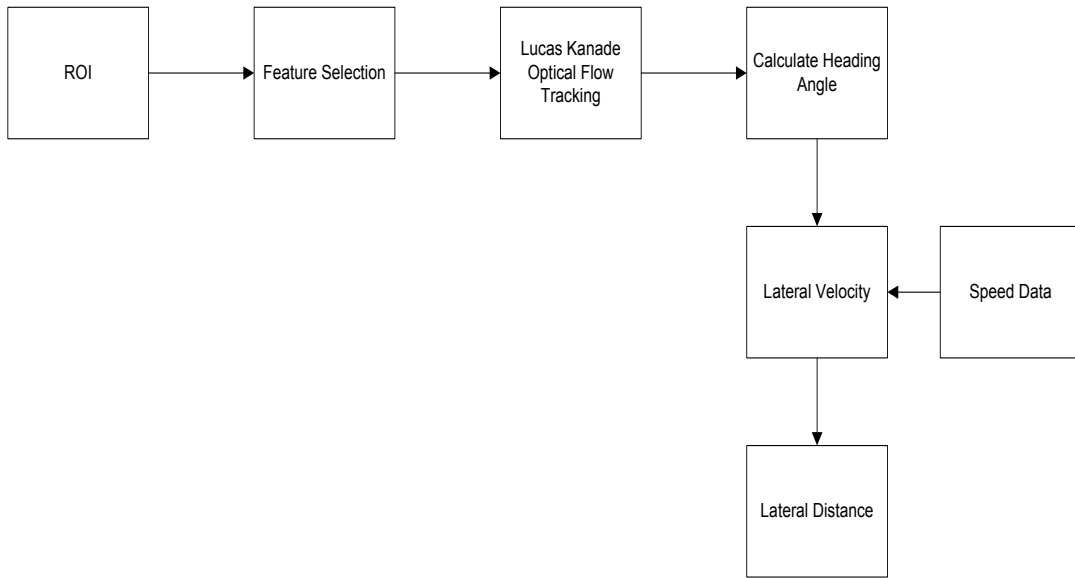
Once the lane boundaries are detected, the next step is to find the distance of lane boundaries from the center of the car. This is done by converting the pixels into units of feet. Depending upon the distance to the lane markers, an appropriate action is taken – whether to sound an alarm or not.

### **2.3 Point Tracking**

This section deals with the Point Tracking algorithm of the system. The point tracking algorithm is a combination of feature selection and optical flow method. Since the focus of this study is on the point tracking algorithm rather than the lane marker detection algorithm, the in-depth details of the point tracking algorithm will be deferred until next chapter. The block diagram of the point tracking algorithm is shown in Figure 2.5.

#### **2.3.1 ROI**

The concept of ROI is the same as in the previous case. The only difference is its size. In the point tracking algorithm the chosen dimension is larger because the more the area that can be analyzed, the higher are the chances of finding objects/features to track. Hence, an ROI of size 640- pixel by 210- pixel is selected.



**Figure 2.6: Point tracking block diagram**

### 2.3.2 Feature Selection

It is in our best interest to know about the vehicle orientation by extracting information from the images. In order to do so, it is often required that the object(s) in the images be followed. Situations may arise where one has to track unidentified objects, i.e. track those objects that are significant to follow based on the motion of the vehicle. Therefore, it is necessary to identify which visual key points are important for our purpose. This step helps to recognize those points by using the Shi and Tomasi method [25].

### **2.3.3 Feature Tracking**

This part of the system tracks the features between two consecutive image frames to assess the motion between them. The tracking technique used here is pyramidal form of the Lucas-Kanade Optical flow [26]. Also, this part helps to eliminate the points that are not suitable for tracking in our system.

### **2.3.4 Heading Angle and Lateral Displacement Computation**

In the previous section, the new pixel coordinates of good features in an image were found in its subsequent frame. Utilizing the old and new pixel coordinates of the features an angle is calculated to determine the path the vehicle is heading. This angle along with the speed information of the vehicle is used to estimate the lateral displacement of the vehicle from its previous position.

### **3. Description of the Point Tracking Approach**

Video tracking is generally used to follow particular object or objects of interest over time using a camera. Relating the concerned object(s) in a series of video frames is the main idea of tracking. To understand the motion of an object, the basic step would be to identify it in the scene. Identification of interested objects can be done through various algorithms such as color histogram, the Hough transform, and contours to name a few.

Nevertheless, tracking objects could be a difficult task. This is true especially when objects change their shape and appearance in the course of time. Moreover, problems may arise due to occlusion in the scene, motion of the camera, sudden change in motion of the object, etc. In addition to that, there may be certain cases which require tracking things that are yet to be recognized, and hence could be challenging. Unidentified objects may be tracked because of their peculiar motion or their motion provides some useful information.

#### **3.1 Feature Selection**

In a visual scene there are different features available. Knowing which image feature would be suitable to track is very important. Choosing the right feature may differ according to scenarios for which tracking is performed and the kind of information that is needed. In general, the most important property the potential feature or feature set should possess is its uniqueness. The probability of locating a unique feature in following frames is higher than distinguishing identical or similar features. Also, in order to be able to

compare the features in subsequent frames the candidate features should be parameterizable.

In our system the types of features we are interested in are point features, also referred to as corners. Corners are the unique two dimensional image points which have significant changes. They could be thought of as junction of two edges going toward different directions. To be more specific, corners can be defined as those regions in an image which have high intensity changes in two orthogonal directions, i.e., horizontal and vertical directions. The process of detecting and extracting corners from an image is called Corner Detection. The most common and influential approach to corners and its detection was given by Harris [27] which was later improved by Shi-Tomasi [26]. Harris corners detection became popular because of its rigidness and better results in presence of image noise, illumination change, change in scale, and image rotation. According to Harris, “Corners are places in the image where the autocorrelation matrix of the second derivatives has two large eigenvalues” [26]. Given an image, let us consider any pixel  $(x, y)$ . The image function at  $(x, y)$  is  $I(x, y)$ . If this point is shifted by  $(a, b)$ , we could check how alike are the functions at  $(x, y)$  and  $(x+a, y+b)$  by taking the sum squared difference of intensity between them which is given by

$$M(a,b) = \sum_x \sum_y w(x, y) (I(x, y) - I(x+a, y+b))^2 \quad (1)$$

Here,  $w(x, y)$  is a weighted function either constant or Gaussian. If the calculated change in image intensities is nearly equal to zero it implies that the image patches are nearly



constant patches. The larger value means that the image patches are unique and could be a good point to track. Hence, the large value of the function  $M(a,b)$  is desired at patches to be considered as corners.

Using the Taylor's series approximation, the equation can be written as

$$M(a,b) = [a \ b] \ G \begin{bmatrix} a \\ b \end{bmatrix} \quad (2)$$

where  $G = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$ ;  $I_x$  and  $I_y$  are partial derivatives of image gradient.

If  $\lambda_1$  and  $\lambda_2$  are two eigenvalues of the autocorrelation matrix  $G$ , there are three possible cases. If both are small in value, it represents a flat region of unvarying intensity. A large  $\lambda_1$  and a small  $\lambda_2$  (or vice versa) indicates only sharp change in one direction which is an attribute of an edge. Finally, two large eigenvalues imply that there is high contrast in perpendicular directions, indicating the presence of a corner. Calculation of eigenvalues of the autocorrelation matrix  $G$  through eigen decomposition requires extensive computation. Harris resolved this issue by suggesting a score system that requires calculating only determinant and trace of  $G$ , and that is enough to check the presence of a corner. For each window, a score  $R$  is computed and if  $R$  is greater than a certain value it is considered as a corner.

$$R = \det(G) - \alpha (\text{trace } G)^2 \quad (4)$$

where,  $\alpha$  is a constant and

$$R = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2 \quad [28] \quad (5)$$

Measurement of cornerness of a point was further simplified by the discovery of Shi and Tomasi which gave even better results. According Shi and Tomasi [25], the score R is calculated as the minimum of the two eigenvalues, i.e.  $R = \min(\lambda_1, \lambda_2)$ . So, if the minimum of the eigenvalues is greater than the threshold, then it is a good interest point [25].

### **3.2 The Lucas-Kanade Optical Flow Based Tracking**

The main idea of optical flow is to find 2 - dimensional motion field in an image sequence. In other words, optical flow is the representation of apparent motion of image pixels from one frame to another, which results from the motion of the camera or from moving objects in the scene. Feature trackers based on optical flow locate subject of interest(s) from one image frame to another to obtain information about the motion of the subject of interest or the camera itself. A major assumption of this idea is that images have strong similarity in terms of its content when they are taken at near time instants. For a feature point in an image represented by pixel coordinates (x, y) the purpose of the tracker is to search for a point (x', y') in subsequent frame that is similar to point (x, y) in the previous one.

There are basically two major methodologies of optical flow techniques for tracking - sparse optical flow and dense optical flow [29]. For a point feature with coordinates (x, y) in an image there is an inter-frame displacement (dx, dy) such that the point feature can

be represented in its subsequent frame as  $(x+dx, y+dy)$ . This is, the translation component which is found for each and every frame. A dense optical flow technique attempts to calculate this displacement or the equivalent velocity for every pixel in the image, while a sparse technique requires defining only a subset of pixels that have some desired qualities for tracking and is dependent only on the local information obtained from a window around those subset of pixels. The Lucas-Kanade (LK) tracker is based on sparse technique whose inputs are the corners found from the feature selection method as discussed in the previous section.

Few assumptions have been made for the working of the LK tracker. First is the "brightness consistency", which states that the brightness of pixels that are to be tracked in consecutive images do not change although their location change and that they have the same appearance over that instant. Next is the "spatial coherence" under which it is assumed that the points in the image belong to the same surface and have similar motion.

According to the first assumption, the intensity of a pixel  $(x, y)$  at time  $t$  should be equal to the intensity of the new pixel location  $(x+dx, y+dy)$  at time  $t+dt$ . It could be represented as

$$I(x, y, t) = I(x+dx, y+dy, t+dt) \quad (6)$$

Finding the Taylor series for the right hand side of the equation, we get an approximation in terms of its derivative:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (7)$$

$$I_x dx + I_y dy + I_t dt = 0 \quad (8)$$

Dividing by  $dt$ , we get

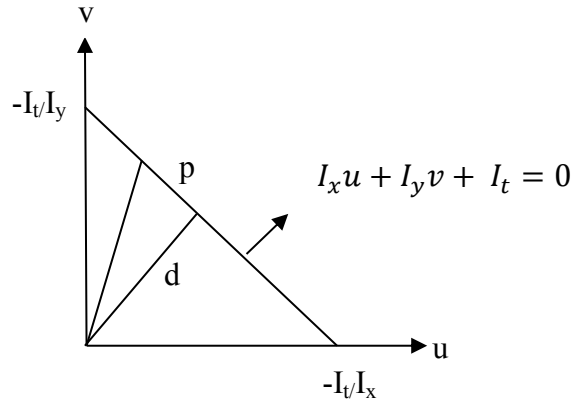
$$I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t = 0 \quad (9)$$

$$I_x u + I_y v + I_t = 0 \quad (10)$$

Here,  $u = \frac{dx}{dt}$  and  $v = \frac{dy}{dt}$  are the changes in  $x$  and  $y$  coordinates with respect to time and together they are known as optical flow. Equation (10) is known as the optical flow equation. Furthermore, the equation can be re-arranged as

$$v = \frac{I_x}{I_y} u - \frac{I_t}{I_y} \quad (11)$$

which is the equation of a line.



**Figure 3.1 Representation of optical flow equation [20]**

The optical flow vector lies on the line, but its exact values cannot be computed since we have only one equation and two constraints. Hence, the system becomes under constrained. The optical flow vector can be divided into two parts, the parallel flow  $p$  and the normal flow  $d$ . The parallel flow changes depending upon where the optical flow lies on the line, however, the normal component of the motion is fixed regardless of the optical flow vector and could be determined as

$$d = \frac{I_t}{\sqrt{I_x^2 + I_y^2}} \quad (12)$$

It is impossible to compute the optical flow vector if only a pixel is considered. The normal flow only detects an edge, not a corner, which does not help in finding the motion of the object. So, considering the second assumption that the neighboring pixels have similar motion, we can consider a window around a pixel. Suppose we have a local patch of a 3 by 3 window. For each pixel in this window we will have an optical flow equation,

thus, a total of 9 equations. We now have an over constrained system which could be solved for the optical flow vector and can be written in matrix form as

$$\begin{bmatrix} I_{x1} & I_{y1} \\ \cdot & \cdot \\ \cdot & \cdot \\ I_{x9} & I_{y9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ \cdot \\ \cdot \\ -I_{t9} \end{bmatrix} \quad (13)$$

$$Bu = I_t \quad (14)$$

$$u = (B^T B)^{-1} B^T I_t \quad (15)$$

From the above equation, we can compute the optical flow velocity  $u$ . Alternatively, if  $I$  and  $J$  are two images the local window can be defined by values  $w_x$  and  $w_y$  such that the window size is given by  $(2w_x + 1) \times (2w_y + 1)$ . For a 3 by 3 window, we choose each  $w_x$  and  $w_y$  to be 1 pixel. Then, the image translation component, i.e., the displacement vector is sought such that the residual function given by equation (16) is minimized

$$E(d) = \sum_{x=-w_x}^{w_x} \sum_{y=-w_y}^{w_y} [I(x, y) - J(x + dx, y + dy)]^2 \quad (16)$$

For the accuracy purpose, it is better to choose a small integration window. However, the preference of small window comes with a trade-off of allowing to track only smaller motions. Meanwhile, selecting a larger integration window increases the chances of incorrect tracking. Hence to accommodate for larger motions still keeping the robustness of the tracking algorithm, pyramidal form of the Lucas Kanade optical flow is used where

the images are processed in a pyramid fashion in which the optical flow is first computed at a top layer. Then the results from this layer is used as an initial guess for the next layer in the pyramid, and so forth until the lowest level is reached which is the image pixel. For each level forward in the pyramid, the resolution of a level becomes half the resolution of level below it.

Let the original image  $I$  be represented as  $I^0$ ,  $I^0$  being the zero<sup>th</sup> layer of the image pyramid. The next level  $I^1$  is computed from  $I^0$ ,  $I^2$  from  $I^1$ , and so on. Generalizing the image pyramid levels, we have  $I^L$ ,  $L = 0,1,2,3,\dots,L_m$ ;  $L_m$  being the height of the pyramid. At each level, the coordinate  $(a_x, a_y)$  of a point 'a' can be calculated by

$$a_x^L = a_x / 2^L \quad (17)$$

and

$$a_y^L = a_y / 2^L \quad (18)$$

The initial guess for the optical flow at the highest level of the pyramid  $L_m$  is set as  $g^{L_m} = (0, 0)$ . The residual displacement vector  $d^{L_m}$  for the level is computed such that the error function is minimized to get the optical flow. This optical flow is propagated to the next level of the pyramid as the initial optical flow value for that level, which is given by

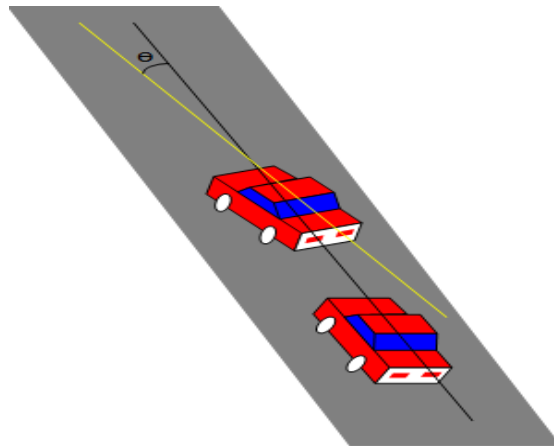
$$g^{L_m-1} = 2(g^{L_m} + d^{L_m}) \quad (19)$$

These steps are repeated for each pyramid level until the finest resolution is reached which is the original image (i.e., level 0). For all the levels, the size of the integration window is constant, i.e.,  $(2w_x + 1) \times (2w_y + 1)$  [30].

### 3.3 Vehicle's Lateral Characteristics

The development of a LDW system requires the vehicle's lateral characteristics to be known. Lateral characteristics include the lateral position of the vehicle with respect to the edge of the road. In order to determine the lateral position of the vehicle, computation of the vehicle's lateral speed and heading angle is required. Lateral speed needs, for its calculation, the heading angle to be estimated first. In addition to that, it is necessary to know the forward speed of the vehicle.

Let  $a$  be the forward speed of the vehicle, the vertical axis be the longitudinal axis of the vehicle, the horizontal axis be the lateral axis at right angles to the longitudinal axis, and  $\Theta$  be the angle between the longitudinal axis and the forward moving direction of the vehicle, that is, the heading angle.



**Figure 3.2: Heading angle estimation**



If  $x$  is the current position of the vehicle, the lateral speed can be given by

$$\dot{x} = a \sin(\Theta) \quad (20)$$

where  $\dot{x}$  is the derivative of  $x$  with respect to  $t$ . If  $\Theta$  is small, we can approximate the above equation as

$$\dot{x} = a \Theta \quad (21)$$

The heading angle  $\Theta$  can be determined from the locations of features in the consecutive frames found from the Lucas Kanade algorithm as discussed in Section 2.2. Let us consider the number of features tracked over two successive image frames to be  $n$ . Considering just one of the many tracked features and naming it as feature  $m$ , we can define its coordinates as  $x_{i+1,m}$  to be the  $x$ -coordinate at frame  $i+1$ ,  $x_{i,m}$  to be the  $x$ -coordinate at frame  $i$ ,  $y_{i+1,m}$  to be the  $y$ -coordinate at frame  $i+1$ , and  $y_{i,m}$  to be the  $y$ -coordinate at frame  $i$ . Let us also define the displacements  $\Delta y_m = y_{i+1,m} - y_{i,m}$  and  $\Delta x = x_{i+1,m} - x_{i,m}$ , where  $\Delta y_m$  always greater than zero.

Meanwhile, the displacement along the  $x$  direction can have one of the following three conditions; it can be either greater than or less than or equal to zero. If  $\Delta x_m > 0$ , then the heading angle  $\Theta_m = - (90^\circ - \alpha)$  indicating the vehicle is moving to its left. For the vehicle moving to the right,  $\Delta x_m < 0$  and  $\Theta_m$  is given by  $(90^\circ - \alpha)$ . Lastly, if  $\Delta x_m$  equals 0 then we set  $\Theta_m = 0$ . Here,  $\alpha$  is given by

$$\alpha = \frac{180^\circ * \beta_m}{\pi} \quad (22)$$

and

$$\beta_m = \tan^{-1} \left| \frac{\Delta y_m}{\Delta x_m} \right| \quad (23)$$

After calculating individual heading angle for each tracked feature, the final heading angle,  $\Theta$ , from one frame to the other is determined by averaging the individual heading angles of the  $n$  features which can be represented as

$$\Theta = \frac{1}{n} \sum_{m=1}^n \theta_m \quad (24)$$

## 4. Experiments and Results

In order to determine the lateral position of the vehicle, a series of experiments were performed. We used visual studio and an open source library known as OpenCV (Open Source Computer Vision Library) [20]. It is a library of computer vision functions which focuses on real-time image processing. Developed by Intel, this library is written in C and C++. Interfaces with other languages such as Matlab, Python, etc. are also available, but for our purpose we chose C++. We used OpenCV 2.3.1 version to implement our algorithm.

### 4.1 Camera



**Figure 4.1: Microsoft LifeCam Cinema 720p HD webcam used for the experiments**

The in-vehicle camera for our project required having characteristics such as live streaming as continuous sequence of road images will be needed, reasonable angle of the lens to cover enough visual scene, clear video, etc. The camera used for this study is Microsoft LifeCam Cinema 720p HD webcam (Item Model Number: H5D-00003). The

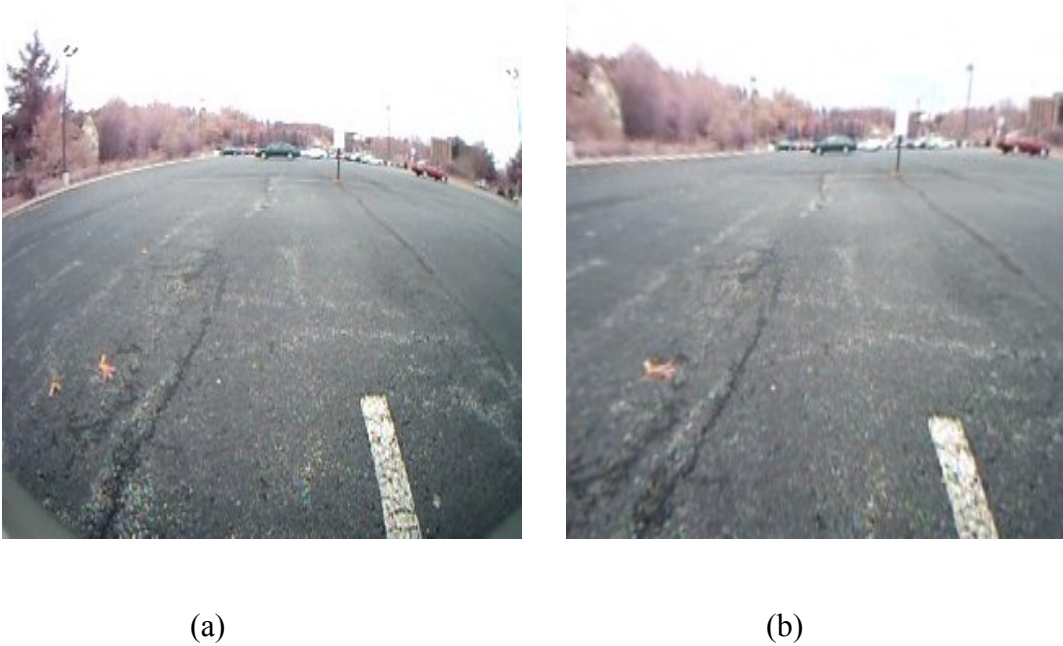
rate at which the video can be taken by this camera is up to 30 frames per second. The camera is capable of conducting a real - time data streaming, and the angle of lens is about 74 degrees.



**Figure 4.2: Camera installed in the vehicle with the help of a suction cup**

The camera was positioned at the center (roughly) of the windshield inside the vehicle and was mounted with the help of a suction cup attached to it (as shown in red box in Figure 4.2). It was placed overlooking the road at a certain angle such that the area covered by the lens of the camera is the road about 50 feet from the vehicle. We chose an angle of 15 degrees with the horizontal so that we had enough area of field to cover the road edge for our tracking.

Initially, a camera with a wide angle of view was used for the experiment. The angle of lens was near about 120 degrees. Having a great width of lens caused the images to be distorted and have a bulging effect. This effect is known as barrel or fisheye effect. To reduce the effect caused by this lens, the camera was calibrated to find its distortion parameters, which would then be applied to the images for its correction. However, we found that such ultra-wide lens are not needed for our purpose. So, we opted for aforementioned camera with a lesser angle of view i.e., 90 degrees.



**Figure 4.3: (a) Scene from the wide angle camera lens with distorted fish eye effect  
(b) Scene after removal of the fish eye effect through calibration**

## 4.2 Histogram Equalization

A series of images were taken through the camera while the vehicle moved. Histogram equalization was performed on the front view images for enhancement by increasing the contrast.



(a)

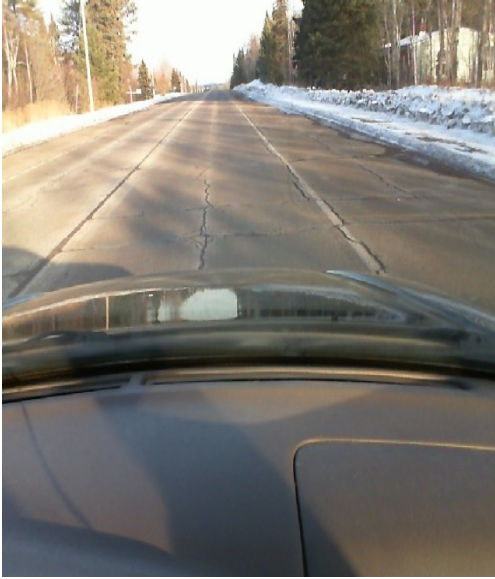
(b)

**Figure 4.4: (a) View of the from camera (b) View after histogram equalization**

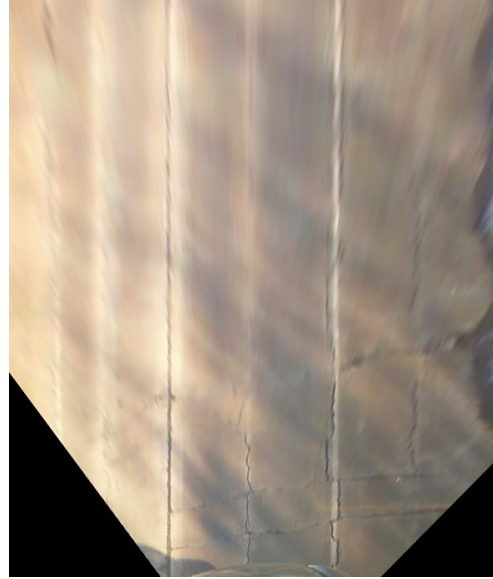
## 4.3 Front to Top-View Conversion

To remove the perspective effect that is induced by the lens of the camera in the front view images, the images were converted to their respective top-view images. Conversion to top view was done via homography (a homography matrix was calculated beforehand, which was then used during the processing of images). The images that resulted from this

conversion were such that the image was seen from a significant height. Hence, it is also known as a bird's eye view or aerial view image



(a)

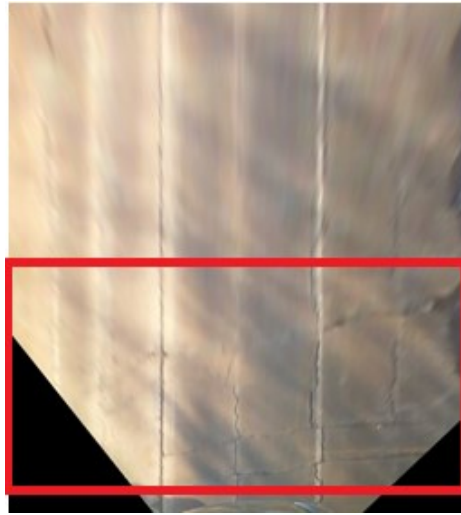


(b)

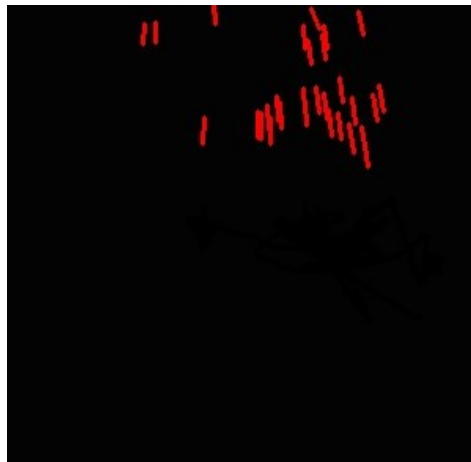
**Figure 4.5: (a) Front-view image (b) Conversion of front-view to top-view using homography**

#### **4.4 Region of Interest**

The processing of images were limited to only a portion of its bird's eye view image. The advantages of doing that are many. Firstly, it helps to speed up the computations by concentrating only on a small area. This allows our algorithm to run in real time more efficiently. Secondly, the distortion caused by the front-to-top view conversion can be neglected, which is prominent at the far edges of image. In Figure 4.6 ROI in top-view image is represented in the red box.



**Figure 4.6: Region of interest**



**Figure 4.7: Feature displacement**

#### **4.5 Road Test**

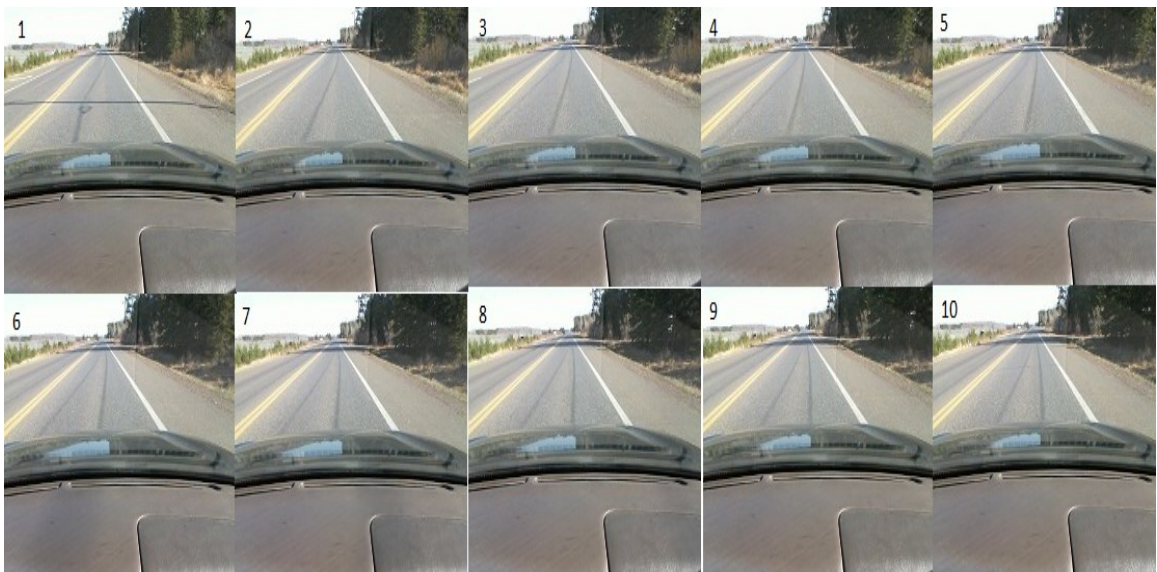
Several road tests were conducted to compute the heading angle through our algorithm and to evaluate the results. The test sites included Lavaque Road, Jean-Duluth Road



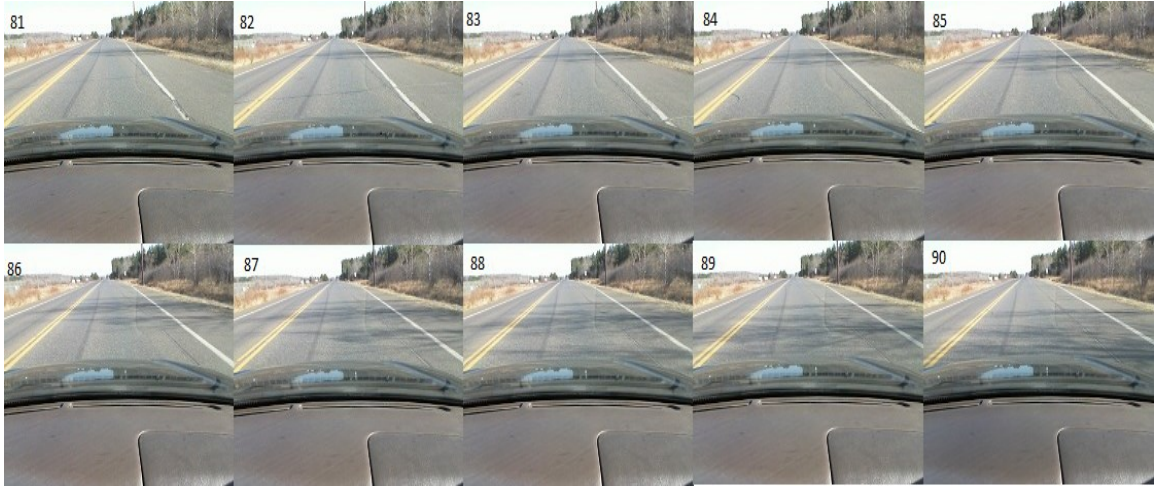
Martin Road, and US Highway 61. The speed on these roads ranges from 50 to 65 mph. Numerous tests were conducted on these roads on different days and time.

Since the algorithm does not address the road curvature issue, only a straight portion of roads were considered in the tests. While conducting the tests, large amount of data was obtained. In this Chapter, only the data on Jean-Duluth Road will be discussed.

During this test, a total of 200 frames were processed with time difference of roughly 100 milliseconds between each frames adding up to a total of 200 frames in 20 seconds. It would not be feasible to provide all 200 images in this thesis. Therefore, the total frames for two time intervals ( between 0 and 1 second, and between 8 and 9 seconds) are provided in this thesis. Figure 4.8 shows the first 10 images taken from the camera during the test in a duration of the first one second. Similarly, Figure 4.9 shows frames 81 to 90 captured between 8 and 9 seconds.



**Figure 4.8: Front view images from frames 1 to 10**



**Figure 4.9: Front - view images from frames 81 to 90**

Table 4.1 presents the corner points tracked between consecutive frames along with the angle calculated between them for image frames 1 and 2. The data for first 10 frames can be found in Appendix A.

**Table 4.1: Result of the test showing the points tracked and angle between them**

Frame Comparison code (Z)	Corner Ax	Corner Ay	Corner Bx	Corner By	Angle between A and B (Alpha)	No. of total Points in frame (Z-1)	No. of points found in frame Z
1						292	0
2	537.842	152.298	533.557	154.827	59.4527		
2	566.798	39.0564	553.675	54.804	39.8056		
2	382	114	398.574	150.165	-24.6213		
2	374	115	358.128	150.808	23.9051		
2	361.668	196.924	364.1	201.167	-29.8192		

2	296.807	132.631	305.813	149.351	-28.305		
2	366	114	347.477	149.769	27.3773		
2	408.51	140.383	406.808	144.445	22.7336		
2	337	113	328.77	147.927	13.2595		
2	298	114	304.792	149.27	-10.9001		
2	376	115	365.808	151.066	15.7793		
2	289	115	290.922	150.128	-3.13171		
2	360.917	194.841	357.705	198.249	43.304		
2	549	183	546.881	183.839	68.4086		
2	566.794	39.0647	553.683	54.7712	39.8545		
2	135	121	81.7853	135.047	75.2126		
2	606	167	602.733	169.033	58.1093		
2	557	130	549.707	165.905	11.4822		
2	300	115	299.454	150.105	0.890864		
2	324	113	323.084	147.696	1.51195		
2	295	114	295.106	148.997	-0.173468		
2	291	114	292.372	148.909	-2.25003		
2	258.145	206.626	260.506	206.722	-87.6632		
2	561.195	71.6656	556.204	72.8728	76.403		
2	333	113	337.301	147.45	-7.11655		
2	614.955	104.387	610.529	107.642	53.6709		
2	407.36	198.495	406.054	201.796	21.595		
2	364	114	343.273	149.246	30.4589		
2	269.555	191.733	271.486	196.236	-23.2087		
						292	29

In Table 4.1, the frame comparison code (Z) represents that images Z and Z-1 are compared to see how the orientation of the vehicle has changed from one frame to another. Corner Ax and Ay denotes the coordinate points of the individual features found in image Z-1 that are to be searched for in the next image Z. The total number of features in image Z-1 is also shown, but only the coordinates of those features found in both Z and Z-1 are given here.

The algorithm tracks features  $A_x$  and  $A_y$  in the subsequent frames to find their corresponding new locations  $B_x$  and  $B_y$ . Not all the features found in image  $Z-1$  could be tracked in the following image. Sometimes the features cannot be found or it may be that the feature did not pass a threshold of accuracy.

The last column gives the number of features that were able to be tracked. Once the coordinates of the same feature has been found, the angle (Alpha) between them is calculated which tells us how the feature has moved in those two images. The angle is determined for each feature and averaged for each  $Z$ . For instance, in  $Z = 2$  when images 1 and 2 are compared, 292 points were found in image 1 that were considered as good features to track. However, only 29 points could be tracked in image 2. Thus, the angle of these 29 points are estimated and averaged.

**Table 4.2: Average angle calculated from features**

Frame Comparison code (Z)	Average of Alpha for each Z (Beta)	
1	0	
2	16.0699	
3	0	
4	15.1962	
5	-0.494116	
6	48.2741	
7	48.8728	
8	17.4914	
9	13.7032	
10	-13.8123	
Number of frames considered for taking average in this batch		8

Average angle of this batch		18.162648
11	-20.1296	
12	14.2584	
13	24.2733	
14	12.0961	
15	-0.57142	
16	6.47175	
17	18.1311	
18	26.9381	
19	-7.80429	
20	7.085	
Number of frames considered for taking average in this batch		10
Average angle of this batch		8.074844

Table 4.2 shows the resulting angle (Beta) for each Z derived from averaging the feature angles. While averaging, a condition is placed such that if the number of features tracked is below a certain number (we have set it to be greater than 5) then its averaged angle (Beta) is set to zero. The reason for applying such a condition is that sometimes only one or two points may be tracked, and given that the algorithm may not always track features with 100 percent confidence it may skew the result. Thus, for each Z there should be at least 6 tracked features averaged, otherwise, the angle is by default set to zero.

After processing a batch of every ten number of Z's ( that is at every 1 second) average of the Beta angles of that batch is computed, which will be our final heading angle at that time with respect to the previous position. In table 4.2 it can be seen that after Z=10 and 20, average of those batches are taken to compute the final heading angle at time  $t = 1$

and 2 seconds to be 18.16 and 8.07 degrees respectively. Hence, using this heading angle (from  $t= 1$  to 19 seconds), the lateral displacement of the vehicle from the center of the road is computed as shown in Table 4.3.

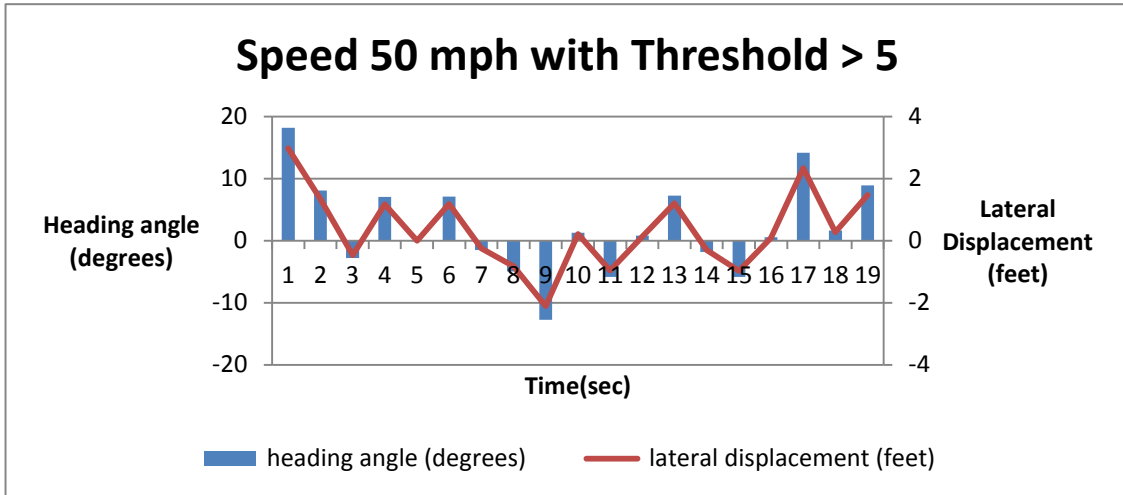
**Table 4.3: Values of heading angle and lateral displacement**

Total time ,t (in seconds)	Heading angle (degrees)	Lateral displacement (feet)
1	18.1626	2.97236
2	8.07484	1.33942
3	-2.77933	-0.462371
4	7.067	1.17315
5	-0.04038	-0.00672027
6	7.09555	1.17787
7	-1.51551	-0.25219
8	-4.963	-0.82494
9	-12.7655	-2.10698
10	1.28358	0.213603
11	-5.8179	-0.966587
12	0.787401	0.13104
13	7.23265	1.20051
14	-1.80271	-0.299968
15	-5.85942	-0.97346
16	0.541994	0.0902005
17	14.1561	2.33205
18	1.65034	0.274622
19	8.88192	1.47227

## 4.6 Results

In this section, we present graphical representations of data found while performing our tests. We show and discuss the results from the tests conducted at different speeds. The 50 mph speed tests were carried out on Martin Road, Jean-Duluth Road, and Lavaque Road. The 65 mph speed tests were conducted on Highway 61 outside the city of Duluth

limit. The data shown in Table 4.3 (for Jean-Duluth Road) are plotted in graph and shown in Figure 4.8.



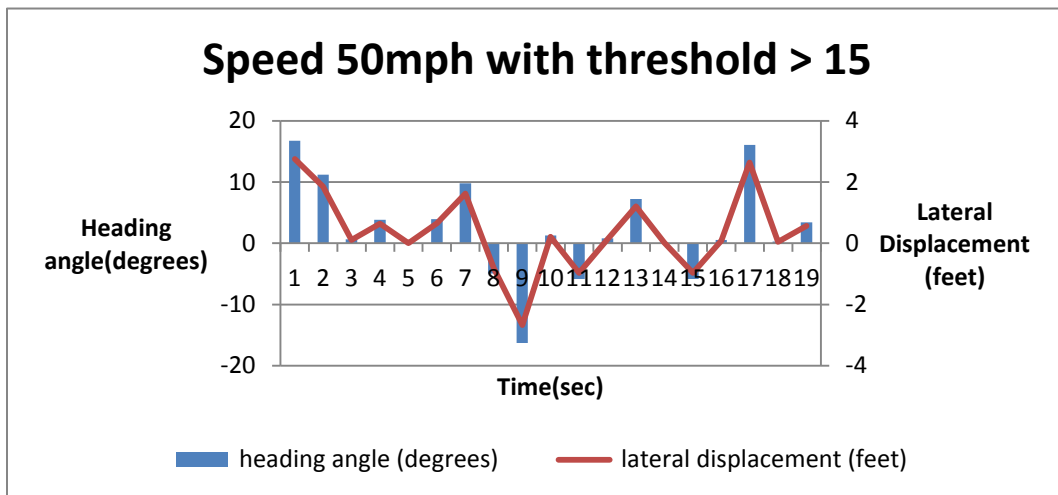
**Figure 4.10: Heading angle and lateral displacement with 50 mph speed**

In Figure 4.10, the heading angle and lateral displacement are plotted against time in seconds. We can see that the heading angle has both positive and negative values. The positive heading angle indicates that the vehicle is moving towards its right, and the corresponding lateral displacement represents deviation from the center of the lane. Likewise, negative heading angle means that the vehicle moving to the left. As we can see from Figure 4.8, the vehicle is heading slightly towards right from the first frame to the tenth frame (that is, between 0 and 1 second). Hence, a positive angle in Figure 4.10 at time  $t = 1$  second, which represents data from those first ten frames. Similarly, from Figure 4.9 (which gives all frames between 8 and 9 seconds), it can be seen that the vehicle drifts towards left, and therefore results to a negative angle at  $t = 9$  seconds. The

range of the heading angle lies between 18.7 degrees to -12.76 degrees, and the lateral displacement ranges from 2.97 feet right from the lane center to -2.11 feet on left.

Please note that while conducting the driving test, we tried to keep the vehicle straight with heading angle equals to zero at all times. However, we found that to keep the steering absolutely fixed was nearly impossible. Therefore, we experienced some changes in our heading angle, and hence we can see variations in the heading angle although we were driving on a straight road.

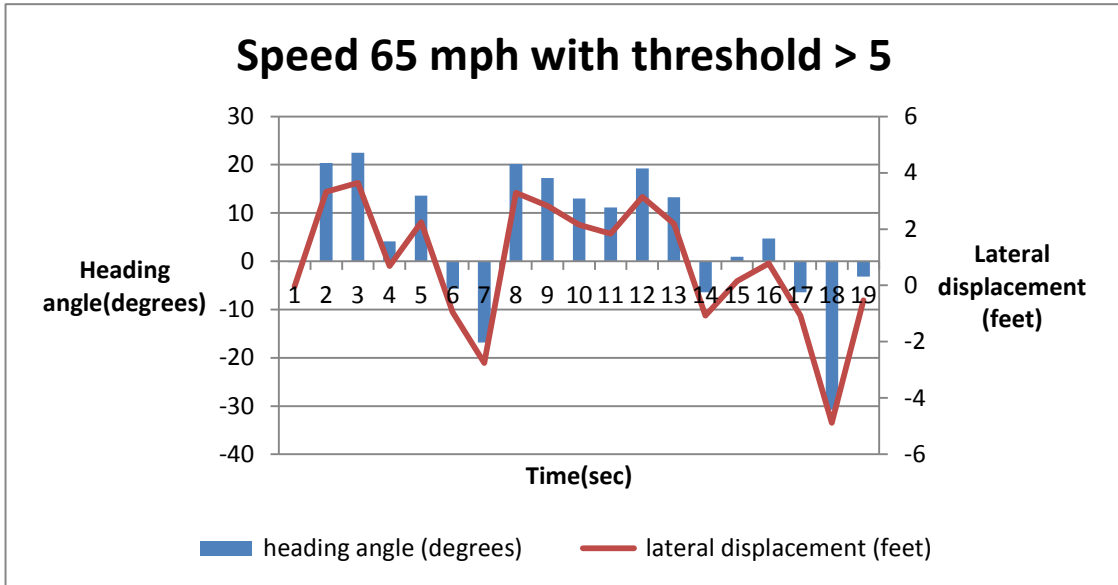
Also, the data was analyzed so that the effect of the threshold for the feature selection in the computation of heading angle and lateral displacement could be studied. The threshold was set to greater than 5 and 15. Figure 4.10 showed the results when the threshold was kept at greater than 5. Figure 4.11 shows results for the same set of images as in Figure 4.10, the only difference being the threshold set to greater than 15. Comparing Figure 4.8 and Figure 4.9, we cannot see much of a difference.



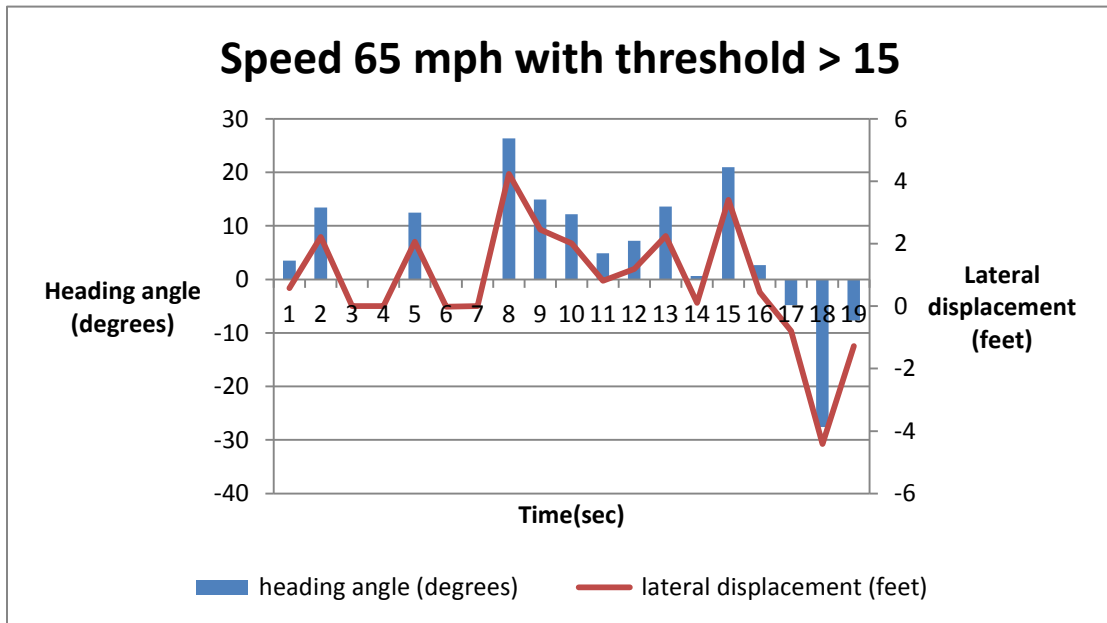
**Figure 4.11: Heading angle and lateral displacement with threshold > 15**



Similarly, for speed 65 mph test on Highway 61, the same conditions were applied for threshold and the results are shown in Figure 4.12 and Figure 4.13.



**Figure 4.12: Heading angle and lateral displacement at 65mph with threshold > 5**



**Figure 4.13: Heading angle and lateral displacement at 65mph with threshold > 15**

As stated above, the test was performed on a small straight section of the road. Ideally, the heading angle should be zero during the course of the test. But, in reality, maintaining a zero angle would be impossible, that is, a few degrees change should be expected. There are a lot of factors that could cause the discrepancy in the heading angle, and lateral displacement for that matter.

- Even though points tracked pass a certain threshold of accuracy, still some points might not have been tracked accurately enough.
- Sometimes the consecutive images may look very similar, making it hard for the algorithm to detect change between those image frames. Results could be that no good distinguishable features be found and tracked or if found, they could be tracked incorrectly as everything looks the same.
- The default setting of angles (Beta) to zero in the case where the number of features found are lesser than the threshold (as mentioned earlier) could deviate from the angle while averaging.

## **5. Conclusion**

In this chapter, we summarize the results from our attempt to develop a prototype of lane departure warning system and mention the recommendations for future work on this research.

### **5.1 Summary**

The main objective of this study was to investigate and develop a lane departure warning system by determining the lateral characteristics of the vehicle. It is an innovative approach in the area of lane departure warning systems because this approach does not rely completely on the knowledge of lane markers or vehicle's GPS position.

The main goal of the project was to determine the lateral position of the vehicle from the image sequences captured along the way. First, an image is captured through a camera, which is then transformed to a top-down view image through homography (matrix) which is pre-calculated and incorporated in the algorithm. Secondly, the image is processed to find suitable features, referred to as corners, to be tracked in the following image. When another image is captured, the algorithm tries to determine the locations in the current image of the corners found in the preceding image. Once the locations have been determined with sufficient threshold, an angle is calculated between each tracked corners. From these individual angles, a heading angle is then determined, which indicates the direction the vehicle is moving. From the heading angle, the vehicle's lateral position is computed.

## 5.1 Issues and Recommendations

The design prototype of the lane departure warning system was tested through experiments performed at different locations in the Duluth area. Analyzing the data, we found that the accuracy seem satisfactory, but further improvements are needed in order to be used in the real world.

Below are some reasons that may have caused skewness in data:

1. In digital image processing, noise is one of the biggest factors. In many methods only Gaussian distributed noise is taken into account. Other noise can be addressed and compensated for if the camera parameters could be obtained. One way to do that is through the camera calibration.
2. Shadows on roads are also a difficult issue. The intensity difference between shadowed and non-shadowed part could be misleading and be picked up as good tracking point, and, thus, lead to tracking error.
3. Since the vehicle moves at a relatively higher speed, the assumptions used in the optical flow method may not be valid. If this is the case, then we need to increase the sampling rate.
4. Resetting of lateral position is required after certain time intervals so that the data doesn't deviate much should any frame tracking contains significant error. This resetting is possible through the lane detection algorithm discussed in Section 2.1.
5. The primary focus of this research was on straight roads to check the working of the algorithm. Addressing road curvatures are definitely the future work of this research.

## 6. References

- [1] SCr 001 Centerline and Outside Shoulder Rumblestrip. (2012).California Department of Transportation. Available: <http://www.dot.ca.gov/dist05/traffic/rumblestripppt.pdf>. Last accessed 1st May 2013.
- [2] A. Amditis, M. Bimpas, G. Thomaidis, M. Tsogan, M. Netto, S. Mammar, A. Beutner, N. Mohler, T. Wirthgen, S. Zipser, A. Etemad, M. Da Lio, and R. Cicilloni,"A Situation-Adaptive Lane-Keeping Support System: Overview of the SAFELANE Approach" in IEEE Transactions on Intelligent Transportation System, vol. 11, pp. 617-629, (2010)
- [3] I. Han.(2010, Feb). Safe driving and parking with Cadenza!. Available: <http://kia-buzz.com/safe-driving-and-parking-with-cadenza/>. Last accessed 1st May 2013
- [4] CITREON TECHNOLOGY. (2012). Available: <http://www.citroen.com.au/home/#/citroen-universe/technology/>. Last accessed 1st May 2013.
- [5] Valeo signs exclusive agreement with Iteris for lane departure warning systems for cars. (2004). Available: <http://www.theautochannel.com/news/2003/10/16/170760.html>. Last accessed 1st May 2013.

- [6] CORPORATE OVERVIEW. (2010). Available:  
[http://www.iteris.com/upload/datasheets/CorporateBrochure\\_FINAL\\_web.pdf](http://www.iteris.com/upload/datasheets/CorporateBrochure_FINAL_web.pdf). Last accessed 1st May 2013.
- [7] A. Shaout, "Advanced Driver Assistance Systems - Past, present and future," 7th International Computer Engineering Conference (ICENCO), pp. 72-82, (2011)
- [8] H. Godthelp, P. Milgram, and G. Blaauw, "The development of a time-related measure to describe driver strategy," Human Factors, vol. 26, no. 3, pp.257–268, (1984)
- [9] D. Pomerleau, "RALPH: Rapidly Adaptive Lateral Position Handler," Proceedings of IEEE Symposium on Intelligent Vehicles, Detroit, MI, (1995)
- [10] D. LeBlanc, P. Johnson, P. Venhovens, G. Gerber, R. DeSonia, R. Ervin, C. Lin, A. Ulsoy, T. Pilutti, "CAPC: A Road Departure Prevention System," IEEE Control Systems Magazine, vol. 16, pp. 61-71, 1996
- [11] T. Teshima, H. Saito, S. Ozawa, K. Yamamoto, T. Ihara, "Vehicle Lateral Position Estimation Method Based on Matching of Top-View Images," Proceedings of The 18th International Conference on Pattern Recognition, vol. 4, pp. 626-629, 2006
- [12] C. R. Jung, C. R. Kelber, "A Lane Departure Warning System Using Lateral Offset with Uncalibrated Camera," Proceedings of the 8th IEEE International Conference on Intelligent Transportation Systems, pp. 102-107, 2005

- [13] J. Sousanis.(2011).World Vehicle Population Tops 1 Billion Units. Available: [http://wardsauto.com/ar/world\\_vehicle\\_population\\_110815](http://wardsauto.com/ar/world_vehicle_population_110815). Last accessed 1st May 2013.
- [14] Most of Us Still Drive to Work - Alone. (2007). Available: [http://www.census.gov/newsroom/releases/archives/american\\_community\\_survey\\_acs/cb07-cn06.html](http://www.census.gov/newsroom/releases/archives/american_community_survey_acs/cb07-cn06.html). Last accessed 1st May 2013.
- [15] M. Peden, R. Scurfield, D. Sleet, D. Mohan, A. A. Hyder, E. Jarawan, and C. Mathers, "World report on road traffic injury prevention," World Health Organization, Geneva, (2004)
- [16] Car Accident Statistics. (2012). Available: <http://www.lawcore.com/car-accident/statistics.html>. Last accessed 1st May 2013.
- [17] J. S. Jermakian, "Crash avoidance potential of four passenger vehicle technologies." Accident Analysis & Prevention, vol. 43, no. 3, pp. 732-740, (2011)
- [18] C. Visvikis, T. L. Smith, M. Pitcher, and R. Smith, "Study on lane departure warning and lane change assistant systems," Transport Research Laboratory, (2008)
- [19] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, second edition, (2004)
- [20] ] G. Bradski and A. Kaebler. Learning OpenCV Computer Vision with the OpenCV Library. Cambridge, MA, (2008).

- [21] Gaussian Filtering. (2005). Electronics and Computer Science, University of Southampton. Computer Vision Demonstration Website. Available: [http://users.ecs.soton.ac.uk/msn/book/new\\_demo/gaussian/](http://users.ecs.soton.ac.uk/msn/book/new_demo/gaussian/) Last Accessed 14th May 2013
- [22] M. Aly, "Real time Detection of Lane Markers in Urban Street" Proceedings of IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 2008
- [23] Hough transform. (2012). Available: [http://en.wikipedia.org/wiki/Hough\\_transform](http://en.wikipedia.org/wiki/Hough_transform). Last accessed 1st May 2013.
- [24] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. (2003). Hough Transform. Available: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>. Last accessed 1st May 2013.
- [25] J. Shi, C. Tomasi, "Good Features to Track", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994
- [26] C. Tomasi, L. Kanade, "Detection and Tracking of Point Features", Technical Report CMU-CS-91-132, (1991)
- [27] C. Harris, M. Stephen, "A Combined Corner and Edge Detector, " Proceedings of The Fourth Alvey Vision Conference, pp. 147-151, 1988
- [28] G. Bebis. (2013, Jan). Interest Point Detectors. Available: [www.cse.unr.edu/~bebis/CS491Y/Lectures/InterestPointsDetection.ppt](http://www.cse.unr.edu/~bebis/CS491Y/Lectures/InterestPointsDetection.ppt)



[29] Y. Wu, "Optical Flow and Motion Analysis," Electrical Engineering and Computer Science Northwestern University, Evanston, IL

[30] J. Y. Bouquet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm", (2000)

## 7. Appendix

### 7.1 Result of the test showing the points tracked and angle between them for image frames 1 to 10

Frame Comparison code (Z)	Corner Ax	Corner Ay	Corner Bx	Corner By	Angle between A and B(Alpha)	No. of total Points in frame (Z-1)	No. of points found in frame Z
1						292	0
2	537.842	152.298	533.557	154.827	59.4527		
2	566.798	39.0564	553.675	54.804	39.8056		
2	382	114	398.574	150.165	-24.6213		
2	374	115	358.128	150.808	23.9051		
2	361.668	196.924	364.1	201.167	-29.8192		
2	296.807	132.631	305.813	149.351	-28.305		
2	366	114	347.477	149.769	27.3773		
2	408.51	140.383	406.808	144.445	22.7336		
2	337	113	328.77	147.927	13.2595		
2	298	114	304.792	149.27	-10.9001		
2	376	115	365.808	151.066	15.7793		
2	289	115	290.922	150.128	-3.13171		
2	360.917	194.841	357.705	198.249	43.304		
2	549	183	546.881	183.839	68.4086		
2	566.794	39.0647	553.683	54.7712	39.8545		
2	135	121	81.7853	135.047	75.2126		
2	606	167	602.733	169.033	58.1093		
2	557	130	549.707	165.905	11.4822		
2	300	115	299.454	150.105	0.890864		
2	324	113	323.084	147.696	1.51195		
2	295	114	295.106	148.997	-0.173468		
2	291	114	292.372	148.909	-2.25003		
2	258.145	206.626	260.506	206.722	-87.6632		
2	561.195	71.6656	556.204	72.8728	76.403		
2	333	113	337.301	147.45	-7.11655		

2	614.955	104.387	610.529	107.642	53.6709		
2	407.36	198.495	406.054	201.796	21.595		
2	364	114	343.273	149.246	30.4589		
2	269.555	191.733	271.486	196.236	-23.2087		
						292	29
3	406.222	203.421	405.898	205.61	8.41081		
3	318	147	320.175	174.667	-4.49576		
3	577.67	74.5839	550.239	101.613	45.4228		
3	406.609	96.862	409.38	97.7225	-72.7515		
3	221.704	207.025	222.575	210.235	-15.1736		
						242	5
4	563.897	189.254	561.071	190.113	73.0924		
4	560.078	130.196	562.185	132.291	-45.1748		
4	574.704	132.097	572.486	135.418	33.7393		
4	574.678	132.103	572.411	135.436	34.219		
4	275.42	204.585	273.608	207.099	35.7883		
4	328.461	185.626	328.177	190.167	3.57675		
4	404.778	209.882	405.413	215.605	-6.33333		
4	369.269	178.846	364.678	181.687	58.2509		
4	328.483	185.618	328.198	190.17	3.58698		
4	576.145	161.324	568.06	177.203	26.9837		
4	564.076	189.175	561.263	190.038	72.957		
4	297	188	294.271	188.535	78.9114		
4	562.265	160.903	558.97	161.219	84.5227		
4	560.079	130.196	562.182	132.295	-45.0599		
4	328.182	185.712	327.697	187.918	12.3943		
4	329.286	193.335	329.33	196.329	-0.85606		
4	325.784	195.758	326.678	197.778	-23.862		
4	600.236	147.425	548.532	189.977	50.5454		
4	313.488	194.469	315.318	196.716	-39.1587		
4	407.748	162.344	408.66	168.108	-8.99597		
4	327.551	154.568	330.471	156.69	-53.9876		
4	336.608	175.028	335.05	178.53	23.9953		
4	419	183	420.808	188.07	-19.6231		
						96	23
5	600.369	164.301	590.598	169.929	60.0568		
5	618.691	21.3194	620.387	24.0675	-31.6831		
5	310.227	200.861	315.613	214.031	-22.2436		
5	409.405	135.165	408.635	137.436	18.721		

5	400.05	165.494	399.96	170.679	0.997157		
5	406	191	405.31	194.123	12.4631		
5	405	205	405.119	207.646	-2.5678		
5	594.571	153.323	594.053	156.142	10.3995		
5	352.707	201.093	352.15	204.129	10.3985		
5	257.612	191.448	262.209	192.461	-77.5748		
5	376.191	188.744	375.184	192.351	15.598		
						116	11
6	592	173	578.257	179.511	64.6484		
6	344.665	183.456	344.719	186.723	-0.948344		
6	407	182	404.438	182.689	74.9432		
6	407	160	404.498	166.691	20.5043		
6	348.687	203.067	348.546	205.325	3.56623		
6	301.899	197.273	297.735	197.447	87.6036		
6	301.898	197.273	297.734	197.447	87.6016		
						80	7
7	541.729	139.937	538.324	142.085	57.7557		
7	541.729	139.937	538.324	142.085	57.7557		
7	619.774	11.6249	598.192	27.7642	53.2098		
7	620.413	40.3524	608.611	45.9355	64.683		
7	593.725	159.49	589.754	160.814	71.5566		
7	546.662	199.344	540.572	200.726	77.2172		
7	404.465	204.574	402.809	206.698	37.9363		
7	630.467	98.0915	607.104	107.906	67.2136		
7	546.026	200.821	539.898	200.885	89.3996		
7	385.029	203.43	382.179	205.421	55.0571		
7	389.235	207.867	389.292	212.509	-0.701023		
7	617	7	597.329	28.0895	43.0068		
7	635.551	129.881	607.882	134.5	80.523		
7	618	5	598.99	25.2621	43.1738		
7	621.627	85.2113	608.925	99.6925	41.2533		
7	404.472	204.583	402.816	206.69	38.172		
7	597.59	88.885	581.078	105.997	43.9762		
7	366.213	189.374	366.19	193.45	0.332035		
7	315.832	190.454	320.824	196.177	-41.0983		
7	638.677	55.2294	615.93	70.3085	56.4603		
7	386.034	204.383	383.574	206.219	53.2608		
7	550	187	538.83	194.121	57.4825		
7	348.119	194.362	342.633	195.684	76.449		

						97	23
8	582.38	115.911	573.698	126.243	40.0393		
8	541	206	536.749	209.091	53.9821		
8	542	184	536.01	192.394	35.5099		
8	540	184	538.468	188.341	19.4355		
8	585.727	151.422	579.275	153.913	68.8914		
8	535.669	200.2	540.024	200.674	-83.7806		
8	625.401	38.1711	596.517	67.4448	44.6163		
8	625.399	38.1714	596.51	67.4447	44.6215		
8	609	46	586.284	66.0073	48.6281		
8	535.621	200.201	539.66	200.639	-83.8027		
8	609.915	138.619	603.387	146.94	38.1192		
8	585.724	151.421	579.273	153.912	68.8896		
8	322.82	192.9	326.32	193.901	-74.0298		
8	340.868	200.997	337.871	204.473	40.7652		
8	597	8	596.908	29.8639	0.241518		
8	348.331	192.945	349.778	196.609	-21.5523		
8	545.901	175.141	549.041	177.262	-55.9615		
8	632.743	89.6293	624.042	116.506	17.9395		
8	600	25	586.093	36.8729	49.5105		
8	551.304	147.801	543.963	150.972	66.6371		
8	619	112	607.269	141.74	21.5266		
8	356.639	207.283	351.697	212.297	44.5835		
						129	22
9	588.909	70.3767	591.48	114.01	-3.37228		
9	604	82	593.164	83.5917	81.6435		
9	603.149	108.425	589.5	116.325	59.9365		
9	607	163	604.948	163.515	75.9257		
9	609	132	619.282	134.587	-75.8758		
9	574.52	67.438	566.397	78.9267	35.2628		
9	551.745	102.22	548.889	104.097	56.6926		
9	609.189	28.7929	615.699	45.9004	-20.8329		
9	402.683	194.069	402.194	196.087	13.6203		
9	335.316	173.655	338.884	173.906	-85.968		
						131	10
10	601.483	11.6543	603.888	12.6731	-67.0436		
10	592.884	53.6972	596.962	71.7839	-12.7059		
10	537.58	206.3	539.596	206.536	-83.3301		
10	601.484	11.6559	603.889	12.6742	-67.0536		

10	564.806	176.149	571.206	176.195	-89.5893		
10	592.881	53.6962	596.96	71.7835	-12.7084		
10	602	25	602.946	28.9132	-13.5842		
10	606	25	606.346	29.2894	-4.60859		
10	633.703	81.2756	609.639	107.189	42.8803		
10	570.938	165.214	575.137	167.907	-57.3243		
10	552.636	151.466	550.208	152.787	61.4553		
10	399	206	399.639	210.393	-8.27104		
10	603	15	605.606	15.3023	-83.3828		
10	389.416	207.231	389.055	211.174	5.23983		
10	389.406	207.216	389.048	211.398	4.89405		
10	604	164	597.811	165.477	76.5762		
10	559.25	147.485	549.181	153.49	59.1897		
10	378.084	208.026	379.193	211.583	-17.3108		
10	337.452	202.625	343.861	203.229	-84.6178		
10	587	59	591.646	67.7544	-27.9536		
10	239.71	173.704	237.505	173.735	89.1893		
						140	21