Estimation of Vehicle's Lateral Position via the Lucas-Kanade Optical Flow Method

Final Report

Prepared by:

Jiann-Shiou Yang

Department of Electrical Engineering
University of Minnesota Duluth

Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

CTS 12-31

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
CTS 12-31			
4. Title and Subtitle Estimation of Vehicle's Lateral Position via the Lucas-Kanade Optical Flow Method		5. Report Date September 2012 6.	
7. Author(s) Jiann-Shiou Yang		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Electrical Engineering University of Minnesota Duluth 271 Marshall W. Alworth Hall 1023 University Drive Duluth, MN 55812		10. Project/Task/Work Unit No. CTS Project #2011014 11. Contract (C) or Grant (G) No.	
12. Sponsoring Organization Name and Address Intelligent Transportation Systems Institute Center for Transportation Studies University of Minnesota 200 Transportation and Safety Building 511 Washington Ave. SE Minneapolis, Minnesota 55455		13. Type of Report and Period Covered Final Report 14. Sponsoring Agency Code	
15. Supplementary Notes http://www.its.umn.edu/Publications/ResearchReports/			

16. Abstract (Limit: 250 words)

The use of rumble strips on roads has proven to be an effective means of providing drivers lane departure warning (LDW). However, rumble strips require an infrastructure and do not exist on a majority of roadways. Furthermore, rumble strips present a difficult issue of where to establish the rumble-strip distance threshold. To develop an effective virtual rumble-strip LDW system where the rumble-strip threshold is allowed to vary according to the risk of the vehicle departing the road, it is essential to know the vehicle's lateral characteristics; in particular, the vehicle's lateral position and speed. In this report, we use image processing via an in-vehicle camera to estimate the vehicle's lateral position and speed. The lateral position is estimated by determining the vehicle's heading angle via a homography and the Lucas-Kanade optical flow techniques; while the lateral speed is determined via the heading angle and the vehicle's On Board Diagnostic (OBD)-II forward speed data access. The detail of our approach is presented in this report together with our findings. Our approach will only need the minimal set of information to characterize the vehicle lateral characteristics, and therefore, makes it more feasible in a vehicle application.

17. Document Analysis/Descriptors	18. Availability Statement		
Heading angle, On-board diagnostics (OBD), Homography, Histograms, Harris corner, Feature selection and tracking, Optical flow, Vehicle electronics, Histograms, Detection and identification	No restrictions. Document available from: National Technical Information Services, Alexandria, Virginia 22312		
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 45	22. Price

Estimation of Vehicle's Lateral Position via the Lucas-Kanade Optical Flow Method

Final Report

Prepared by:

Jiann-Shiou Yang

Department of Electrical Engineering
University of Minnesota Duluth

Northland Advanced Transportation Systems Research Laboratories
University of Minnesota Duluth

September 2012

Published by:

Intelligent Transportation Systems Institute
Center for Transportation Studies
University of Minnesota
200 Transportation and Safety Building
511 Washington Ave. S.E.
Minneapolis, Minnesota 55455

Acknowledgments

The study was funded by the Intelligent Transportation Systems (ITS) Institute, a program of the University of Minnesota's Center for Transportation Studies (CTS). Financial support was provided by the United States Department of Transportation's Research and Innovative Technologies Administration (RITA).

The project was also supported by the Northland Advanced Transportation Systems Research Laboratories (NATSRL), a cooperative research program of the Minnesota Department of Transportation, the ITS Institute, and the University of Minnesota Duluth College of Science and Engineering. The author would like to thank graduate student Rini Shrestha and undergraduate student Benjamin Nelson for their time and efforts to make this research work possible.

Table of Contents

List of Tables

List of Figures

Executive Summary

Roadway departure fatalities including run-off-the-road (ROR) and head-on fatalities are a serious problem in this country. Development of various techniques such as lane departure warning (LDW) systems can improve traffic safety significantly. An effective LDW system should be able to detect when the driver is in danger of departing the road and then trigger an alarm to warn the driver early enough to take corrective action. Before developing an LDW system, it is necessary to know the traveling vehicle's current status and provide that information to the system. Therefore, it is essential to know the vehicle's lateral characteristics. The main goal of this research is to estimate the vehicle's lateral position while traveling on the highway. In particular, the vehicle's lateral position and speed. Since the lateral position can be determined if we know the vehicle's heading angle, in this report we focus on determination of the vehicle's heading angle. An in-vehicle camera is used to capture the front-view images in real time. We then use the homography to remove the perspective effect from these front-view images. This transformation involves re-sampling the incoming image and re-mapping each pixel in the captured image toward a different position and producing a new two-dimensional array of pixels. The resulting image represents a top view (or the bird's eye view) of the road region in front of the vehicle, as it was observed from a significant height. The histogram equalization is then used by effectively spreading out the most frequent intensity values to enhance the global contrast of the images. This operation is particularly useful to deal with those images captured with backgrounds and foregrounds that are both bright or both dark. The Harris corners detection and tracking via the Lucas-Kanade optical flow technique to determine the vehicle's heading angle is then presented. The heading angle of the vehicle, and thus its lateral displacement, is calculated by evaluating the relationship between these tracked features. Instead of using the Harris's method, we use the Shi and Tomasi's method to determine good corners (features). In addition to determine the vehicle's heading angle, we access its real time forward speed via its On-Board Diagnostic (OBD-II) data port with a OBD-II interpreter and a CAN protocol controller and the physical bus in our test vehicle. The entire schematic diagram and signal flow to implement our approach is given. Finally, the road tests were conducted on Highway 61, Lavaque Road and Jean Duluth Road outside the Duluth city limit, and the results and findings are presented with recommendations for future improvements.

Chapter 1. Introduction

Roadway departure fatalities including run-off-the-road (ROR) and head-on fatalities are a serious problem in this country. According to the National Motor Vehicle Crash Causation Survey (NMVCCS) data [1], ROR crashes contribute to a large portion of fatalities and serious injuries to motor vehicle occupants and over 95% of the critical reasons for single-vehicle ROR crashes were driver-related. The category of the critical reasons attributed to driver performance errors (e.g., poor directional control, overcompensation) (27.7%), followed by driver decision errors (e.g., too fast for curve/conditions) (25.4%), critical non-performance errors such as sleeping, physical impairment (22.5%) and driver recognition errors (e.g., internal and external distractions) (19.8%). Moreover, statistics data indicated that 70% of ROR fatalities occur on rural highways and about 90% occur on two-lane roads [2]. Development of various techniques such as lane departure warning (LDW) systems can improve traffic safety significantly. An LDW system should be able to detect when the driver is in danger of departing the road and then trigger an alarm to warn the driver early enough to take corrective action. The LDW systems are mainly associated to lane departure warning system to detect lane markings and use this information to determine the vehicle's position on the road. The simplest LDW systems make no assumptions other than the built-in threshold distance, provides no look-ahead or prediction and requires no vehicle state information. One of the alternatives for LDW systems is to use the "time-to-lane-crossing" (TLC), which is a measure of the time remaining before a vehicle on a given trajectory will departure the road [3]. TLC was first proposed as a metric to evaluate human factors issues in vehicle driving [4], where the TLC value represented the time predicted for a potential lane edge crossing. In [4], Hayward calculated TLC using simple particle dynamics where the vehicle lateral acceleration and velocity are used. Pomerleau calculated TLC by using a preview driver model and his approach includes a computer-vision system to estimate the vehicle's position in the lane and provides an estimate of the instantaneous road curvature [5]. In another approach, LeBlanc *et. al.* implemented a TLC algorithm using a dynamic estimate of the effective steering gain to predict the vehicle path [6]. The vehicle path is compared to a computer-vision generated image of the upcoming road to predict potential lane crossings. In [7], Jung and Kelber proposed a lane departure warning system using lateral offset with a uncalibrated camera. Their method uses a linear-parabolic model to determine the lateral offset of the vehicle with respect to the center of the lane without needing information about camera parameters. Other forward looking, camera-based systems that use algorithms to interpret video images to estimate vehicle state and roadway alignment include AutoVue (Iteris's device), the Infiniti Lane Departure Prevention (LDP) system, the Volvo's and BMW's LDW systems, the Lexus LS 460 multi-mode Lane Keeping Assistant (LKA) system, etc. AutoVue is already being offered as a factory-installed option on Freightliner's Century and Argosy trucks. Automakers offering a lane departure warning system on some of their models include Audi, Infiniti, BMW, Cadillac, Volvo and Buick. The California PATH program has demonstrated the use of in-road magnetic markers and magnetometer-equipped vehicles for lane tracking [8].

In addition to the popular TLC method for LDW systems, the most common approach to prevent single vehicle lane departure is the use of rumble strips [9, 10] on road shoulders. Actually, the use of rumble strips on roads has proven to be an effective means of providing drivers departure warning [11]. But rumble strips require an infrastructure and are not available on all roadways. Furthermore, rumble strips presents a difficult issue of where to establish the rumble-strip

distance threshold (see the headline news: “Rumble Strips Unpopular in Other Minnesota Counties, Too”, “Roadsides Rumbles Cause a Grumble”, *Duluth News Tribune*, September 12, 2010; “County Vows to Reduce Rumble”, *Duluth News Tribune*, September 29, 2010). Placed rumble strips too close to the road edge guarantees that no road departures will be missed; yet it increases the frequency of false alarms. Placement further from the road edge reduces false alarms, but decreases the effectiveness of the warning in terms of driver response. In general, TLC provides more warning time than roadside rumble strips because warnings are triggered when a driver is predicted to be in danger. However, this prediction can be wrong because its calculation takes only into account vehicle’s trajectory not driver’s actions, and therefore, the number of false alarms is generally higher than roadside rumble strips. This could reduce the effectiveness and also driver confidence when using such a system. However, rumble strips require infrastructure and do not exist on a majority of highways. Recently, Batavia proposed an extension to TLC and road side rumble strips idea by allowing the driver to drift beyond the lane boundary by adding a virtual lane boundary adjusted by a learning phase [11]. The comparison of LDW systems on real driving conditions was also reported in the literature (e.g., [12, 13]).

The main goal of this research is to develop an innovative LDW system without knowledge of lane makers and vehicle’s GPS location. However, before developing such a system, it is essential that we know the vehicle’s lateral characteristics; in particular, the vehicle’s lateral position and speed. Therefore, our study mainly focuses on determination of the vehicle’s heading angle, which in turn is used to determine the vehicle’s lateral position and speed and thus, the vehicle’s lateral characteristics. We use the homography to remove the perspective effect from the front-view images captured by an in-vehicle camera to convert them to top-view (or the bird's eye view) images. The resulting image represents a top view of the road region in front of the vehicle, as it was observed from a significant height. We then use feature selection and tracking from these top-view images, via the the Lucas-Kanade optical flow technique [14, 15], to determine the vehicle’s heading angle. The heading angle of the vehicle and thus the lateral displacement is calculated by evaluating the relationship between these tracked features. Using this lateral position estimation as an input variable, in our Phase II we will develop a rule-based fuzzy logic decision making operation for lane departure warning.

This report is organized as follows. Chapter 2 describes the vehicle’s lateral characteristics and the heading angle determination. The homograpgy and histogram equalization of image processing are discussed. The schematic diagram and implementation flow chart to determine the vehicle’s heading angle are then followed. In Chapter 3, the Lucas-Kanade optical flow method including the Region of Interest (ROI), the Harris corners detection and tracking is introduced in some detail. Chapter 4 presents the road test results and findings. Finally, Chapter 5 gives the conclusion.

Chapter 2. Vehicle's Lateral Characteristics

In order to develop an effective LDW system, we need to know the vehicle's lateral characteristics. That is, the lateral position and its tendency in lateral movement/momentum. To identify the lateral position of the vehicle, it is essential that we determine the vehicle's heading angle and its lateral speed. The vehicle's lateral speed can be determined once we know its heading angle (to be discussed later) and forward speed. Let *y*-axis be the vehicle's longitudinal axis, *x*-axis be the perpendicular lateral axis, u be the vehicle's forward speed and θ its heading angle (i.e., the angle between the vehicle moving direction and its longitudinal y-axis as shown in Figure 2.1).

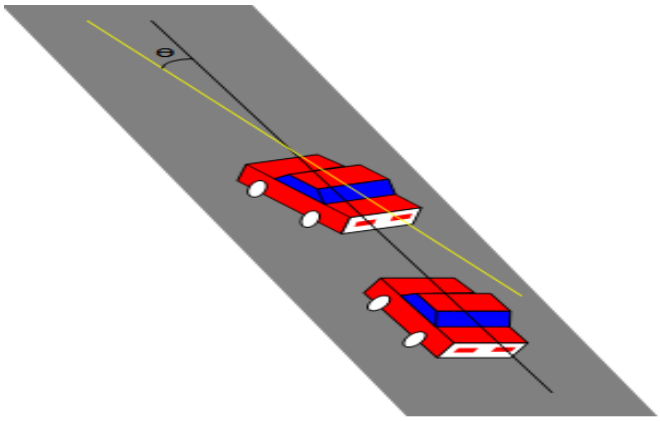

Figure 2.1 Vehicle's heading angle.

Then, given the vehicle's current position (x, y) it is easy to see that the vehicle's lateral speed can be expressed as

$$\dot{x} = u \sin(\theta)$$

where $\dot{x}$ means the time derivative of *x*. For small θ, we have

$$\dot{x} \approx u\,\theta$$

and u can be further replaced by

$$u \approx \bar{u}$$

where $\bar{u}$ is the vehicle's average forward speed. Therefore, the vehicle's lateral speed is determined by

$$\dot{x} = \bar{u}\,\theta$$

Note that the assumption of small heading angle and constant forward speed are justifiable for highway driving conditions and the vehicle's forward speed can be measured through the vehicle's On-Board Diagnostic (OBD)-II. In Section 2.1, we discuss about how we can access the vehicle's forward speed in real time.

2.1 Vehicle's Forward Speed

On-Board Diagnostic systems can be found in most cars and light truck on the road today. During the 1970' and early 1980' manufactures started using electronic means to control engine functions and diagnose engine problems. Through the years on-board diagnostic systems have become more sophisticated. OBD-II, a new standard introduced in the mid-'90, provides almost complete engine control and also monitors parts of the chassis, body and accessory devices, as well as the diagnostic control network of the vehicle. According to the information from [16], all light duty vehicles (e.g. less than 8,500 pounds) sold in North America since 1996, as well as medium duty vehicles (e.g. 8,500-14,000 pounds) beginning in 2005, and heavy duty vehicles (e.g. greater than 14,000 pounds) beginning in 2010, are required to support OBD-II diagnostics, using a standardized data link connector, and a subset of the SAE J/1979 defined Parameter IDs (PIDs) (or SAE J/1939 as applicable for medium/heavy duty vehicles), primarily for state mandated emissions inspections. OBD-II PIDs are codes used to request data from a vehicle, used as a diagnostic tool. SAE standard J/1979 defines many PIDs, but manufacturers also define many more PIDs specific to their vehicles. Typically, an automotive technician will use PIDs with a scan tool connected to the vehicle's OBD-II connector, i.e., (a) the technician enters the PID, (b) the scan tool sends it to the vehicle's bus (CAN, VPW, PWM, ISO, KWP. After 2008, CAN only), (c) a device on the bus recognizes the PID as one it is responsible for, and reports the value for that PID to the bus, and (d) the scan tool reads the response, and displays it to the technician. Since it is important to send the forward speed data to the computer, we need to access the vehicle's OBD-II. The schematic diagram shown in Figure 2.2 is used to measure the vehicle's forward speed and thus, we can determine its lateral speed $\dot{x}$ via $\dot{x} \approx \bar{u}\,\theta$ (where the heading angle θ will be explained later).

As mentioned above almost all the vehicles manufactured in the US today are required, by law, to provide an interface for the connection of diagnostic test equipment. The data transfer on these interfaces follow several standards, but none of them are directly usable by PCs or smart devices. In Figure 2.2, the ELM 327 chip, a multi-function OBD-II interpreter, is used to act as a bridge between OBD ports and a standard RS232 interface for systems that use the ISO 15765-4 CAN, SAE J1850 PWM, SAE J1850 VPW, ISO 9142-2, ISO 14230-4 and SAE J1939 protocols.

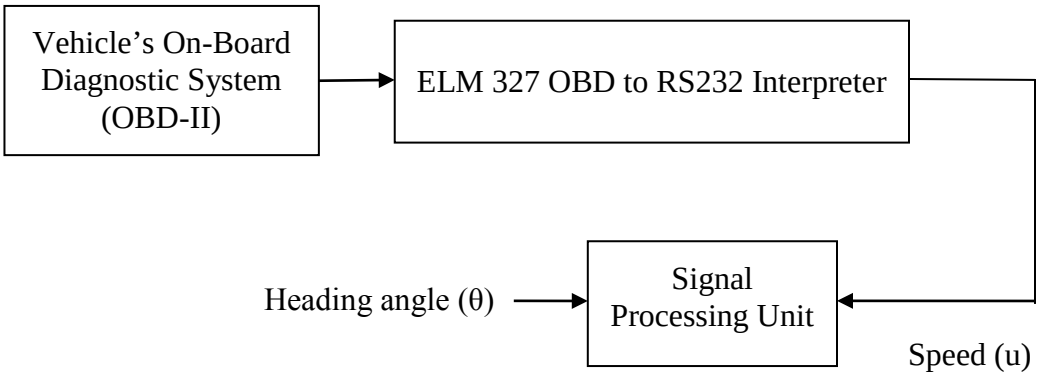

Figure 2.2 Measurement of the vehicle's forward speed via OBD-II.

2.2 The Homography

To determine the vehicle's heading angle, we used the image-based technique via an in-vehicle camera. Several existing methods such as the optical flow based method [14, 15], the lane marker based method [17], the focus of expansion (FOE) method [18] can be found in the literature. In our study, in order to determine the vehicle's heading angle more accurately, we converted the front-view images captured by the camera to their corresponding top-view images (i.e., the bird's eye view images) via a homography (see Figure 2.3). In the field of computer vision, any two images of the same planar surface in space are related by a homography. There are many practical applications such as image registration, rectification, computation of camera motion (rotation, translation) between two images. In this study, we mainly deal with the accumulated lateral displacement of an in-vehicle camera in order to find the vehicle's lateral position, Once camera rotation and translation have been extracted from an estimated homography matrix, this information will be later used in our feature detection and tracking of the consecutive top-view images via the Lucas-Kanade optical flow method [14] to determine the vehicle's heading angle. The optical flow method will be explained in the next chapter.

Our image processing and conversion are conducted via an in-vehicle camera as shown in Figure 2.3. It is known that camera lens can cause distortion and distortion causes fisheye effect on the images (see Figure 2.4 (a)); therefore, the camera needs to be calibrated to remove such a distortion (see Figure 2.4 (b)). This step is important and is also necessary to produce a "good" quality of front-view images and thus, provide a better feature selection/detection and tracking in the image processing. We used the Open Source Computer Vision Library (OpenCV) to find camera intrinsic and distortion properties (parameters). The software OpenCV is a library of programming functions mainly aimed at real time computer vision, developed by Intel, which we will used later in our image processing including the homography and optical flow method. Note that the conversion to top-view image in Figure 2.3 involves removing the perspective effect in the front-view image. That is, re-sample the incoming image and re-mapping each pixel in the captured image toward a different position and producing a new two-dimensional array of pixels. The resulting image represents a top view (or the bird's eye view) of the road region in front of the vehicle, as it was observed from a significant height.

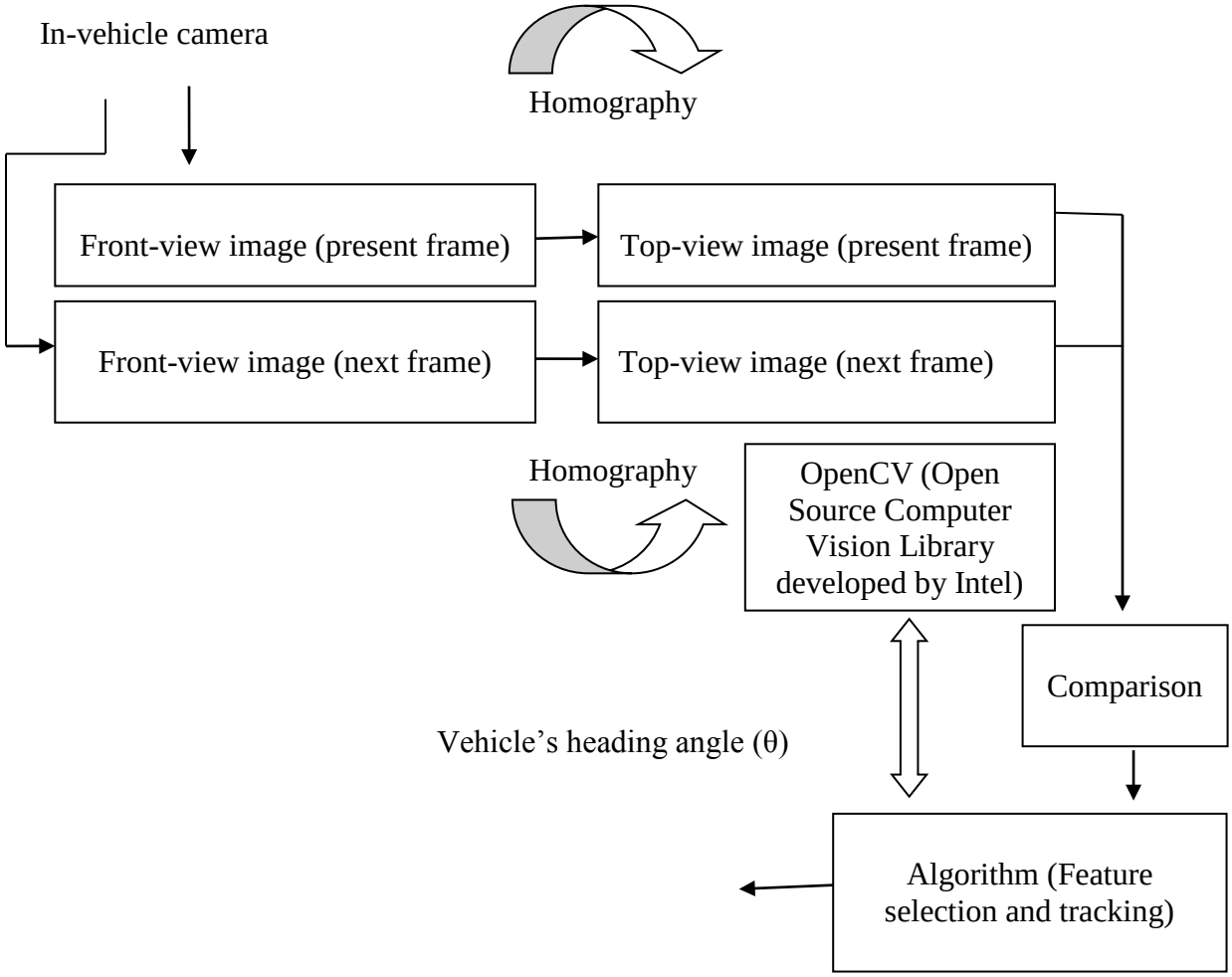

Figure 2.3 Image-based sensing to determine the vehicle's heading angle

Figure 2.4 Distorted front-view image (left) and distortion removed image (right).

The camera used in this study conducts real-time data streaming and its maximum rate is 30 frames per second. Figure 2.5 shows an example of the two consecutive front-view images taken by the camera at frame i and frame i+1 as the test vehicle is moving forward.

Figure 2.5 Two consecutive front-view images taken by the camera at time i and i+1.

After using the camera distortion file and by removing the perspective effect via the homograpgy, the images in Figure 2.5 are converted to their corresponding top-view images as shown in Figure 2.6.

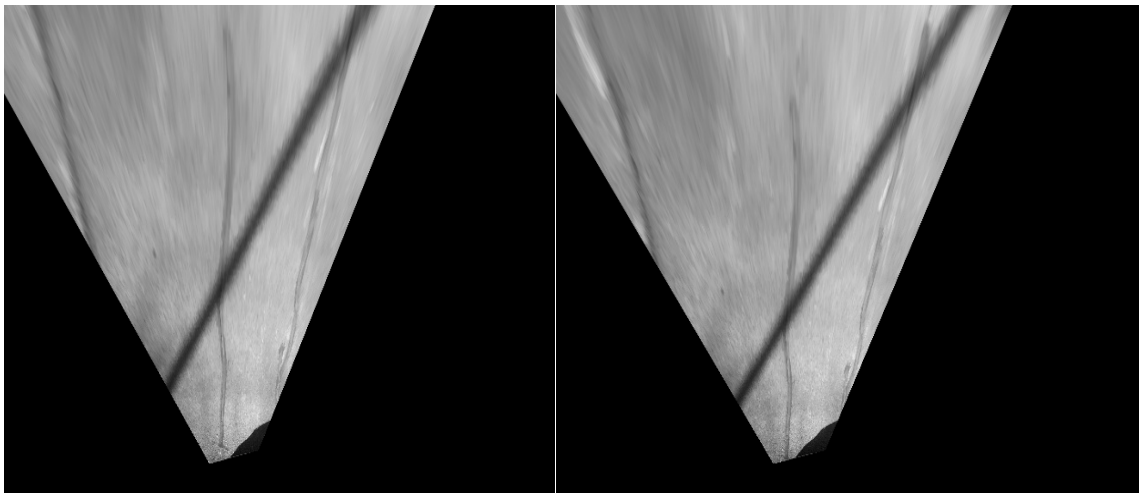

Figure 2.6 Top-view images at time i and i+1 after homography.

However, we found that if a non-wide-angle camera is used then the nonlinear distortion near the boundaries of the images can be eliminated. Thus, the camera calibration together with a distortion file are not necessary. Therefore, in determining the vehicle's heading angle, we used a Microsoft H5D LifeCam Cinema CMOS image sensor to capture front-view images. Figure 2.7 shows a snapshot taken from this camera and Figure 2.8 shows its corresponding top-view image after the homography transformation.

Figure 2.7 A Snapshot of a front-view image using the Microsoft H5D camera.

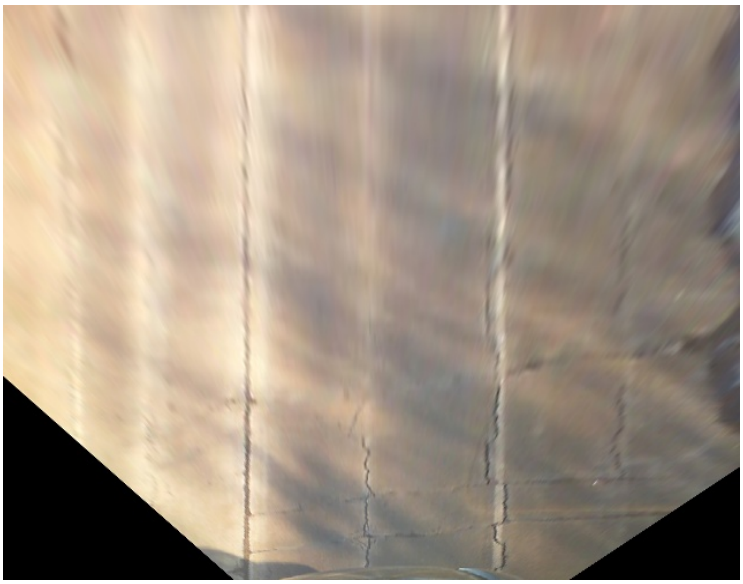

Figure 2.8 The front view image of Figure 2.7. after the homography.

2.3 Histogram Equalization

The contract of the images in a scene captured by the camera is very important to improve the quality and success rate of feature detection and tracking when using the optical flow techniques. Therefore, the images need to be preprocessed to enhance their contrast. Histogram equalization can increase the global contrast of images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast. The mathematics behind histogram equalization involves mapping one distribution (the given histogram of intensity values) to another distribution (a wider and ideally uniform distribution of intensity values). Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. This method is useful in images with backgrounds and foregrounds that are both bright or both dark. The brightness distribution is equalized thereby increasing the contrast of the image. Therefore, we applied histogram equalization to the front-view images so that there is sufficient texture variations to detect good features later. The effect of applying the contrast enhancement technique to Figure 2.7 can be seen in Figure 2.9, where Figure 2.9 (a) shows the original image (i.e., Figure 2.7), while Figure 2.9 (b) shows the image after contrast enhancement. Two sample images after contrast enhancement at two consecutive time instants i and i+1 are also shown in Figure 2.10 for comparison purpose.

(a) (b)

Figure 2.9 Effect of histogram equalization (a) before and (b) after.

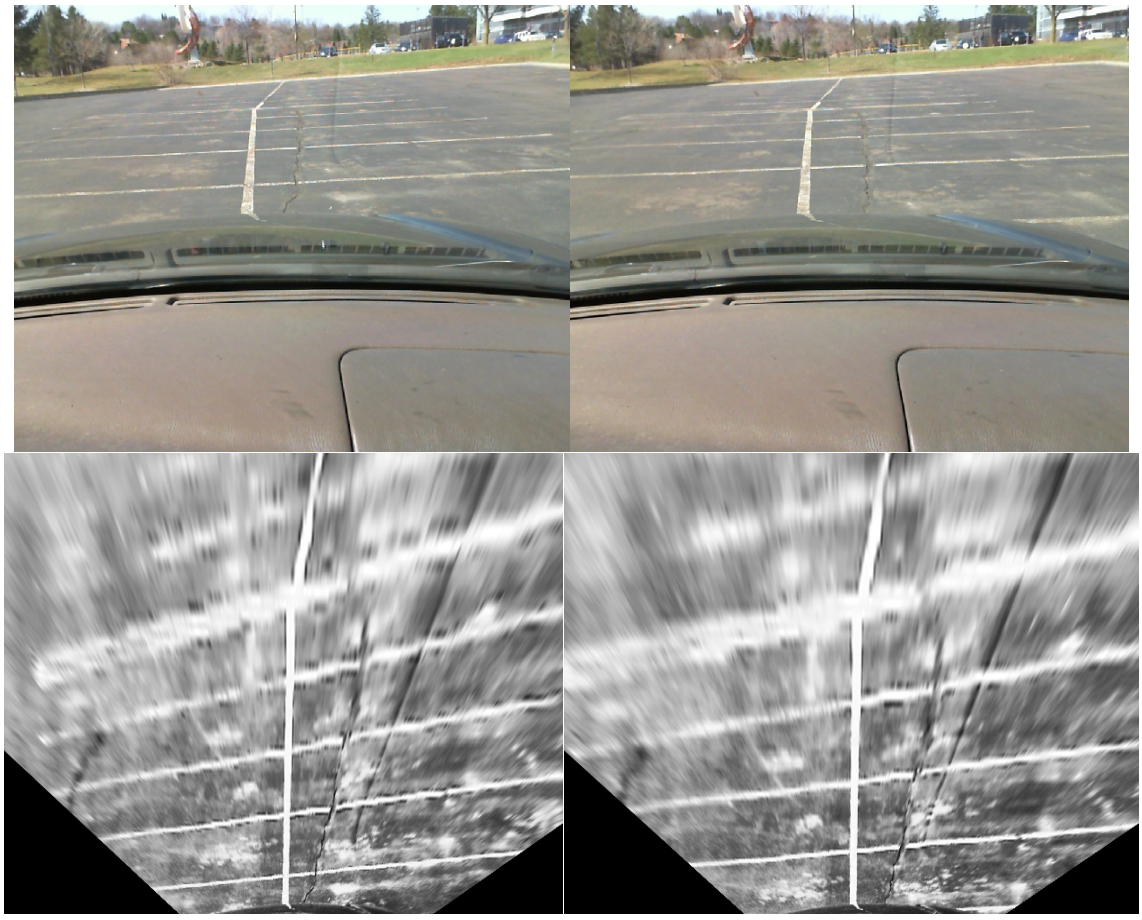

Figure 2.10 Two consecutive top-view images taken by the camera at time i and i+1 after histogram equalization.

2.4 Determination of the Vehicle's Lateral Position Determination

Based on the real time contrast enhanced top view images, we use the Lucas-Kanade optical flow technique [14, 15] to determine the vehicle's heading angle. The schematic diagram of the data flow and signal processing to identify the vehicle's lateral position is shown in Figure 2.11. Based on the vehicle's current speed and its heading angle, the algorithm calculates the lateral position relative to its initial position. Note that in Figure 2.11 the vehicle's relative velocity can be determined by the vehicle's forward speed multiplied by $\sin(\theta)$, and for small heading angle θ, $\sin(\theta)$ can be approximated by θ; therefore, the lateral speed can be determined by the forward speed multiplied by its heading angle.

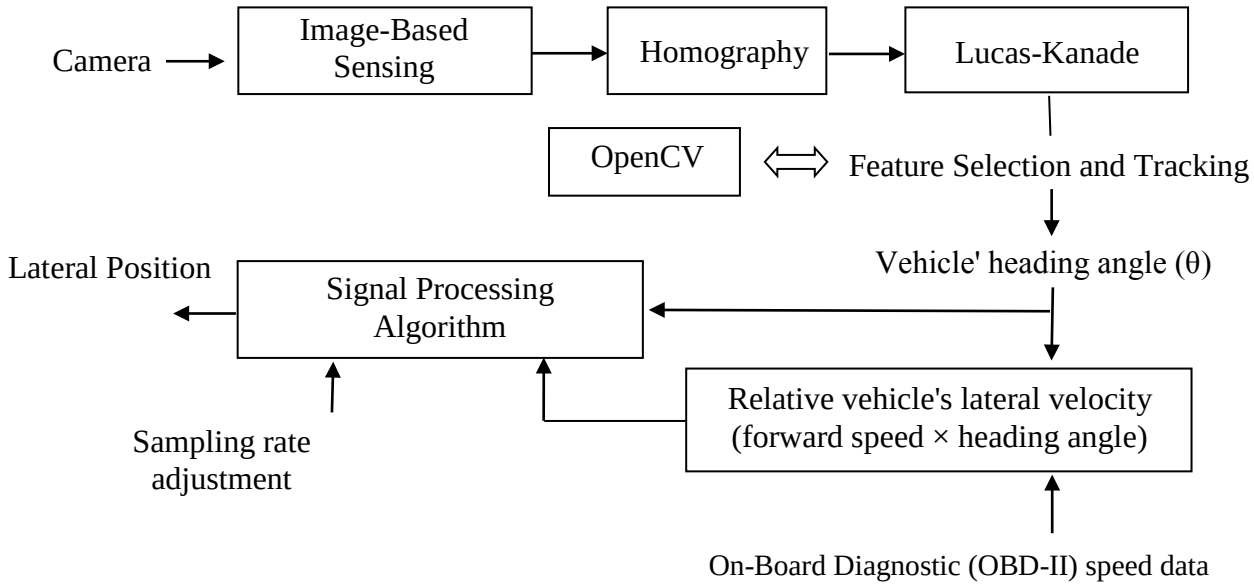

Figure 2.11 Schematic diagram showing the signal processing flow to determine the vehicle's lateral position.

The flow chart to implement Figure 2.11 is shown in Figure 2.12, where the optical flow technique will be discussed in Chapter 3. The LDW system will be designed and developed in such a way that it will be activated by the driver once the vehicle's initial lateral position has been set. The subsequent lateral position input will be determined by the signal processing algorithm in Figure 2.11. In other words, the driver starts driving the vehicle to the lane he is heading. Once the vehicle moves to the center of the lane (i.e., its "initial" lateral position), the driver then activates the LDW system and leaves the system on while traveling on the roadway. Instead of setting its default "initial" lateral position, another approach is to use any road edge detection methods (several techniques are currently available) to determine its initial position. However, this information will only be used once (i.e., to locate the vehicle's initial position). Of course, this later approach will not need the driver's visual maneuvering close to the center of the lane before activating the system. Note that by decreasing the sampling period, the system can deal with the road curvature changes, and the camera-based sensing in Figure 2.3 can also help to provide road curving information to the system. The vehicle "initial" lateral position will be reset as often as possible once additional surrounding information becomes available (that is, are lane markers available and can be seen by the camera? Is the road curving?). A mechanism to reset the vehicle's "initial" lateral position via the camera detection of the road curvature will be built into the operation algorithm in our Phase II study. Instead of using the RGB color space, the image data streaming in the hue-saturation-luminance (intensity) color space can be used to distinguish and separate the road (the vehicle is travelling) from its surrounding background and thus, we can deal with the curvature issue. A preliminary study on this shows good results. This will be integrated into the flow chart shown in Figure 2.12 in our Phase II study.

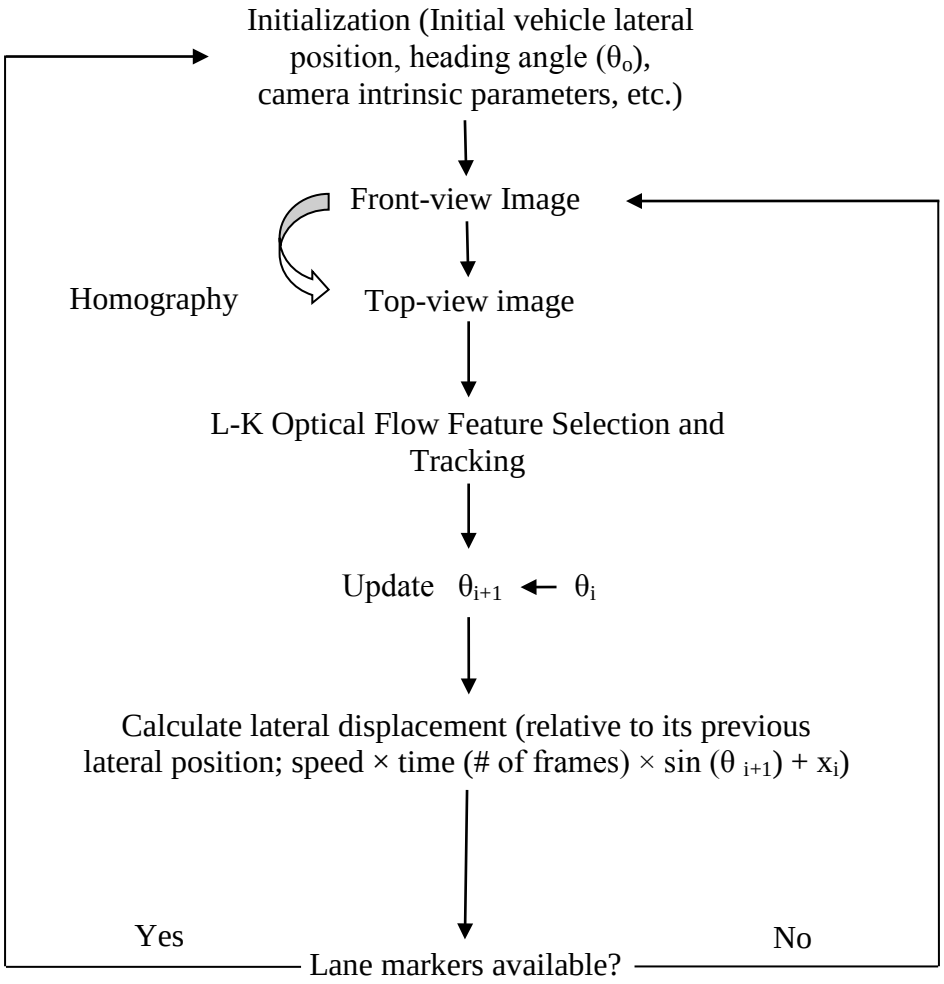

Figure 2.12 Implementation flow chart.

Chapter 3. Image Feature Selection and Tracking – The Lucas-Kanade Optical Flow Method

Optical flow is an approximation of the image motion based on local derivatives in a given sequence of images. It is the distribution of apparent velocities of movement of brightness patterns in an image. In a two-dimensional (2D) plane it specifies how much each image pixel moves between adjacent images. Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements. Consequently, optical flow can give important information about the spatial arrangement of the objects viewed and the rate of change of this arrangement. There are several methods available for determining optical flow including the phase correlation method [19], the block-based method [20], the differential methods [14, 21] and discrete optimization method [22], etc. The differential methods are based on partial derivatives of the image signal (e.g., the Lucas-Kanade method [14], the Horn-Schunck method [21]). In this study, we focus on the Lucas-Kanade optical flow method to determine the vehicle's heading angle via the top-view image processing. In the following, we give a brief concept introduction about this approach.

3.1 The Lucas-Kanade Optical Flow Method

The Lucas-Kanade (LK) algorithm is based on the following three assumptions: (a) brightness constancy, (b) temporal persistence (implies "small movements"), and (c) spatial coherence [14]. The first assumption means that a pixel from the image of an object in the scene does not change in appearance as it moves from frame to frame. That is, the brightness of a pixel does not change as it is tracked from frame to frame. The second assumption simply means that the image motion changes slowly in time, i.e., the motion is slow relative to the frame rate. The third assumption implies that neighboring points in a scene stay neighbors. In other words, the neighboring points belong to the same surface, have similar motion, and project to nearby points on the image plane.

The Lucas-Kanade optical flow method tries to calculate the motion between two image frames which are taken at times t and $t + \Delta t$ at every pixel position. This method is also called differential since it is based on the local Taylor series approximations of the image signal; that is, it uses partial derivatives with respect to the spatial and temporal coordinates. Let's consider a 2D case (3D or higher-order dimensional cases are similar). Let the brightness intensity $I(x, y, t)$ is the center pixel in a n × n neighborhood and moves by Δx, Δy in time Δt to $I(x+\Delta x, y+\Delta y, t+\Delta t)$. The brightness constancy assumption is the requirement that pixels in one tracked patch look the same over time. Therefore, we have the following expression between the two image frames

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{1}$$

Assuming the movement to be small, the image constraint at $I(x, y, t)$ with the Taylor series expansion becomes

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \ldots \tag{2}$$

From the assumption (a), we have

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \tag{3}$$

and, thus

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0 \tag{4}$$

which results in

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0 \tag{5}$$

where V_x, V_y are the x and y components of the velocity or optical flow of $I(x, y, t)$ and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$, rewritten as I_x, I_y and I_t respectively, are the derivatives of the image at (x, y, t) in the corresponding directions. That is,

$$I_x V_x + I_y V_y = -I_t \tag{6}$$

or

$$\nabla I^T \cdot \vec{V} = -I_t \tag{7}$$

where the spatial intensity gradient $\nabla I = [I_x \; I_y]^T$ (T means transpose) and $\vec{V} = [V_x \; V_y]$ (the image velocity or optical flow at pixel (x, y) at time t). This is the 2D motion constraint equation with two unknowns for any given pixel. Clearly, this equation can not be used to obtain a unique solution for the 2D motion at that point. However, if a local patch of pixels moves coherently, then we can easily solve for the motion of the central pixel by using the surrounding pixels to set up a system of equations. This spatial coherence is exactly the 3rd assumption proposed by the Lucas-Kanade optical flow method. That is, neighboring points in a scene belong to the same surface, have similar motion, and project to nearby points on the image plane (i.e., neighboring points stay neighbors). The Lucas-Kanade optical flow implemented a weighted least-squares (LS) fit of local motion constraint equation for $\vec{V}$ in each small spatial neighborhood Ω by minimizing

$$\sum_{x,y\in\Omega} W^2(x,y)[\nabla I(x,y,t)\bar{v} + I_t(x,y,t)]^2 \tag{8}$$

where $W(x, y)$ is a weighting function (typically chosen as 2D Gaussian) that gives more influence to constraints at the center of the neighborhood than those at the periphery. The solution to the above minimization reduces to

$$\bar{V} = [A^T W^2 A]^{-1} A^T W^2 b \tag{9}$$

where, for N pixels (i.e., for a n × n neighborhood N = n^2), $(x_i, y_i) \in \Omega$ at time *t* and

$$A = [\nabla I(x_1, y_1), \cdots, \nabla I(x_N, y_N)]$$

$$W = diag[W(x_1, y_1), \cdots, W(x_N, y_N)]$$

$$b = -[I_t(x_1, y_1), \cdots, I_t(x_N, y_N)]^T \tag{10}$$

where *diag* means the diagonal matrix. The solution to the motion constraint equation can be solved in closed form when $A^T W^2 A$ is nonsingular (i.e, when it is of rank 2) and

$$A^T W^2 A = \begin{pmatrix} \sum W^2(x,y) I_x^2(x,y) & \sum W^2(x,y) I_x(x,y) I_y(x,y) \\ \sum W^2(x,y) I_y(x,y) I_x(x,y) & \sum W^2(x,y) I_y^2(x,y) \end{pmatrix} \tag{11}$$

where all sums are taken over pixels (*x, y*) in the neighborhood Ω.

In our study, homographies are obtained by the computation of the rotation matrix according to the information of the calibrated camera, the vehicle's speed, and the angle of vehicle's direction. The movement obtained from each frame is accumulated to obtain the lateral position of the camera, and thus, the vehicle's lateral position from its initial position. A recent method to estimate the lateral position of the vehicle based on matching of top-view images from a sequence of moving camera images inside the vehicle has been reported by Teshima, *et al.* [23]. It is based on the so-called plane projective transform (homography) between the ground and the image plane of the present and next frame. The authors in [23] claimed that the effectiveness of their approach has been demonstrated by researchers from Mitsubishi Fuso Truck & Bus Corporation and Keio University in Japan, and their approach does not rely on extraction of features such as lines, rail, flow vectors or lane markers, but based on matching of warped top-view images between two consecutive frames. In this study, we used the camera-based image sensing. The concept of our approach is similar to the Teshima's method [23] but with some modifications. Instead of using a hypothesis of guessing the best candidate among several possible candidate heading angles with calculated "sum of absolute difference" (SAD) at each frame, we used the feature selection and tracking of the consecutive top-view images via the Lucas-Kanade optical flow method [14, 15] to determine the heading angle.

3.2 Region of Interest (ROI)

The feature selection/detection and tracking are based on the top-view images obtained. At each time frame, a set of "good" features (or "corners") are determined. Trackable (and reliable) features are points that contain enough information to be picked from the current frame to the next frame. They should have brightness constancy, sufficient texture variations, and do not deform much over time [15]. These "corners" can be found by calculating the eigenvalues of the image's 2nd moment matrix, and larger eigenvalues (above a preset threshold) imply good feature. To reduce real-time computation, a Region of Interest (ROI) instead of the whole region

in a top-view image is chosen. For instance, for the top-view image shown in Figure 2.8, we choose its ROI to be a rectangular box image, a 300 pixel × 165 pixel sample of the image, centered to be straight in front of the vehicle showing the width of the lane about 15 feet. This ROI image, as shown in Figure 3.1, starts right after the vehicle's hood on the bottom of Figure 2.8 and goes up to show about 10 feet in front of the vehicle. Notice the right and left lane markers in Figure 3.1.

Figure 3.1 The ROI image.

We want to find good features in the limited area instead of using the whole images. The ROI region is just wide enough to fit the width of a normal lane on a road or highway and we use the ROI to greatly speed up the computations, allowing the algorithm to run in real time. In addition, we don't need to worry about the possible distortion effects created by converting to the top-view images since these only arise around the edges of the original image.

3.3 Harris Corners Detection and Tracking

A point of our interest in an image is a point which has a well-defined position and can be robustly detected and tracked. This means that an interest point can be a corner but it can also be an isolated point with local intensity maximum, or minimum. Many trackable points are called "corners", and the words "corner" and "feature" are sometimes used interchangeable in literature. if we choose a point that is unique (i.e., a "good" feature) in a image frame then we have a pretty good chance of finding that point again in the next image frame. In practice, the point or feature we select should be unique, or nearly unique, and should be parameterizable in such a way that it can be compared to other points in another image.

The most commonly used definition of a corner was provided by Harris [24]. This definition relies on the matrix of the second-order derivatives $(\partial^2 x, \partial^2 y, \partial x \partial y)$ of the image intensities. That is, we take the second-order derivatives of images, taken at all points in the image, to form a new "second-derivative images" (or a new Hessian image). This terminology comes from the Hessian matrix around a point, which is defined as

$$\begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \tag{12}$$

The autocorrelation matrix of the second derivative images over a small window around each point is defined as

$$M(x,y) = \begin{pmatrix} \sum_{-k \le i,j \le k} w_{i,j} I_x^2(x+i, y+j) & \sum_{-k \le i,j \le k} w_{i,j} I_x(x+i, y+j) I_y(x+i, y+j) \\ \sum_{-k \le i,j \le k} w_{i,j} I_x(x+i, j+j) I_y(x+i, y+j) & \sum_{-k \le i,j \le k} w_{i,j} I_y^2(x+i, y+j) \end{pmatrix} \tag{13}$$

where $w_{i,j}$ is a weighting function (e.g., Gaussian weighting). The Harris corners are points in the image where the autocorrelation matrix of the second derivatives has two large eigenvalues. The advantage of considering only the eigenvalues of the autocorrelation matrix is that they are invariant to rotation, which is important because objects that we are tracking might rotate as well as move. However, instead of using the Harris's method, we use the Shi and Tomasi's method to determine good corners (features), which result as long as the smaller of the two eigenvalues is greater than a minimum threshold [25].

To explain the corner detection and tracking, let's consider the two consecutive top-view images in Figure 2.6 and their corresponding ROIs shown in Figure 3.2.

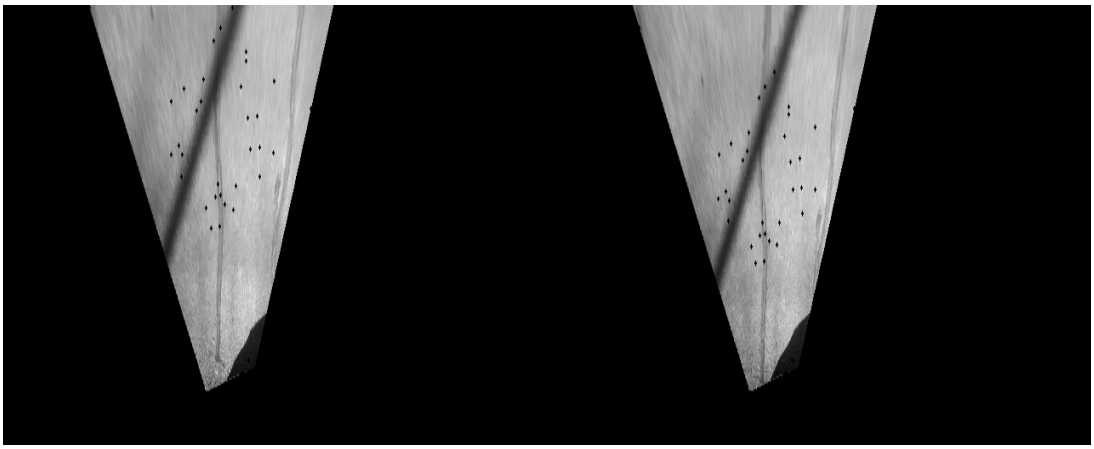

Figure 3.2 ROI at time i and i+1.

In Figure 3.2, the black dots at time i are the Harris corners to be tracked at frame i+1 and the dots identified at frame i+1 are to be tracked at frame i+2. We used a pyramidal approach of the Lucas-Kanade optical flow technique [4, 5] together with the OpenCV to detect the large motions which gave the feature displacement as shown in Figure 3.3. In this figure, the moving vectors (in red) show how the features selected moved from frame i and i+1.

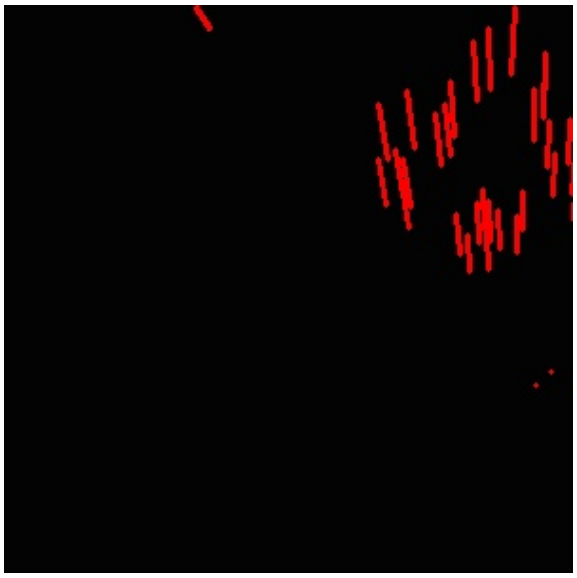

Figure 3.3 Feature displacement.

The OpenCV software package [15], a strong focus on real-time applications originally developed by Intel, is used to process the image conversion and transformation. The schematic diagram showing this image-based sensing approach is given in Figure 2.11.

3.4 Heading Angle

Base on the feature displacement, the coordinates of the features in two consecutive frames can then be determined (i.e., by superimposing two ROIs and comparing where these selected points moved). Note that only those points, forming a sunset of points selected from ROI, with increasing y-coordinate are tracked. Averaging the total angles associated with these selected features (with respect to the *x*-axis) gives us the vehicle's heading angle at that time instant.

Consider two consecutive top-view images at frame i and i+1. Let p be the number of points selected in the ROI at frame i, $x_{i+1,j}$ be the *x*-coordinate of the j^{th} point at frame i+1, $x_{i,,j}$ be the *x*-coordinate of the j^{th} point at frame i, $y_{i+1,j}$ the *y*-coordinate of the j^{th} point at frame i+1, and $y_{t,,i}$ be the *y*-coordinate of the j^{th} point at frame i. Also define $(\Delta x)_j = x_{i+1,\,j} - x_{i,j}$, $(\Delta y)_j = y_{i+1,\,j} - y_{i,j}$ and let n represent the number of points with $(\Delta y)_j < 0$. Then, If $(\Delta x)_j > 0$ let $\theta_j = -(90^o - \beta_j)$ (left turn); If $(\Delta x)_j < 0$ let $\theta_j = (90^o - \beta_j)$ (right turn); and if $(\Delta x)_j = 0$ then set $\theta_j = 0$. Note that β_j is defined

$$\beta_j = \frac{180^o \times \alpha_j}{\pi} \tag{14}$$

where

$$\alpha_j = \tan^{-1} \left| \frac{\Delta y_j}{\Delta x_j} \right| \tag{15}$$

and j= 1 to n. The heading angle is determine by a simple averaging as follows

$$\theta = \frac{1}{n}\sum_{j=1}^{n}\theta_j \tag{16}$$

The entire process and the concept were tested on the roads. Except for some outliers, overall speaking they showed good results. The data flow and signal processing to identify the vehicle's lateral position shown in Figure 2.11 is implemented. Based on the vehicle's current speed and its heading angle, the algorithm calculates the lateral position relative to its initial position. Note that in Figure 2.11 the vehicle's relative velocity can be determined by the vehicle's forward speed multiplied by sin(θ), and for small heading angle θ, sin(θ) can be approximated by θ; therefore, the lateral speed can be determined by the forward speed multiplied by its heading angle.

Chapter 4. Results and Findings

The vehicle's lateral position is determined by implementing the schematic diagram shown in Figure 2.11. We use the Open Source Computer Vision Library functions, known as OpenCV, to implement Figure 2.11 together with the iterative signal flow operation given in Figure 2.12. OpenCV is an open source computer vision library originally developed Intel. The library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Matlab, and other languages. OpenCV was designed for computational efficiency and with a strong focus on real time applications. Its library contains over 2,500 optimized algorithms that span many areas in vision, including medical imaging, user interface, camera calibration, stereo vision, and robotics. Therefore, we use the open source library functions OpenCV 2.3.1 to implement our operation algorithms. In this chapter, we discuss about the results and finding during our road tests.

4.1 Road Tests

We conducted road tests to check the accuracy of our operation algorithm in finding the heading angle of the vehicle, and thus its lateral displacement. The tests were performed in different areas in Duluth, Minnesota at 50 miles per hour (mph) and 65 mph speeds. Highway 61 was chosen as a test site with driving speed 65 mph while the tests at 50 mph were performed on segments of Martin Road, Lavaque Road, and Jean-Duluth Road. Figure 4.1 shows a sample screenshot of one of the many spreadsheets the algorithm generated (with the heading angle data shown in real time) while we conducted the road test on one segment of June Duluth Road lying north of Martin Road. Table 4.1 gives the results of this road test showing the points (Harris corners) tracked and the angle changes between the those tracked points.

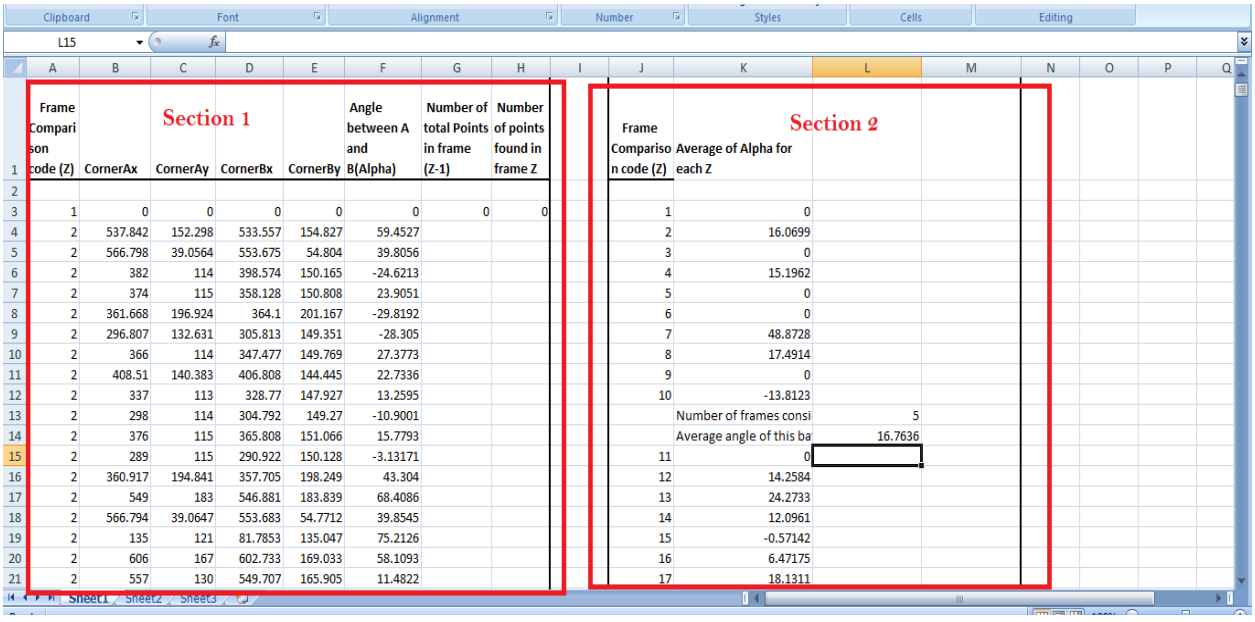

Figure 4.1 A snapshort of the data spreadsheet.

The test (i.e., Table 4.1) was conducted at 9:30 am on a sunny morning. Note that only a straight stretched portion of the road was considered for the analysis as the current algorithm does not account for curved roads. The handling of the road curvature issue will be reported in our Phase II study, where a lane detection algorithm using the Hough transform [26, 27], will be integrated. During part of this road test a total of 200 image frames were processed over a time period of 20 seconds (i.e., the time between consecutive frames is 100 milliseconds).

Table 4.1 Results of the test showing the points tracked and the angle between the tracked points.

Frame Comparison Code (Z)	Corner Ax	Corner Ay	Corner Bx	CornerB y	Angle between A and B (Alpha)	Number of total points in frame (Z-1)	Number 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

Table 1 provides some details about the captured image frames and the calculated angles from individual frame features. Although all 200 frames were analyzed, only a section of the results (i.e., the first 10 frames) has been shown in Table 4.1.

The frame comparison code (Z) in Table 4.1 represents that the 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 denote the coordinate points of individual features (Harris corners) found in the image Z-1 that are to be searched for in the next image Z. The total number of features in the 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 the features (Ax, Ay) in the subsequent frame to find their corresponding new locations (Bx, By). Not all the features found in the image Z-1 could be tracked in the following image. Sometimes the features are not found or it may be that the features did not pass a preset 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 consecutive images. The angle is then determined for each feature and averaged for each Z as given in Equations 14-16. For instance, in Z = 2 when the captured 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 the image 2. Thus angle of these 29 points are estimated and averaged. Table 4.2 shows the angle averaged over each Z.

Table 4.2 Averaged 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 gives the resulting angle (Beta) for each Z derived from averaging the feature angles. In Table 4.1, detailed results of Z from 1 to 10 are shown while the outcome for the first 20 Z's are shown in Table 4.2. While averaging, a condition is placed such that if the number of features tracked is below a certain threshold (we have set it to 5 here) then its averaged angle (Beta) is set to zero. The reason 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 features tracked to be averaged, or else the angle is by default assigned zero. After processing a batch of every ten number of Z's, that is at every 1 second, average of 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; that is at time t = 1

and 2 seconds the angles are 18.16^o and 8.07^o, respectively. Hence using this heading angle, the lateral displacement of the vehicle from the center of the lane is computed as shown in Table 4.3 from t = 1 to 19 seconds.

Table 4.3 A sample test data showing the heading angle and lateral displacement.

Time (sec)	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

4.2 Results

In this section, we explain the some of the results we found while conducting the tests on Martin Road, Jean Duluth Road, Lavaque Road and Highway 61 outside the city of Duluth limit. As mentioned in Section 4.1, the speed limit on part of Jean Duluth and Lavaque is 50 mph, while it is 65 mph on Highway 61. Figure 4.2 shows the heading angle and lateral position of the vehicle traveling at 50 mph. In Figure 4.2, during the first 10 seconds, the heading angle is positive meaning that the vehicle is moving toward its right direction and thus, the vehicle deviates from the center of the lane but less than 2 ft. The following negative angle indicates the vehicle is moving to the left relative to its previous direction. Overall, the deviation from its center is within the range of -0.5 to1.5 ft in this case.

The results in Table 4.3 are shown in Figure 4.3. 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 practice, maintaining a zero angle would be impossible; so a few degrees variations are expected.

In Figure 4.3, the heading angle and lateral displacement with respect to its previous position is shown at every second. As can been seen, the range of the heading angle lies between 18.7^{o} and -12.76^{o}, and the lateral displacement is from 2.97 feet (right) to -2.11 feet (left).

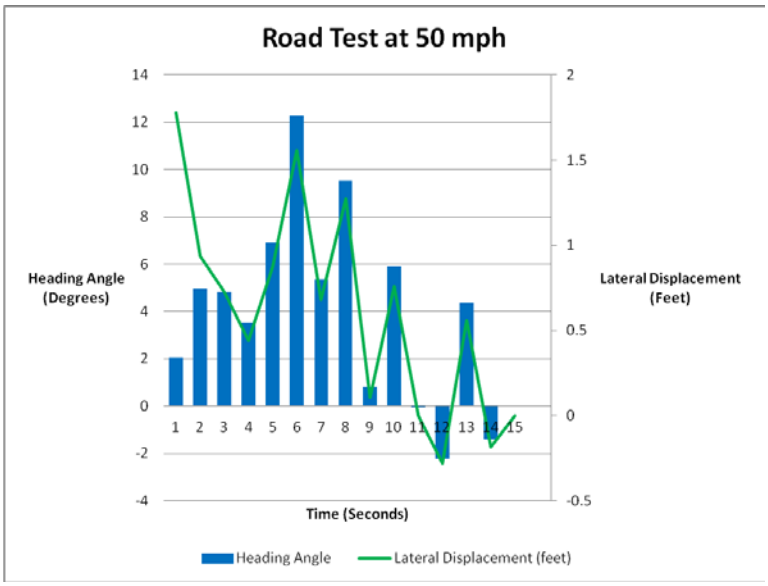

Figure 4.2 Road test showing vehicle's heading angle and lateral position with 50 mph speed.

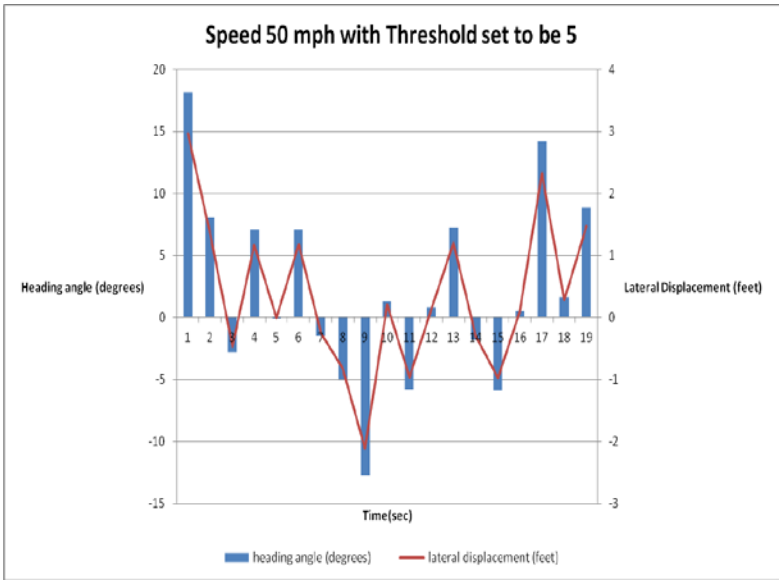

Figure 4.3 The vehicle's heading angle and lateral position – Case 1.

In order to study the effect of changing the threshold of the feature selection on the heading angle, and thus, the lateral displacement, we analyzed the data with threshold set to 5 and 15. Figures 4.3 and 4.4 show the heading angle and lateral position with 50 mph speed but a different threshold (i.e., 5 in Figure 4.3 and 15 in Figure 4.4). Overall, they look similar. The

threshold has some influence on the heading angle but not significant in this case. We see that the vehicle deviates from its center of the lane but remains within ± 4 ft. Figures 4.5 and 4.6 show a similar results with a higher 65 mph speed on Highway 61. Again, the deviation from the center of the driving lane lies within the 5 ft range.

In order to improve the accuracy of the system operation over time, it is important to reset the vehicle's "initial" position as often as possible, as shown in Figure 2.12. Therefore, in our Phase II study, a lane detection algorithm was also integrated with the operation algorithm to implement Figures 2.11 and 2.12. We used the Hough transform [16, 26, 27, 28] which is a method for findings lines, circles, or other simple forms in an image and it is a feature extraction technique concerning with the identification of lines in the images. We used this technique to detect the lane edges and thus, to reset the "initial" position when the lane markers become available. It can also be used to deal with the road curvature issue. We will report our findings in the Phase II final report.

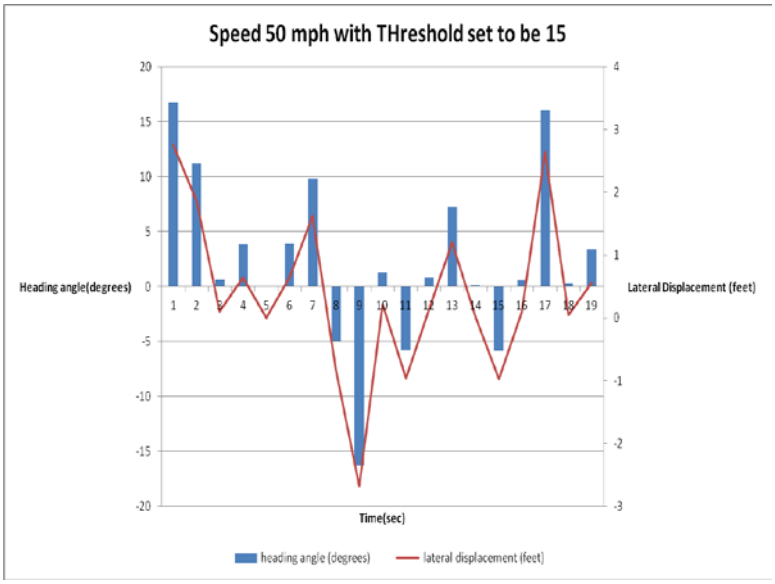

Figure 4.4 The vehicle's heading angle and lateral position – Case 2.

Note that there are a lot of factors that could cause the discrepancy in the heading angle we found and thus, the vehicle's lateral displacement. Possible factors that could affect the results are:

- Even though those points tracked pass a certain threshold of accuracy, it is still some points might not have been tracked accurately enough.
- Sometimes the consecutive images may look very similar (i.e, at higher sampling rate and/ot high speed), making it harder for the algorithm to detect changes 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 histogram equalization could help by increasing the image contrast.
- The default setting of the angles (Beta) to zero could deviate the actual angle while averaging.

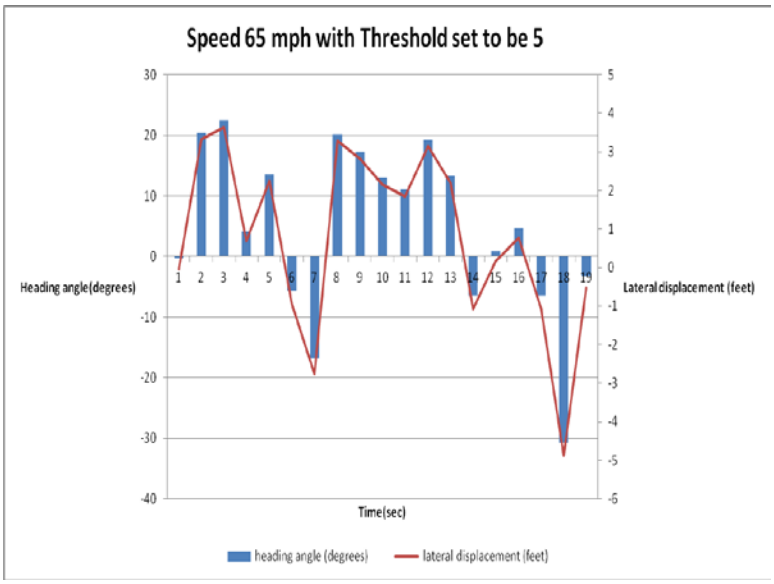

Figure 4.5 The vehicle's heading angle and lateral position – Case 3.

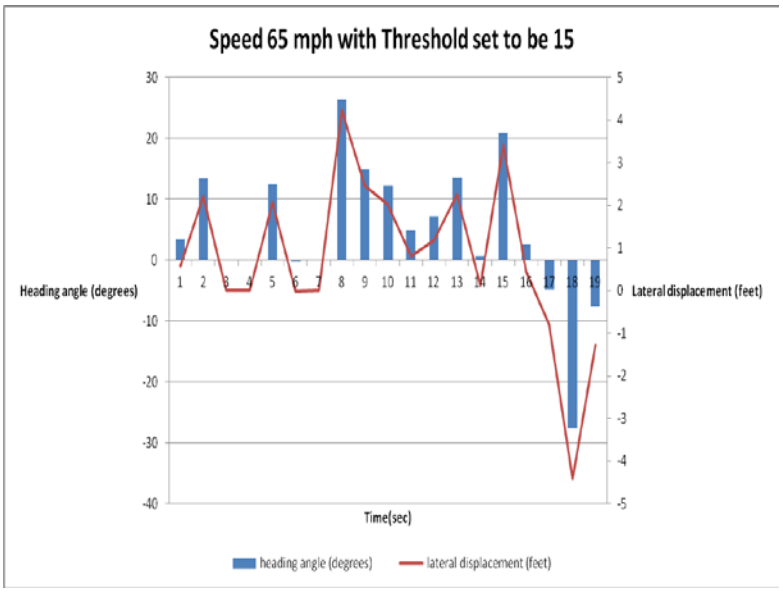

Figure 4.6 The vehicle's heading angle and lateral position – Case 4.

Overall, the results showing the vehicle's lateral position seem reasonable to reflect the actual driving situation and are also satisfactory. However, refinement of the operation algorithm shown in Chapter 2 will be continued via further extensive road tests and will be used and integrated in our Phase II study of developing a LDW system.

Chapter 5. Conclusion

The main objective of this research is to determine the vehicle's lateral characteristics to lay a foundation for the development of an innovative LDW system without knowledge of lane makers and vehicle's GPS location. However, before developing such a system, it is essential that we know the vehicle's lateral characteristics, in particular, the vehicle's lateral position and speed. In this study, we use image processing via an in-vehicle camera to estimate the vehicle's lateral position and speed. The lateral position is estimated by determining the vehicle's heading angle obtained via a homography transformation and the Lucas-Kanade optical flow techniques, while the lateral speed is determined via the heading angle and the vehicle's OBD-II port and an OBD-II interpreter. The histogram equalization is used to enhance the image contrast quality to improve the feature selection and tracking. Based on the implemented operation algorithm, we conducted extensive road tests with 50 mph speed on segments of Jean Duluth Road and Lavaque Road, located north of Martin Road, and we also did tests on Highway 61 with 65 mph speed. The detail of our approach is presented in this report together with our findings. We found possible factors that could cause the discrepancy in the heading angle, and thus the vehicle's lateral displacement. Overall, the results showing the vehicle's lateral position are satisfactory. Extensive road tests with refinement will be continued in the Phase II study of developing a fuzzy rule-based LDW system.

Our approach only needs the minimal set of information to characterize the vehicle lateral characteristics, and therefore, makes it more feasible in a vehicle application.

References

[1] C. Liu and T. Ye, *Run-Off-Road Crashes: An On-Scene Perspective* (DOT HS 811 500), National Highway Traffic Safety Administration (2011).

[2] C. F. Lin and A. G. Ulsoy, "Time to Lane Crossing Calculation and Characterization of Its Associated Uncertainty," *ITS Journal*, vol. 3, no. 2 (1996), 85-98.

[3] H. Godthelp, P. Milgram, and G. Blaauw, "The Development of a Time-related Measure to Describe Driving Strategy," *Human Factors*, vol. 36, no. 3 (1984), 257-268.

[4] J. Hayward, "Near-miss Determination through Use of a Scale of Danger," *Highway Research Rec*ord, no. 384 (1972), 24-34.

[5] D. Pomerleau, "RALPH: Rapidly Adapting Lateral Position Handler," *Proceedings of the IEEE Intelligent Vehicles Symposium*, (1995), 506-511, Detroit, MI.

[6] D. LeBlanc, G. Johnson, P. Venhovens, G. Gerber, R. DeSonia, R. Ervin, C.-F. Lin, A. G. Ulsoy, and T. Pilutti, "CAPC: A Road Departure Prevention System," *IEEE Control Systems Magazine*, vol. 16 (1996), 61-71.

[7] C. R. Jung and 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* (2005), 348-35, Vienna, Austria.

[8] S. Shladover, C. Desoer, K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated Vehicle Control Developments in the PATH Program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1 (1991), 114-120.

[9] R. E. Turochy, "Shoulder Rumble Strips: Evolution, Current Practice, and Research Needs," *Transportation Research Board Annual Meeting*, Washington, D.C. (2004).

[10] K. Perrillo, *The Effectiveness and Use of Continuous Shoulder Rumble Strips*, National Highway Sleep Foundation, Washington, D.C. (1998).

[11] P. Batavia, "Driver-adaptive Lane Departure Warning System," Ph.D. Dissertation, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA (1999).

[12] M. Gonzalez-Mendoza, B. Jammers, N. Hernandez-Gress, A. Titli, and D. Esteve, "A Comparison of Road Departure Warning Systems on Real Driving Conditions," *Proceedings of The 2004 IEEE Conference on Intelligent Transportation Systems*, (2004), 349-353, Washington, D. C.

[13] J. Manigel and W. Leonhard, "Vehicle Control by Computer Vision," *IEEE Transactions on Industrial Electronics*, vol. 39, no. 2 (1992), 181-188.

[14] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," *Proceedings of Imaging Understanding Workshop* (1981), 121-130, Washington, DC.

[15] G. Bradski and A. Kaebler, *Learning OpenCV Computer Vision with the OpenCV Library*, Cambridge, MA: O'Reilly, 2008.

[16] Wikipedia http://en.wiki pedia.org/wiki/On-board_diagnostics

[17] H. Wang M. Ren, and S. Shao, "Lane Markers Detection based on Consecutive Threshold Segmentation," *Journal of Information and Computer Science*, vol. 6, no. 3 (2011), 207-212.

[18] T. Teshima, H. Saito, S. Ozawa, K. Yamamoto, and T. Ihara, "Estimation of FOE Without Optical Flow for Vehicle Lateral Position Detection," *Proceedings of IAPR Conference on Machine Vision Applications* (2005), 406-409, Tsukuba Science City, Japan.

[19] E. De Castro and C. Morandi "Registration of Translated and Rotated Images Using Finite Fourier Transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5 (1987), 700-703.

[20] E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*, Chichester, UK: John Wiley & Sons, 2003.

[21] B. K. P. Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, no. 1-3 (1981), 185-203.

[22] B. Glocker, N. Komodakis, G. Tziritas, N. Navab & N. Paragios, "Dense Image Registration through MRFs and Efficient Linear Programming," *Medical Image Analysis*, vol. 12, no. 6, (2008), 731-741.

[23] T. Teshima, H. Saito, S. Ozawa, K. Yamamoto, and T. Ihara, "Vehicle Lateral Position Estimation Method Based on Matching of Top-View Images," *Proceedings of The 18th International Conference on Pattern Recognition* (2006), 626-629, Hong Kong, China.

[24] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proceedings of the 4th Alvey Vision Conference* (1988), 147–151, Manchester, UK.

[25] J. Shi and C. Tomasi, "Good Feature to Track," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (1994), 593-599, Seattle, WA.

[26] P. V. C. Hough, "Machine Analysis of Bubble Chamber Pictures," *Proceedings of the International Conference on High Energy Accelerators and Instrumentation* (1959), 554-556, Geneva, Switzerland.

[27] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the Association for Computing Machinery*, vol. 15 (1972), 11-15.

[28] L. Shapiro and G. Stockman, *Computer Vision*, Upper River Saddle, NJ: Prentice Hall, 2001.