Road and Traffic Analysis from Video

by

Erhan Bas

A Thesis Submitted to the
Graduate School of Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Electrical and Computer Engineering

Koc University

August 2007

Koc University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Erhan Bas

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Dr. A. Murat Tekalp

_____

Asst. Prof. F. Sibel Salman

_____

Asst. Prof. Engin Erzin

Date: _____

To my parents, and my rose

ABSTRACT

This thesis proposes two video-based traffic analysis systems, one for traffic monitoring with fixed cameras, and one for driver warning applications with on-board cameras looking outwards from the windshield. In the fixed-camera traffic monitoring system, Gaussian Mixture Model (GMM) based background subtraction is applied with a new adaptive bounding box size criteria to detect and track vehicles. An automatically extracted road mask is used to reduce the computational complexity. Furthermore, a new occlusion reasoning algorithm is proposed for robust tracking and counting of vehicles, where features such as size and width of the vehicles are used. Proposed system is tested under different lighting and weather conditions, such as night and winter recordings. In the driver warning system with on-board camera, host vehicle localization with respect to lane marks and vehicle-to-vehicle distance calculation are addressed. A feature based lane mark detection scheme with two step aggregation method is proposed. Edge features are used in this study, and aggregated into more meaningful structures by a new two-step aggregation method. Tracking of the individual lane mark is handled by four Kalman filters for each of the lane mark corner. After analyzing the tracking results, two modes are defined for the host vehicle: in-lane and passing modes. Reliability of the proposed system is tested for host vehicle localization and vehicle-to-vehicle distance on a video sequence including both modes. Moreover, a new scene initialization procedure based on global motion estimation is used in this study. Experimental results show that the proposed algorithms perform well, and they are robust to environmental conditions.

ÖZET

Bu tezde birbirinden ayrı 2 tane video tabanlı trafik analiz sistemi gerçekleştirilmiştir. Birincisi sabit kamera düzeneğinden trafik gözetimi, ikincisi ise araç üstüne yerleşik kameradan sürücü uyarı sistemidir. Sabit kamera düzeneği ile trafik izleme sisteminde araç tespiti ve takibi için, Gaussian Mixture Model (GMM) tabanlı arkaplan çıkarımı yöntemi kullanılmıştır. Yeni öngörülen uyarlamalı kuşatan kutu yöntemi uygulanmıştır. Hesaplama karmaşıklığını azaltmak izin otomatik çıkarılan yol maskesi kullanılmıştır. Ayrıca, gürbüz araç takibi ve sayımı için araç boyutlarının kullanıldığı yeni bir örtüşme algoritması oluşturulmuştur. Önerilen sistem farklı ışıklandırma ve hava durumları için test edilmiştir. Araç üstü monte edilmiş kamera yardımıyla sürücü uyarı sistemi için yol şeridinden araç lokalizasyonu ve araçlar arası mesafe tespiti uygulamaları ele alınmış, özniteliğe dayalı iki adımlı şerit işareti tespiti algoritması gerçekleştirilmiştir. Bu çalışmada kenarlar öznitelik olarak kullanılmış ve yeni iki basamaklı gruplama yöntemi ile daha anlamlı yapılara kümelenmiştir. Şerit işaretlerinin bireysel takibi, şerit işaretlerinin her köşesine yerleştirilen Kalman filtreleri ile yapılmıştır. Takip sonuçlarına dayalı olarak iki tane mod tanımlanmıştır: şerit arası modu ve geçiş modu. Öngörülen sistemin güvenirliği iki modu da içeren imge dizisi için test edilmiştir. Ayrıca, sistem ilklendirmesi için global hareket kestirimi yöntemi kullanılmıştır. Deney sonuçları önerilen algoritmaların düzgün çalıştığını ve çevresel etkenlere karşı gürbüz olduğunu göstermektedir.

ACKNOWLEDGEMENTS

First I would like to thank my supervisor Prof.A.MuratTekalp who have been a great source of inspiration and provided his right balance of suggestions, criticism, and freedom.

I am grateful to members of my thesis committee for critical reading of this thesis and for their valuable comments.

I also would like to thank Istanbul Municipality for providing the video recordings that used in the second chapter of this thesis.

Finally I thank my mother, for her thoughtful nature, my father for his kindness, Erdal the Great for being my brother, his wife for being the zotu , and my rose for providing me a morale support that helps me in hard days of my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

DFD	Displaced Frame Difference
GMM	Gaussian Mixture Model
KLT	Kanade-Lucas-Tomassi Feature Tracker
IRLS	Iteratively Reweighted Least Squares
RANSAC	Random Sample Consensus
RMSE	Root Mean Square Error
ROI	Region of Interest
TZ	Tracking Zone
$\|\|\boldsymbol{i}\|\|$	i^{th} Detected Line

Chapter 1

INTRODUCTION

1.1 Motivation

As vehicle population increases, ITS (Intelligent Transportation Systems) becomes more significant and mandatory in today's overpopulated world. Vital problems in transportation such as mobility and safety of transportation are considered more, especially in metropolitans. ITS mainly aims to obtain safer traffic conditions, comfort in transportation, and to increase the road- traffic efficiency by improving the functionality of cars and roads [1]. For this purpose, information systems for locations, and warning systems for vehicle safety have previously been implemented to enhance driver's ability to sense the surrounding environment.

ITS applications can be applied to different areas such as traffic infrastructure management and intelligent vehicles [2]. In this thesis, vision-based ITS application is considered for road traffic monitoring using static cameras and driver warning systems using an onboard camera. In road monitoring application, the video cameras are placed on posts above the ground to obtain complete view of the road and passing vehicles. In the driver warning application, the on-board camera is placed looking outwards from the windshield.

Road traffic monitoring aims at the acquisition and analysis of traffic figures, such as presence and numbers of vehicles, and automatic driver warning systems are developed mainly for localization and safety purposes [3]. Video systems for either traffic monitoring or

driver warning normally involve two major tasks of perception: i) estimation of road geometry and ii) vehicle and obstacle detection.

Temporal differencing and background modeling techniques are widely used for vehicle and obstacle detection in road traffic monitoring systems. Although background modeling techniques provide more reliable results, computational complexity is a trade off that one has to consider. Adaptive systems to environmental changes have been proposed, and robust tracking algorithms are implemented under various conditions such as occlusion. 2-D and 3-D algorithms have been implemented to analyse the scene, where robust feature sets are used. In order to obtain valid feature sets, most of the studies use close-up static cameras, where sufficient number of features can be obtained. It is worth noting that there are less number of studies considering road geometry for detecting and tracking of the vehicles. Thus, implementing traffic monitoring systems requiring less features for detection/tracking, and that can be used for general camera setups is beneficial.

The Traffic Control Center of Istanbul Municipality collects real-time images using a video processor system consisting of 110 cameras of various characteristics. Currently, all of the images are displayed at a control room and are monitored by operators to detect any incidents such as accidents or unexpected road conditions. In the second chapter of this thesis, a feasibility study that can be used to analysis the traffic flow for mentioned purposes is implemented.

Automatic driver warning system is another crucial application that used in ITS applications. Road geometry and vehicle detection algorithms have been thoroughly examined, and different warning applications have been proposed in the literature. Most of the vehicle warning systems can be classified into two groups for onboard camera applications: feature and model based. In the first group, feature based methods are used. This group is mainly derived from detection of edges as features in the image, and aggregation of these features into meaningful models [22]-[23]. In general, feature driven approaches are highly dependent on the methods used to extract features and they suffer from

noise effects and irrelevant feature structures. Lane tracking is predicated on lanes rather than lane marks, and priori information about the lane structure is needed for lane model [22]. Often in practice the strongest edges are not the road edges, so that the detected edges do not necessarily fit a straight-line or a smoothly varying model. The second group is the model based methods. In this group, deformable templates are used to describe a scene with a set of parameters that fits to the scene model. The road boulders and lane markings are often approximated by circular arcs on a flat-ground plane. Model-driven approaches provide powerful means for the analysis of road edges and markings. Their main advantage is that the lane can be tracked with a statistical technique, thus, false detections are almost completely avoided. However, the use of a model has certain drawbacks, such as the difficulty in choosing and maintaining an appropriate model for the road structure, the inefficiency in matching complex road structures and the high computational complexity [25] - [31].

To increase transportation safety, to decrease the number of traffic accidents and hence to save lives and valuable resources, Drive-Safe project that is supported by the State Planning Organizations is currently in progress. The aim of this project is to create conditions for prudent driving on highways and roadways with the purposes of reducing accidents caused by driver behavior. To achieve these primary goals, critical data is being collected from multimodal sensors (such as cameras, microphones, and other sensors) to build a unique databank on driver behavior. In the third chapter of this thesis, a feasibility study for a driver warning system that is based on the windshield camera recordings is implemented parallel to the Drive-Safe project.

Chapter 2 provides a feasibility application for road traffic monitoring system, which considers the adaptive blob size method and occlusion of vehicles for robust traffic parameters estimation, and can be used in traffic flow analysis.

Chapter 3 presents a new lane detection and tracking methodology that is robust to various environmental conditions for automatic vehicle warning applications. Different cases

are discussed for host vehicle motion behavior, and a simple vehicle to vehicle distance algorithm is described for safety applications.

The thesis is concluded with a short summary of the performed study in Chapter 4.

1.2 System Overview

In this thesis we design two separate systems that will be used in traffic monitoring and driver warning applications. Innovative approaches are implemented for each system to analyse traffic flow and vehicle localization under different environmental cases.

In most of the traffic monitoring applications, reliable detection and counting of the vehicles are the major issues. In previous studies, different approaches have been proposed to handle these tasks. Vehicle detection is mainly obtained by comparison of current frame with the empty road; therefore the task is equal to how we model the empty road. For this purpose temporal differencing [5] and GMM [7-8-15] based modelings are the commonly used models for the road. Although, temporal differencing has smaller computational complexity compared to GMM [15], adaptation to environment is another important and mandatory issue for road modeling. Likewise the complexity; it is a shortcoming of background modality that it requires priori information about the scene as described in [7]-[8]. Detection of vehicles is mainly implemented by extracting the foreground regions, which is a highly correlated procedure with the image geometry. Previous works generally do not consider the image geometry. Moreover, to eliminate such effects camera position is selected accordingly or airbone cameras are preferred. Subsequently, several vehicle features such as length, width, velocity, etc., are extracted for the traffic flow analysis. However, several environmental variations will heavily affect the accuracy and robustness of traffic flow analysis. For example, occlusions will result in the failure of vehicle detection and further degrade the accuracy of vehicle counting. Another important issue in traffic monitoring applications is that most of the model based detection, tracking and classification algorithms are

implemented for close-up videos, where it is easier to fit a pre-defined vehicle or motion model to understand the scene[12]-[15],[18]. Close-up recordings are also used for feature based applications such as [9], where at least 4 corner points have to be found on a vehicle as features. In all these works, the use of close-up images enables the selection of reliable vehicle model and features for tracking and occlusion reasoning algorithms. It is essential to build a traffic monitoring application that is robust to occlusions, and uses only low level features such as size for traffic analysis which can be used for non close-up recordings.

Automatic driver warning systems are also studied in this thesis. We implemented a feature based lane (and/or lane mark) tracking scheme that low level features are used to aggregate into more complex structures. As it is the case in the previously works implemented in driver warning systems, edge features are used in this study. Proposed lane mark model in this study is similar to the one implemented in [22], where lane marking structure is used in both studies. The problem with such a structure is that the feature based methods are highly dependent to extracted features, i.e. edges. Most of the previous works include the robust line detection algorithms in the presence of shadow edges, then aggregation of detected lines into lane structures. However, there are only a limited number of studies examining the problem as an aggregation problem rather than line detection event. Moreover, previously proposed algorithms are built on strict lane structure assumptions, such as constant lane width. To reduce false detection and to achieve reliable tracking a novel two-step aggregation approach is maintained in this study, which considers the dynamic change of the lane structure, as well. The common problem, obtaining reliable features, in feature based lane detection methods is handled by this two-step approach Another notable problem in automatic driver warning systems is that in model based methods, choosing an appropriate model for road geometry is a difficult task. In addition, possible matching errors can be resulted for simple road structures [28], especially when the lane marks are occupied by other vehicles. This matching error is also seen in feature based methods that tread the lane marks as single lane. To prevent this error, individual tracking algorithms are

implemented for each lanes present in the scene in this study. One last issue in driver warning systems is the initialization of the proposed systems. There are already some applications using ROI to narrow detection and tracking region for computational complexity purposes. Some applications assume constant search space [32, 34, 43], or adaptive systems as in [29, 33] have been presented to restrict the search space for lane detection. Authors of [33] implemented a preprocessing step that calculates vanishing point from the all available lines. They use vanishing point to restrict the search space for lane detection. All the works cited above use the tracked lane parameters that are calculated at the previous frame, but in terms of initialization they make some assumptions to detect initial ROI. We implemented a novel dynamically updated ROI initialization method, which does not need any specific scene assumptions. Our only assumption about the scene is the flatness of the road, which provides us freedom in scene analysis. Localization and motion analyse of the host car is estimated according to the tracking of lane parameters, where a simple yet efficient vehicle to vehicle distance system is implemented in this study.

1.3 Contributions

This thesis study has several contributions to the study areas of traffic monitoring and driver warning systems. The contributions to the traffic monitoring applications can be stated as follows:

- Introduction of a new traffic monitoring system application working under non close-up images that uses adaptive bounding box size for detection of vehicles:

  In order to achieve reliable extraction of vehicles from detected foreground regions, adaptation to image geometry is an important but mostly discarded task in the literature. For this purpose a detection algorithm that uses adaptive bounding box size of vehicles according to camera position is implemented in this study.
- Automatic road mask extraction:

For reliable detection of vehicles, GMM based background subtraction scheme is followed in this study. To reduce computational complexity of the modeling process, road mask is used to constrain the image region, so that other regions will not be processed, where road mask is automatically obtained from the motion history of the vehicles.

- An innovative robust tracking system for traffic flow analysis under occlusions:

  To implement a reliable traffic flow analysis, accuracy of vehicle counting is essential. Occlusion related problems will result in the failure of vehicle detection, and descent in accuracy of vehicle counting. For this purpose a novel occlusion reasoning algorithm is implemented for robust tracking and counting of all vehicles, where easy-accessed features such as size and width of the vehicles are used.

Moreover, this work aims at to implement a lane detection scheme, which compensates the trade off between the feature based methods and model based methods. Considering this aim, contributions on the driver warning system can be grouped under three headings.

- A new two-step aggregation approach:

  To reduce false detection and to achieve reliable tracking, a novel two-step aggregation approach is maintained in this study. In the first step, to increase the number of features, line detection constraints are reduced, and lines are aggregated to more meaningful line segments, i.e. lane mark boundaries, according to their alignments in the scene. In the second step extracted lines are further grouped into lane structures according to the defined lane constraints. Lane constraints mainly depend on the lane geometry and unlike traditional methods; they are adaptive to imaging geometry that does not assume any lane mark model specifications such as constant lane width

- Individual tracking of lane marks:

  In this study, despite the previous feature based methods that combine the lane marks to form a single lane, or as in lane model approach that tries to fit a template to observed

lane, each lane mark in the scene is tracked with four Kalman filters, initialized for each corner of the lane mark to reduce possible matching errors. This approach also enables to analyse the roads having small radius of curvature by comparing the lane parameters, and lane mark changes. Moreover, whole lane curvature can be estimated from the lane mark models assigned to the corresponding lane.

- A new ROI extraction method:

In this study, the proposed system initialization is based on the global motion estimation of the scene that eliminates the relative motion due to the camera. The only assumption in this initialization step is that the road belongs to the estimated region defined by the global motion. Further restriction and correction on the ROI will be present with the tracked lane marking parameters as similar in traditional approaches.

Chapter 2

ROAD AND TRAFFIC ANALYSIS FROM FIXED CAMERA

2.1 Introduction

Traffic congestion is one of the main problems in many metropolises, including the city of Istanbul. Istanbul has over 2.35 million motorized vehicles according to July 2006 figures. Any analysis aimed at improving the problems related to congestion and enabling efficient transportation within the city requires collection of reliable data. In order to monitor the traffic flow, Istanbul municipality has installed more than 110 video cameras along the major arteries in the city [4], and this number is increasing. Hence, it is of interest to digitally process and analyze these videos in real-time in order to extract reliable data on traffic flow and to detect traffic events. For example, as a result of such video analysis, traffic density in major arteries can be estimated and the least congested routes and travel time estimates can be computed and transmitted to drivers over cell phones. In addition, the videos may be analyzed to automatically detect events such as accidents and traffic violations, as well as snow accumulation and other weather conditions. The data may also be used as input for traffic models and related planning problems.

In this chapter of the thesis, we propose a new feasibility study that can be used in traffic monitoring system applications. Geometry of the scene is accounted for, where adaptive bounding box size is used to detect and track vehicles according to their estimated distance from the camera. In the remaining part of this section, we will give a brief summary of the previous work that appeared in the literature, and our contributions. Then in Section 2,

vehicle detection and tracking algorithms are provided. Section 3 presents special considerations that include detection and tracking in night time and in snowy environments. Experimental results will be presented in Section 4 in this chapter.

2.1.1 State-of-the-Art

Several studies exist in the literature on automatic video analysis for vehicle detection and tracking. For example, a double-difference operator with gradient magnitude has been used to detect vehicles [5]; however, it cannot easily handle interframe luminance variations. Optical flow techniques have been used to estimate the motion between subsequent frames [6]. Adaptive background subtraction algorithms have been used for vehicle detection, which allows changes in lighting and weather conditions [7] [8], but they usually require a priori information about the scene without any moving vehicles and have problems with occlusions. Authors of the [9] propose a vehicle tracking system which can detect and monitor vehicles as they break traffic lane rules. Their proposed tracking scheme is based on characteristics of both traffic scene and vehicle, where each vehicle is represented with four feature points. In addition, Lipton et al. [10] used maximum-likelihood estimation criteria with shape features to classify different targets into vehicles and humans. Furthermore, Gupte et al. [11] proposed a region-based approach to track and classify vehicles based on the establishment of correspondences between regions and vehicles. In addition, 3-D models have been previously implemented such as Sullivan [12], which recover trajectories with high accuracy, and classify vehicles into various types like sedan, and hatchback. However, this approach requires detailed geometric object models for all detected vehicles on the highway. Likewise, 3-D tracking algorithms based on detection of vehicles with probabilistic line feature grouping method such as [13] have been previously implemented. In addition, Kalman filter has been widely used in automatic traffic monitoring systems. For example, Xie et al. [14] use position and size information as state variables to track vehicle positions

with different set of features. Similarly, [15] uses Kalman filter for tracking vehicles extracted from background models. They implemented a shadow removal algorithm to extract the size and linearity features of vehicles for the purpose of categorizing them.

Various methods have been suggested which manage occlusions in literature. In [16] authors propose to track vehicles as a set of parts to resolve partial occlusions. To fulfill the tracking purpose a specific segmentation method is proposed, which make use of active contours. The goal is to obtain parts that are the most suitable for a gradient-based tracking algorithm. However, presented results show only occlusions for cars that move out of the field of view and only for one car at a time. The authors in [17] propose a technique for removing outliers from the trajectories of feature points fitting a subspace and removing those points that have large residuals. They generate and track features through the entire video stream using the KLT algorithm, then apply the RANSAC method to detect trajectories that does not follow the rigid movement constraint. In [18] tracking is performed by template matching, with a probabilistic model based on a robust error norm. Matching is performed by finding the image region that yields the maximum likelihood with respect to the calculated probability distribution. Templates are matched by translation, rotation and scaling parameters; during occlusions, a pixel is regarded as an outlier if the measurement error exceeds a predefined threshold. Templates cannot be updated during occlusion, so it is necessary to detect when overlapping ends. This algorithm can manage complete occlusions, but it cannot withstand severe occlusions lasting more than 25 frames, due to wrong Kalman filter estimation. In [19] authors proposed an occlusion robust method based on Spatio-Temporal Markov Random Field Model. Each frame is divided in blocks of 8×8 pixels and the feature correlation between blocks of consecutive images is performed by exploiting motion vectors for each block. In order to determine which object a block belongs to, a complex minimization problem must be solved. The authors of the [20] proposed a method that categorizes vehicles into specific classes by introducing a linearity feature in vehicle

representation. They implemented an algorithm that detects lane-dividing lines which enables to handle vehicle occlusions caused by shadows.

2.2 Video Analysis for Vehicle Detection and Tracking

Video cameras were first introduced to traffic management for roadway monitoring by transmitting closed-circuit television imagery to a human operator for interpretation. Present-day traffic management systems utilize digital video processing to automatically analyze the scene of interest and extract information for traffic monitoring. A video processor typically consists of one or more cameras, a microprocessor-based computer for digitizing and analyzing the imagery, and software for interpreting the images and converting them into traffic flow data. The Traffic Control Center of Istanbul Municipality collects real-time images using a video processor system consisting of 110 cameras of various characteristics. Currently, all of the images are displayed at a control room and are monitored by operators to detect any incidents such as accidents or unexpected road conditions.

Video processors can typically classify vehicles by their length and report vehicle density and speed for each class and lane. Video processors that track vehicles may also have the capability to register turning movements and lane changes. Vehicle density and link travel time are potential traffic parameters that can be obtained by analyzing data from a series of image processors installed along a section of roadway.

2.2.1 Vehicle Detection

In order to distinguish moving vehicles from the static background, we model the background scene with GMM (Gaussian Mixture Modeling) as in [21]. Each pixel color is modeled by a mixture of K Gaussian distributions with specified weight parameters (K is some number from 3 to 5) over a time interval. The weight parameter of a certain mixture is

the data proportion that is accounted for by the corresponding distribution. The idea for moving object detection lies in the wider color characteristic of moving objects due to different reflecting surfaces during the movement. Since steady objects form tight color clusters, the rule to decide whether a new pixel belongs to the background or the foreground is based on the variance of this pixel in comparison to the background model. That is, the color value of every pixel is checked to decide whether it matches the GMM or not. A pixel color value that is less than 2.5 standard deviations from the mean of any of the K distributions is decided to belong to the background. If a match occurs, then that mixture (weight parameter, mean and covariance) is updated with the new pixel color value; if no match occurs then a new mixture model is created with the mean at that pixel value and an initially high variance value. The Least probable (smallest weighted) distribution is replaced with the new model with a small weighting.

It is important to adapt the background model to small changes such as brightness variations or new entries to the background. For this purpose, an online update algorithm is used. The probability of observing a certain pixel value for a channel (a vector for R-G-B channels, or a scalar value for a single Gray level channel) after *t* frames is given as

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2.1}$$

where $w_{i,t}$ is the weighting parameter of the i^{th} Gaussian mixture. Here, $\mu_{i,t}$ and $\sum_{it}$ represent the mean value and covariance matrix, respectively, of the i^{th} Gaussian distribution computed from 'n' channel pixel value history. In our implementation we set n=3. Distributions are ordered in descending probabilities according to their time interval in the scene. For background modeling, the first T among K Normal distributions are used. The number T is found from

$$T = \arg\min_n \left(\sum_{i=1}^{n} w_i > thr\right) \tag{2.2}$$

where the parameter *thr* represents the minimum portion of the data required to form a background model.

Difference images are formed by subtracting the current frame from the background model in each channel. By thresholding each channel (difference images), three binary channels are obtained. By intersecting these foreground images, which belong to different channels, a final single channel binary foreground image is obtained. Then, the foreground objects are detected and labeled using connected component analysis with adaptive blob size, where the blob size varies according to the position of the blob in the picture and imaging as explained below.

2.2.2 Adaptive Blob Size Fitting

For a fixed camera configuration, in imaging geometries where the road is along the z-axis of the camera, vehicles further away from the camera are expected to be smaller in size; hence are modeled by smaller blobs as shown in Figure 2.1. The adaptation of the blob size depends on the relative position of the camera with respect to the road. First, a mask of the road is extracted from vehicle trajectories to reduce the search space (Figure 2.2). This mask is also used to fit a cubic road equation that approximates the road curvature (for example the highway in Figure 2.3).

The relative road equation that is seen in Figure 2.3 is calculated from estimated center strip by fitting a second or higher order curve. Experimentally, we observed that a fourth order curve equation fits better to road curvature. The road equation enables us to approximate the values of size threshold for vehicle detection, and parameters such as the relative pixel speed. These parameters can later be used to estimate traffic flow, and give

information about imaging geometry without having camera calibration parameters. Moreover, by setting an upper threshold value for blob sizes gives priori information about anomaly cases such as occlusion and traffic congestion. Bounding rectangles are fitted to each detected blob, and the centers of the rectangles are marked as the vehicle position (Figure 2.3).

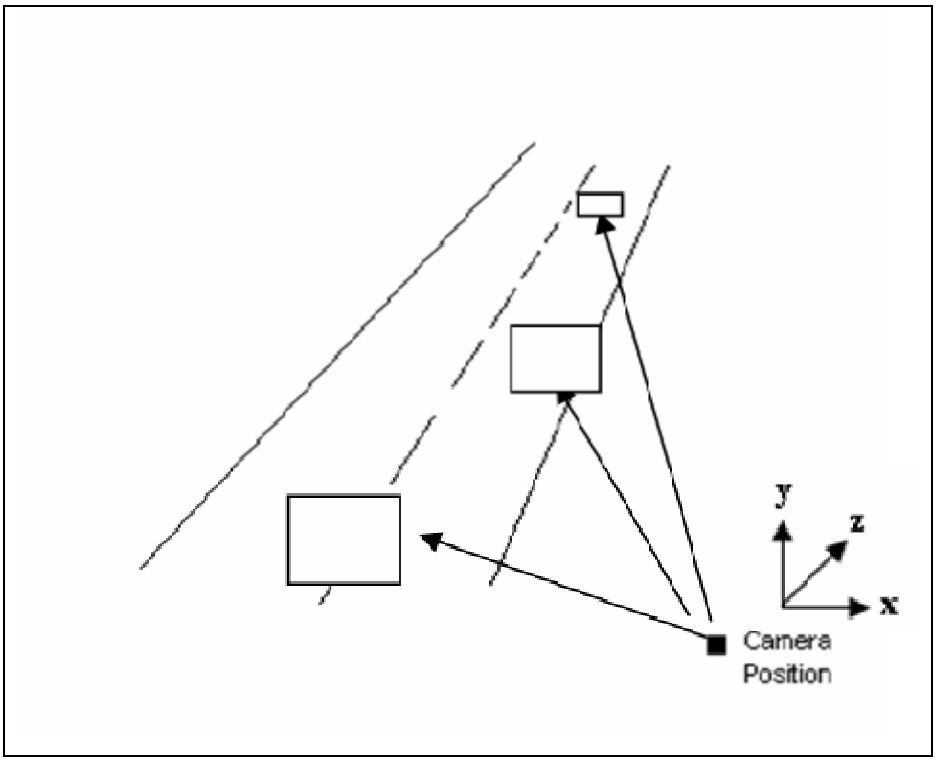


Figure 2.1: Adaptation of blob sizes. For a fixed camera set-up, blob size of a vehicle is approximated using its distance from the camera position.

Figure 2.2: Mask of the road is extracted from the vehicle positions and their trajectories.

Figure 2.3: Road equation, shown as the orange curve, is extracted from the mask of detection region.

2.2.3 Vehicle Tracking

In each frame, each detected vehicle is represented with a two-state Kalman filter, based on the constant-velocity motion model

$$\begin{aligned} x_t &= x_{t-1} + v_{x,t-1} \cdot T \\ y_t &= y_{t-1} + v_{y,t-1} \cdot T \end{aligned} \tag{2.3}$$

where T denotes the frame capture rate of the acquisition system. Velocities in vertical and horizontal directions are represented with v_x and v_y. Here x_t and, y_t denote the center of mass of the rectangles. The state-space model is formulated with state s_t and observation z_t as:

$$\begin{aligned} s_{t+1} &= F \cdot s_t + G \cdot u_t \\ z_t &= H \cdot s_t + o_t \end{aligned} \tag{2.4}$$

where,

$$\begin{aligned} s_t &= \begin{bmatrix} x_t & y_t & v_{x,t} & v_{y,t} \end{bmatrix}^T \\ z_t &= \begin{bmatrix} x_t & y_t \end{bmatrix}^T \end{aligned}$$

and

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The process noise u_t and the measurement noise o_t are assumed to be uncorrelated, with zero-mean white Gaussian distributions and corresponding covariance matrices Q and R as in [12]. Moreover, the rectangle center that is obtained by foreground segmentation is used as the observation, z, for the Kalman filter.

In order to decide if an observed vehicle position belongs to one of i) a previously existing vehicle, ii) a new incoming vehicle, or iii) a missing vehicle that occluded in the previous frames, we use the Euclidean distance between the optical flow estimate of the i^{th} vehicle's center position ($I_{t,i}$) and the observed vehicles' position at time t (θ_t). We denote this distance by $\delta(I_{t,i}, \theta_t)$. Decision varies according to the vehicles' positions in the imaging geometry. Since displacement would be higher for closer pixel locations, a higher threshold value is used for those regions.

The goal of optical flow calculation is to find the location $v = u+d$ in the frame at time $t+1$ for an image point $u = [u_x\ u_y]^T$ at time t, such that the windowed image regions centered at locations u and v, respectively $I_t(u)$ and $I_{t+1}(v)$, are "similar". The vector $d = [d_x\ d_y]^T$ is called the pixel motion or the optical flow vector at u. Similarity is defined in the mean square sense, and d is the vector that minimizes the residual function

$$\varepsilon(d) = \varepsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left(I_t(x,y) - I_{t+1}\left(x+d_x, y+d_y\right)\right)^2 \tag{2.5}$$

where w_x and w_y are integers that define the search neighborhood for optical flow calculation. For each match of observed vehicle position with optical flow estimate of an existing vehicle, matched vehicles' Kalman filter is updated with the observed vehicle position.

2.2.4 Occlusion Reasoning

Different cases are studied by relating observed rectangle centers with optical flow results. Occlusion of multi-vehicles is classified into two categories: occlusion of vehicles and split of occluded vehicles. In each frame, a log is used to keep track of the points, whether they are new observations or observations correspond to existing vehicles, including their occlusion status in terms of occluded frame number. Figure 2.8 and 2.9 illustrates the occlusion and split case consequently. The following steps are used to match an observed rectangle center with an existing vehicle and track them. In each frame:

1. For each optical flow estimate of vehicle center, rectangle centers are matched according to their Euclidean-distance $\delta(I_{t,i}, \theta_t)$ as explained below.

- **a.** If the distance $\delta(I_{t,i}, \theta_t)$ is below the threshold, an exact match occurs and the current Kalman filter ($K_{t,i}$) is corrected with matched rectangle center (θ_t).
- **b.** If the mean-square distance is above the threshold, there appears a split of an occluded vehicle. Kalman filter belonging to the split object is corrected with its prediction. In case of vehicles entering the scene, new Kalman filters are initialized at step 2.
- **c.** If the distance is above the threshold and two different optical flow estimates match with the same rectangle center, occlusion case is valid for the objects. A new Kalman filter is initialized at step 2 and prediction values are used as observations for the occluded objects' Kalman filter.
- **d.** If there is no match, i.e., distance is too high for the optical flow of a vehicle ($I_{t,i}$), the vehicle's Kalman filter is corrected with its prediction since there is not enough observation for the existing vehicle.

2. A new Kalman filter is initialized for each unmatched rectangle center.

By keeping a log table, split objects' centers and their corresponding Kalman filters can be removed; also tracking for occluded vehicles for a desired frame period can be possible. Also for the no match case, vehicles that are no longer in the scene can be detected and it is possible to estimate a traffic flow from the acquired data. Figure 2.7 represents the algorithm result, where detected vehicles are shown by rectangles while tracked positions are shown by dots. Rectangle colors differ according to vehicles positioned lane. Yellow dot is used for tracked vehicles and blue dot for newly entered vehicle into the scene. White line indicates the boundary of inbound and outbound lanes for counting.

In order to count passing vehicles, we define a boundary on the image for each outbound and inbound lane to form a region of interest as in Figure 2.7. When we detect that the position of a tracked vehicle gets out of this region in terms of pixel values, the counting algorithm increases the vehicle count by 1 for the corresponding lane.

2.3 Special Considerations

In order to create a robust, adaptive tracking system that can handle environmental and lighting changes, the proposed algorithm is tested under different video scene conditions such as night time and weather condition

2.3.1 Night Time

For night time recordings, a simple modification is applied to the algorithm. To reduce lighting effects, a higher threshold is set for the bounding blob size and a smaller threshold is set for difference images (Figure 2.4).

Figure 2.4: The white solid line represents the lower limit of the tracking boundary whereas yellow dots represent tracked vehicles and blue rectangles are detected vehicles.

2.3.2 Weather Conditions

The algorithm is further improved with histogram and bounding box size constraints to handle weather conditions such as snow accumulation. To eliminate non-vehicles, histogram of every detected rectangle's background model is calculated. A vehicle or non-vehicle decision is made based on the intensity distribution of the background model for the corresponding rectangle. Moreover, height to width ratio of detected rectangles gives information about the classification of detected objects such as: human, truck, or small vehicle.

An example can be seen in Figures 2.5 and 2.6, where the moving pedestrians are eliminated with the improvement in the algorithm.

Figure 2.5: Tracking result without any improvement.

Figure 2.6: Tracking result with histogram information. The moving pedestrians are eliminated

2.4 Results

In this section we present the results from an initial investigation of the proposed algorithm's effectiveness by implementing it on several video recordings taken at different time periods in a day. Images of a two sided highway with resolutions 640 by 480 in Camlica, Istanbul are taken by a stationary camera at 2 different time periods: (CM1) beginning at 12.01 pm and (CM2) at 4.00 pm. The algorithm is implemented with C++ on a PC with 2.20 GHz speed, 2.00GB Ram, and AMD Athlon(tm) 64x2 Dual Core Processor 4200++ under Windows OS. In this platform our implementation processes 1000 frames in 357 seconds. The algorithm runs at 3 fps at real time, with the use of extracted road mask, algorithm improves to 9 fps. To run the algorithm in real time applications, resolution should be reduced to 480 by 320. In this resolution although detection and tracking are handled correctly, occlusion reasoning algorithm has some decision errors. In Table 2.1 we report the number of departing vehicles counted by the proposed algorithm, and by inspection of the video scenes in order to measure the effectiveness of the algorithm.

Table 2.1: Average Counts of Tracked Vehicles.

Duration of Video in Frames	CM1(A)	CM1(I)	error	CM2(A)	CM2(I)	error
1380 (1)	55-44	54-56	1-12	47-31	50-42	3-11
2760 (2)	51-65	54-66	3-1	59-65	60-60	1-5
4140 (3)	43-51	53-56	10-5	37-60	57-70	20-10
5520 (4)	31-56	34-61	3-5	57-53	61-48	4-5
6900 (5)	43-67	50-46	7-21	55-50	65-42	10-8
8280 (6)	53-43	66-52	13-9	58-56	65-46	7-10
9660 (7)	50-65	66-67	16-2	60-62	59-52	1-10
AVG	47-56	54-58	7.6-7.8	53-54	59-51	6.5-8.4
SUM	326-391	377-404	53-55	373-377	417-360	46-59
%error			14-13			11-16
Std			5.65-6.93			6.75-2.5

Table 2.1 gives the number of vehicles departing from the scene in one minute time intervals. During each minute of the video 1380 frames exist. Cameras used in this work operate at 23 frames per second; hence each row of the table corresponds to approximately a one-minute time interval. In the columns, (A) represents the counts calculated by the algorithm and (I) represents the inspection counts. The two numbers in each column are the counts of the number of departing vehicles in both directions of traffic, separately. For example, during the time period CM1, the algorithm counted the number of vehicles departing in both directions during the first minute of the video and found 55 vehicles in the inbound lanes and 44 vehicles in the outbound lanes (Figure 2.4).

In Figure 2.7 a sample algorithm output can be seen, where yellow dots are used for detected vehicles; whereas pink dots are used for tracked but not detected vehicles. Extracted road boundaries are drawn with red lines and the region determined with the purple lines is the ROI that used for tracking. Lastly, white line indicates the boundary of inbound and outbound lanes for counting.

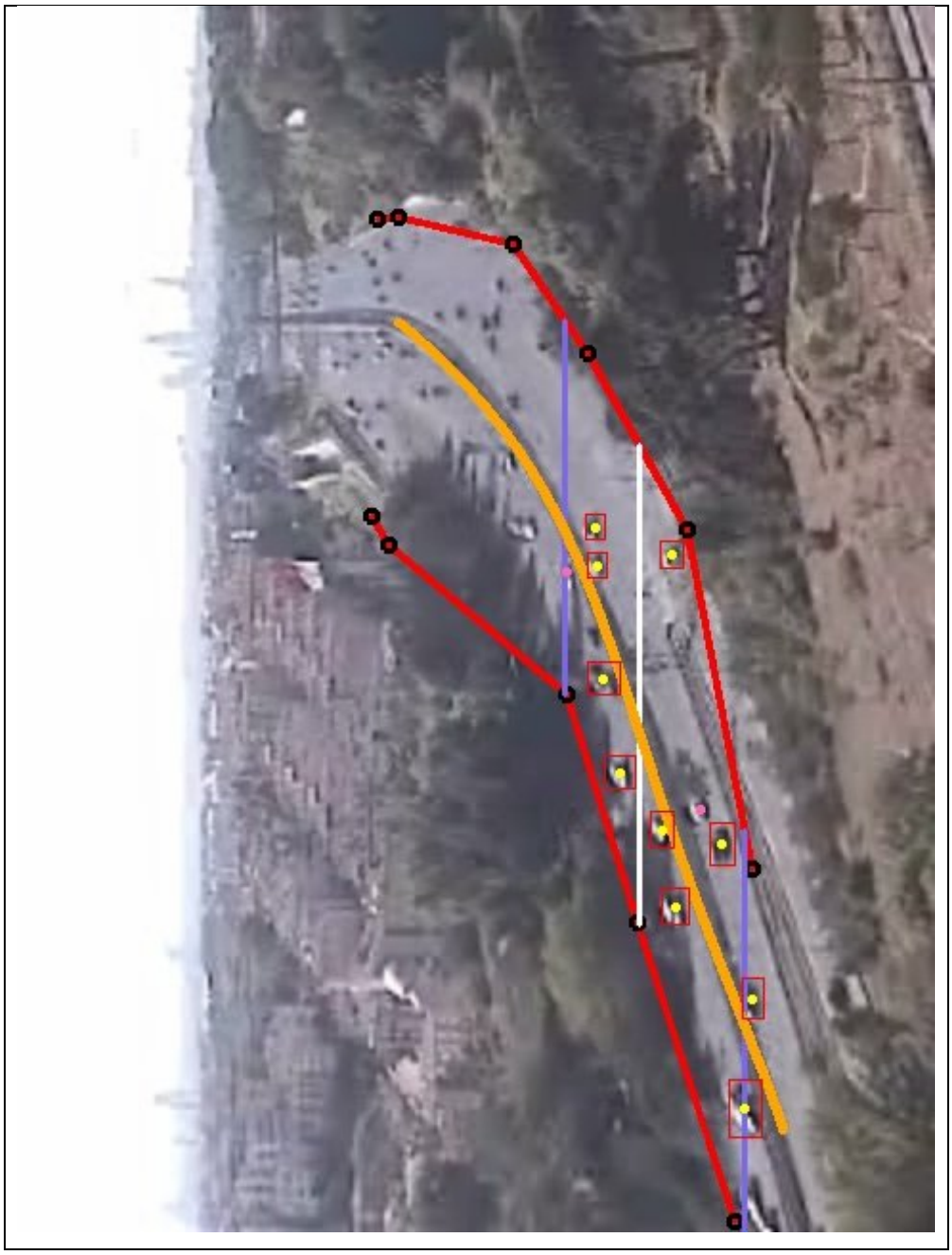


Figure 2.7: A tracking output: Detected vehicles are shown by rectangles while tracked positions are shown by dots.

Figure 2.8: Occlusion decision. Occluded vehicles end rectangle center is represented with green dots.

Figure 2.9: Split decision. Vehicles that are split and their previous position are detected as white dots.

Chapter 3
ROAD AND TRAFFIC ANALYSIS FROM ON-BOARD VIDEO

3.1 Introduction

In order to improve the vehicle safety, extensive studies have been implemented in collision avoidance, lane keeping and road departure warning, vision enhancement, driver condition monitoring and safety-impacting in-vehicle technology integration areas [22]. Different approaches are taken into consideration for describing the motion of a host vehicle for on-board camera systems. A desirable onboard ITS system has to have some crucial features. First of all, it must robustly handle environmental conditions and dynamic changes, i.e. shadow, and illumination changes. Then, it must have a high reliability for safety applications, therefore it has to be robust to vehicle and camera motion, moreover in order to achieve real time processing, system needs faster processing.

In the third chapter of the thesis, we will develop a reliable lane-mark tracking and vehicle-to-vehicle distance calculation scheme that is robust to environmental conditions and dynamical changes. Lane-mark tracking includes the localization of the road, the determination of the relative position between vehicle and road, and the analysis of the vehicle's motion. Vehicle-to-vehicle distance is mainly based on localizing possible vehicles on the host vehicle's path. Robustness is mainly obtained by a lane-mark model with the global motion estimation part as an initialization step. In the remaining part of this chapter, we will give a brief summary of the relevant previous researches, and our contribution. Section 3.2 provides necessary information about the global motion estimation. Section 3.3 describes the line detection procedure. Lane mark model creation is presented in Section 3.4.

Lane tracking and matching algorithm is given in Section 3.5. Section 3.6 outlines the decision fusion procedure, whereas in Section 3.7 system performances and obtained results are presented.

3.1.1 State-of-the-Art

In ITS applications, lane detection is widely used. There have been many applications [22]-[44] that have previously been applied in the literature for lane detection and most of them can be classified in to two groups for onboard camera applications.

In the first group, feature-based methods are used. This group is mainly based on the detection of edges as features in the image, and aggregation of these features in to meaningful models. Detection of edges is mainly handled by Canny edge detection. Most of the previous work use different gradient operators to extract edge information with the magnitude and orientation. Aggregation algorithms are used to combine edge features into meaningful lane structures. The Road Markings Analysis (ROMA) system is based on aggregation of the gradient direction at edge pixels in real-time [22]. To detect edges that are possible markings or road boundaries, it employs a contour following algorithm based on the range of acceptable gradient directions. Authors of the [23] implemented a lane mark detection system that searches pre-defined road regions for horizontal illumination changes. In [24], detection and classification of the lanes are obtained by frequency analysis, where the lane mark type is recognized by Fourier analysis.

In general, feature driven approaches are highly dependent on the methods used to extract features and they suffer from noise effects and irrelevant feature structures. Often in practice the strongest edges are not the road edges, so that the detected edges do not necessarily fit a straight-line or a smoothly varying model. Shadow edges can appear quite strong, highly affecting the line tracking approach. Unlike model-based methods, it is difficult to apply

statistical techniques for lane tracking. However all the characteristics that are present in the image are subject to found.

The second group is the model-based methods. In this group, deformable templates are used to describe a scene with a set of parameters that fits to the scene model. The road boulders and lane markings are often approximated by circular arcs on a flat-ground plane. The authors of the [25] and [26] use a spline-based model that describes the perspective effect of parallel lines, considering simultaneously both-side borders of the road lane. For small to moderate curvatures, a circular arc is approximated by a second-order parabola, where parameters have to be estimated. The estimation can be performed on the image plane [27] or on the ground plane [28] after the appropriate perspective mapping. An extension of the work in [27] is applied detecting arrow signs [29], where the hyperbola-pair lane boundary model is used. The 3D model of the road can also be used in modeling the road parameters through differential equations that relate motion with spatial changes. Such approaches using state variable estimation (Kalman filtering) are developed in [30], where the road model consists of skeletal lines pieced together from clothoids (i.e. arcs with constant curvature change over their run length). A vehicle elimination method is defined in [31], where color information is used to extract lane marks, and size, motion, and shape information is used to robustly detect lane marks in the presence of a similar colored vehicle as the lane mark.

Model-driven approaches provide powerful means for the analysis of road edges and markings. Their main advantage is that the lane can be tracked with a statistical technique, thus, false detections are almost completely avoided. However, the use of a model has certain drawbacks, such as the difficulty in choosing and maintaining an appropriate model for the road structure, the inefficiency in matching complex road structures and the high computational complexity. Comparison of various lane-position detection and tracking techniques can be seen in Table 3.1 obtained from [32].

Moreover, there is a division between methods that treat the lane boundaries as straight lines and methods that consider them as more general curves [33], especially parabolic curves, approximating circular arcs, using a deformable template approach. Both methods assume a parabolic curve in the 3D world. The advantage of assuming a parabolic curve on the road plane rather than in the image plane is that in the Euclidean world, one can assume that all parabolic lane boundaries in the ground plane have approximately the same curvature and have parallel tangents at their x intercepts. The main problem with such an approach is that usually the lane boundary seems to be almost linear at the lower part of the image, and there is only a relatively small area below the horizon where the lane seems to be curved. However, as the image curve is inversely proportional to the *z*-coordinate, it is clear that the area near the horizon is more sensitive to noise. An alternative approach would be to detect the parabolic curves directly on the image plane. However, in such a case, one needs to solve simultaneously for three independent parameters.

Object detection applications have already been investigated in many of the previous works. Different approaches have been implemented according to the isolation method used to extract the moving object from background region. One of the most popular used approaches in object detection is the edge based detection [35]-[42]. In [35], this edge detection is applied to single images to detect the edge structure of even still vehicles. Also, morphological edge-detection algorithms are used in [36] for their superior performance. Alternatively, the edges can be grouped together to form the vehicle's boundary as in [37] and [38]. The algorithm must identify relevant features (often line segments) and define a grouping strategy that allows the identification of feature sets, each of which may correspond to an object of interest (e.g. potential vehicle or road obstacle). Authors of [39] define a system that is based on the search for areas with a high vertical symmetry in multi-resolution images; symmetry is computed using different sized boxes centered on all the columns of the interest areas. All the columns with high symmetry are analyzed to get the width of detected objects. Horizontal edges are examined to find the base of the vehicle in the individuated

area. The aim is to find horizontal lines located below an area with sufficient amount of edges. To improve the shape of object regions [40] and [41] employ the Hough transform to extract consistent contour lines and morphological operations to restore small breaks on the detected contours. In [42], they use color information and shape features based on edges in three directions (horizontal, vertical and diagonal), they build statistical models of vehicle appearance in each feature space. Then, the potential target location is found in each feature space using the mean-shift algorithm.

Table 3.1: Comparison of various lane-position detection and tracking techniques.

System	Use[a]	Road Model	Feature Extraction	Postprocessing	Tracking	Evaluation	Comments
VaMoRs (1992) [16]	A	Clothoid Model with vertical curvature	Edge Elements	eliminates points which are not collinear	Linear vehicle dynamics model	Single frame images	Limited processing power. Simple edge detection fails in difficult situations.
YARF (1995) [33]	A	Circular road segments on flat plane	Hue based segmentation and edge detection	Averaging and linear median squares estimation	Operation on single frame	Positive detection rates for feature extraction, single frame images	Multiple detectors. Limited to yellow and white stripes.
ALVINN (1996) [19], [36]	A	Flat road model for generating training data	Image intensity	Neural Network	None	Road tests, various error measure associated with neural networks	Neural network makes it difficult to decouple control from detection, requires lots of training
RALPH (1996) [25]	A B	Constant curvature on flat plane	scan line matched to template	Template matching to slowly evolving near template and fast evolving far template	No inter-frame tracking described	Single frame images	template methods can fail near construction zone or areas where the road has changed. Shows limited quantitative results
GOLD (1998) [20]	C	Constant lane width on flat plane	Adaptive thresholding of pixel differences	Morphological widening	Operation on single frame	Single frame images	Assumes line markings on dark road, some robustness to lighting and occlusion
LOIS (1998) [34]	B A	Parabolic approximation on flat plane	Edge magnitudes and orientations	Maximum a posteriori estimation evaluated by Metropolis algorithm	Kalman filtering	Error histogram from one drive. Standard deviation of error 13cm	Robust to shadowing in presence of strong lane markings. Otherwise untested.
LANA (1999) [24]	B A	Parabolic approximation on flat plane	DCT coefficients for diagonally dominant edges	Maximum a posteriori estimation	Operation on single frame	Single frame images, comparison to LOIS shown	Only using diagonal DCT coefficients limits detection based on orientation of vehicle
Taylor et al. (1999) [12]	A	Constant curvature on flat plane	Template matching	Hough transform	Kalman Filter input into various control schemes	Performance of controllers shown	Focussed on controller performance. Limited real-world testing.
Ma et al. (2000) [13]	A B C	Circular road model on flat plane	Likelihood based on gradient image	Fusion on radar and optical images	Operation on single frame	Single frame images	Designed for elevated or bordered rural roads.
Southall et al. (2001) [30]	C	Curvature and rate of change of curvature	Threshold both pixel values and cross-correlation to dark-bright-dark function	Factored sampling for particle filter	Particle Filtering via CONDENSATION	Estimates shown for an image sequence, no ground truth or quantitative results	Very limited results and testing. Unclear whether feature extraction will work in difficult situations.
Kwon and Lee (2002) [4], [31]	B	Piecewise linear	multiple "feature transformation modules"	combined with data fusion and constraint satisfaction, heuristic departure warning function	nonlinear filtering	analysis of departure warning system given	Good architecture for sensor fusion. Testing limited to false alarm rate of departure warning.
DARVIN (2002) [5]	A B	DGPS based maps of roads	Image gradient	match to DGPS data	nonlinear filtering	selected frames from experimentation	Directed towards urban driving. Heavy reliance on GPS data.
Lee et al. (2003) [37], [38]	B	Straight road on flat plane	Edge distribution function	Hough transform to extract lanes	Not discussed	Detection rate of lane departure warning	Robust to lighting. Will not work for circular reflectors.
Apostoloff et al. (2003) [29]	C	Not discussed	lane markers, road edge, color, width	Cue scheduling to determine which cues are used	Particle Filtering via Distillation	Success rate, mean absolute error for position, yaw, and road width.	Possibly fail in conditions of strong cues that contradict each other (i.e. fig. 2b)
Kang et al. (2003) [28]	D	Straight road on flat plane	Edge direction and magnitude	Connected-component analysis, Dynamic programming	Single frame operation	Qualitative comparison to hough transform based techniques, Single images shown	Focusses on showing visual comparison to hough transform based technique.
Nedevschi et al. (2004) [22]	D	3D model based on clothoids and roll angle	edge detection	outlier removal based on 3D location found with stereo camera system, roll angle detected	Kalman filtering	single images from road scenes with clearly marked lane boundaries	Simple edge detection not robust to shadows, occlusions
This paper (2004)	C B	Parabolic approximation on flat plane	Steerable filters, adaptive road template	Statistical and motion based outlier removal	Kalman Filtering	Extensive error evaluation described in section V-B	

[a]A = Automated Vehicle Control, B = Lane Departure Warning, C = Driver Assistance, D = Unspecified

3.2 Global Motion Estimation

In dynamic scene analysis, the use of motion models has been proven efficient and convenient in applications such as optic flow computation, motion segmentation, detection of independent moving objects, object tracking, or camera motion estimation. To analyse the dynamic content of the scene, it is important to correctly segment moving objects according to their motion characteristics. In the case of static camera, different algorithms have been efficiently implemented; however in all the works, the solution is devoted to the rather simple situation of a static camera. It may even be reduced to intensity temporal change detection. Hence, in a mobile camera configuration, every pixel may have experience a global motion, a resulted non-compensated background model by [21] would not be adequate or a calculated motion vector field as in [44] would be more complex to classify the moving region. Therefore mentioned algorithms can not be used for mobile camera configuration.

Moreover, to detect the moving objects it there are several requirements. Firstly, the distinction between apparent motion which only depends on movement of the camera should be determined. Secondly, apparent motion, arises from the relative movement of an independently moving object should be determined. In other words, first it is necessary to recover the motion due to camera movement as a global motion, then, to perform detection and tracking. Additionally, in order to consider the dominant motion as the 2D motion due to camera movement, it has to be assumed that the projections of the static components of the scene occupy the main part of the image (with the additional condition that they supply image spatial intensity gradient information [45]), and the projections of moving objects occupy only a minor part of the image [46]. Moreover, in the global motion model, one of the following assumptions is expected to hold. Either the camera is only rotating, or the depth variation in the scene is limited compared to the distance between camera and objects, or the visible surfaces of the static world are approximately located in the same 3D plane such as the planar road in our case. In these cases, it can be assumed that the 2D apparent

motion (due to camera motion) of the static background can be modeled by a 2D parametric motion model and can be considered as the dominant motion. Such a motion model is estimated from frame to frame, and then used in a warping procedure to compute a compensated sequence in which the background is supposed to appear as static. Thus, non-static regions in this sequence can be considered as moving objects.

In this study, to cancel the effect of camera motion on moving objects' apparent motion, a parametric motion model that can estimate global motion under global illumination changes that implemented in [47] is used. To model the global motion Motion2D, which is a multi-platform object-oriented library, is used.

3.2.1 Motion Model

The parametric motion model can be represented with a matrix notation which is linear with respect to the n motion parameters.

$$V_A(p) = \begin{bmatrix} u(p) \\ v(p) \end{bmatrix} = B(p)\,A. \tag{3.1}$$

where p is an image position (x, y) of a point, $V_A(p)$ is the flow vector at point p, B is a matrix whose form depends on the selected parametric motion model, and lastly A is the motion parameter vector that defines the dominant motion of the whole image.

In this study, since affine motion parameters efficiently define motion, and are occasion of low level complexity, 6-parameter affine model is used. This model can cover several kinds of motion such as translation, rotation, scaling, and deformation. It is defined at pixel $p = (x, y)$ as:

$$\begin{bmatrix} u(p) \\ v(p) \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{bmatrix}. \tag{3.2}$$

In this model

$$B(p) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix},$$

and

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^T.$$

For each point *p*, instantaneous temporal variation of the intensity of the moving projected point along its planar trajectory *S* (*p,t*), in other words, total derivative of intensity function *I* is calculated.

$$S(p,t) = \frac{dI}{dt}(p,t). \tag{3.3}$$

This equation can be rewritable partially with respect to spatial gradient $\vec{\nabla} I^t = \left(I_x, I_y\right)$, temporal derivative ($I_t$) of the intensity function, and $\vec{V}^t = \left(dx/dt, dy/dt\right)$ flow vector function.

$$S(p,t) = \frac{dI}{dt}(p,t) = \vec{V}(p,t)\vec{\nabla} I(p,t) + I_t(p,t). \tag{3.4}$$

Normally, one expects the total derivative of intensity function, $S(p,t)$, to be equal to 0, however considering the global illumination changes, $S(p,t)$ is set to a constant, therefore a new variable is introduced to model, and Equation 3.3 becomes,

$$S(p,t) = \frac{dI}{dt}(p,t) = -\Im. \tag{3.5}$$

Applying this equation to 3.4 and calling it the residual function, r_i, yields

$$r_i = \vec{V}(p,t)\vec{\nabla} I(p,t) + I_t(p,t) - S(p,t) \tag{3.6}$$

$$= I_x(p)u(p) + I_y(p)v(p) + I_t(p) + \Im \tag{3.7}$$

Finally replacing $\vec{V}$ with $\vec{V}_A$ result in

$$r_i = \chi\Theta - \gamma, \tag{3.8}$$

where

$$\begin{aligned} \Theta^t &= \left(A^t, \Im\right) \\ \gamma &= -I_t(p) \; and \\ \chi &= \chi\left(p, I_x(p), I_y(p)\right) = \left(\nabla I(p)^t B(p), 1\right). \end{aligned} \tag{3.9}$$

To estimate parameter Θ, error (square of residual) function

$$E\{\Theta\} = \sum_{p \in W} r_i^2 = \sum_{p \in W} \left(\chi\Theta - \gamma \right)^2 \tag{3.10}$$

is minimized according to Θ, where W denotes the whole image in our case since our aim is to detect moving objects. For tracking purposes a ROI can be defined around the moving objects.

$$\hat{\Theta} = \arg\min_{\Theta} \left(E(\Theta) \right). \tag{3.11}$$

3.2.2 Robust Multiresolution Estimation

Since displacement of point *p* is given as

$$\Delta p = V_A(p).\Delta t, \tag{3.12}$$

if the time interval between successive frames is taken 1, then displacement of the point *p* is equal to flow vector $V_A(p)$ at point p. When the displacements between two frames are too large, the motion constraint equation (3.7) is no longer valid. The use of multiresolution algorithm enables to handle such large motions. Consider the model based displaced frame difference:

$$DFD_{\Theta}(p) = I(p + B(p).A, t+1) - I(p,t) + \Im \tag{3.13}$$

To estimate the dominant motion model between successive images I_t and I_{t+1}, a robust gradient based multiresolution estimation [47] is used. For robustness, M-estimation criterion

[49] with a hard-redescending estimator is applied. The goal of the robust estimation is to find the parameter set $\hat{\Theta}$ which best fits the model $M(p, \Theta)$ to observations Y, when data X deviates from the statistical error distribution, meaning when data X is outlier. Thus, the estimated parameter vector is given by:

$$\hat{\Theta} = \arg\min_{\Theta}\left(E(\Theta)\right) = \arg\min_{\Theta} \sum_{p \in W_t} \rho\left(DFD_{\Theta}(p), C\right) \tag{3.14}$$

where the error measure E is formulated as:

$$E(\Theta) = \sum_{p \in W_t} \rho\left(DFD_{\Theta}(p), C\right), \tag{3.15}$$

and $\rho(x)$ is the Tukey, Talwar, Cauchy or Welsh biweight function which is bounded for high values of x and C is a scale parameter to be set. Since our aim is to detect the moving objects, the estimation support W_t consists of the whole image. The minimization is embedded in a multiresolution and incremental scheme based on the Gauss-Newton method. At each incremental step k (at a given resolution level, or from a resolution level to a finer one), we have:

$$\Theta = \hat{\Theta}_k + \Delta\Theta_k \tag{3.16}$$

which also implies that

$$A = \hat{A}_k + \Delta A_k \tag{3.17}$$

$$\mathfrak{I} = \hat{\mathfrak{I}}_k + \Delta\mathfrak{I} \tag{3.18}$$

here $\hat{\Theta}_k$ is the current estimate of the parameter vector Θ. Then, a linearization of $DFD_\Theta(p)$ around $\hat{\Theta}_k$ is performed, leading a residual function $r_{\Delta\Theta_k}(p)$ which is linear with respect to $\Delta\Theta_k$:

$$r_{\Delta\Theta_k}(p) = \vec{\nabla} I_{t+1}\left(p + \vec{V}_{\hat{A}_k}(p)\right).\vec{V}_{\Delta A_k}(p) + \Delta\Im_k + I_{t+1}\left(p + \vec{V}_{\hat{A}_k}(p)\right) - I_t(p) + \hat{\Im}_k \quad (3.19)$$

where $\vec{\nabla} I_{t+1}(p)$ represents the spatial gradient of the intensity function *I* at image position at point *p*, and at time *t+1*. At each step, the increment value $\Delta\Theta_k$ is estimated by minimizing the error function

$$E(\Delta\Theta_k) = \sum_{p \in W_t} \rho\left(r_{\Delta\Theta_k}(p), C\right). \quad (3.20)$$

Since residual function defined in Equation 3.19 is linear with increment value $\Delta\Theta_k$, applying iteratively reweighted least squares (IRLS) method to Equation 3.20 straightforwardly manner will result to convert this M-estimation problem into an equivalent weighted least-squares problem. Since, $\hat{\Theta}_k$ can not be too far from the solution, 0 can be taken as an initial value for the increment value $\Delta\Theta_k$. As a result, minimization of $E(\Theta)$ becomes minimization of Equation 3.20 which is equivalent to minimizing of

$$E(\Delta\Theta_k) = \sum_{p \in W_t} w_i . r^2_{\Delta\Theta_k}(p), \quad (2.21)$$

where

$$w_i = \frac{\psi\left(r_{\Delta\Theta_k}(p)\right)}{r_{\Delta\Theta_k}(p)}. \quad (2.22)$$

This estimation procedure allows getting a robust and accurate estimation of the dominant motion model between two images as seen in Figure 3.1.c, and in our case as background apparent motion. Estimating global motion is the initialization step for the subsequent steps.

One way of obtaining moving regions in a dynamic camera setup is to threshold the motion estimation error. By taking absolute frame difference of current processing image with the back-warped image that obtained from global motion estimation and by thresholding them, we labeled moving regions of the scene according to dominant motion (i.e. background apparent motion) as shown in Figure 3.2. Thresholding is handled in mean sense, that is to say, for every pixel, mean of the 3x3 matrix which is centered at the pixel is subtracted to decide whether the pixel belongs to the background apparent motion or not. The benefit of using local thresholding in mean sense instead of global thresholding is to reduce the global motion estimation errors. Since global motion estimation uses bilinear interpolation, at strong edge regions some errors occur due to the compensation. Therefore, using local thresholding in mean sense instead of using global version helps to reduce this effect. Morphological operations are applied to connect and extend the moving regions. By applying a connected component algorithm with a minimum size constraint and a polynomial fitting algorithm [50], these moving regions are approximates to polygonal curves as seen in Figure 3.3. This helps us to represent moving regions parametrically rather than as an image mask. To initialize a 'Region of Interest' (ROI), and lane mark model parameters we assume that the road lays complementary part of this moving region which satisfies the lane mark model constraints defined in Section 3.4.2.

Figure 3.1:

a) top left: frame at *t*, b) top right: frame at *t*+1, and c) bottom: back-warped image.

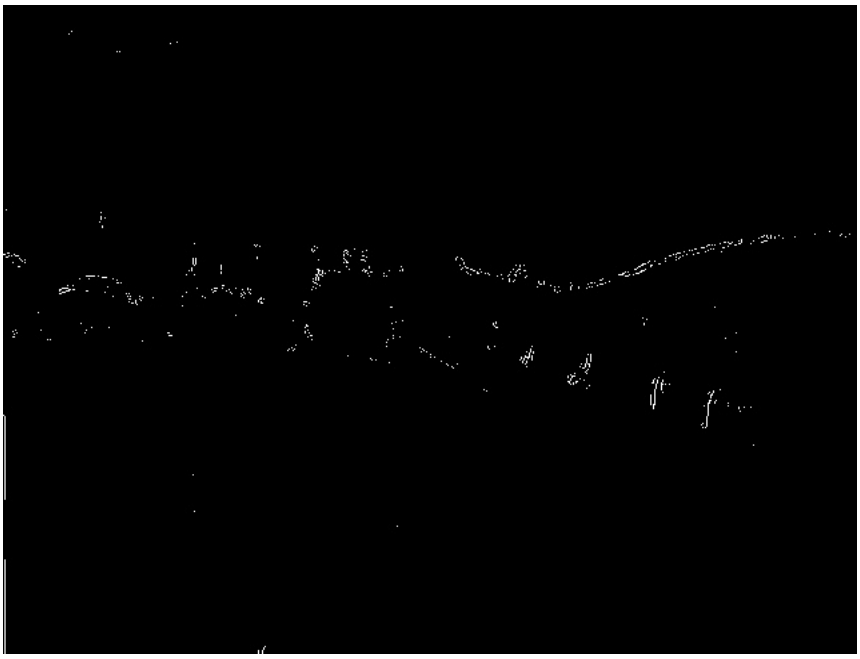


Figure3.2: Thresholding in local mean sense.

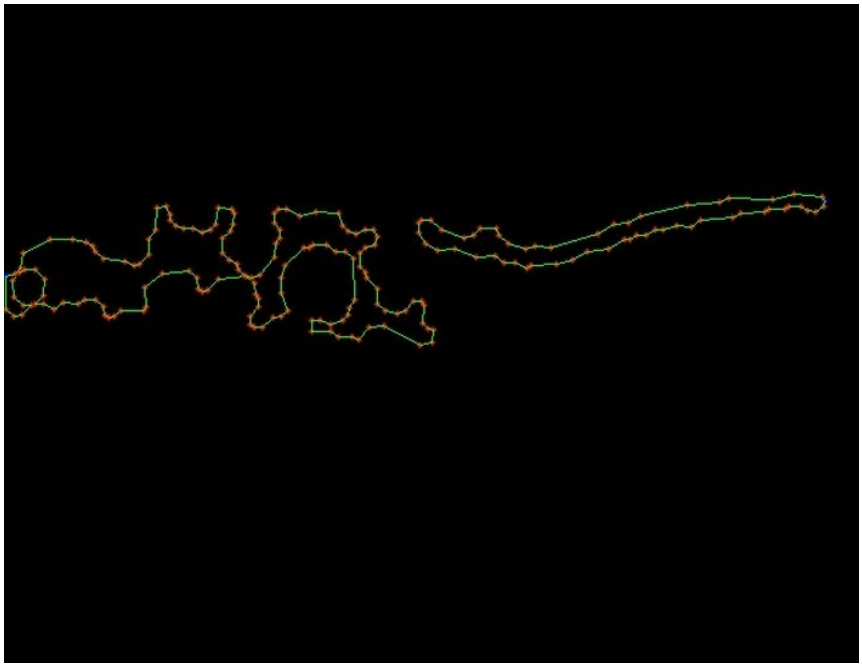


Figure 3.3: Approximation of global motion regions to polynomial curves.

3.3 Line Detection

Line detection is a multi-step procedure that performs consecutive application of Canny edge detection [51] and Hough transform [52]. Strong edges are determined by Canny edge detection method, and then they are used to look for line segments which exist in the whole image by Hough transform.

3.3.1 Canny Edge Detection

The aim of the Canny operator can be stated as good detection (ability to find all edges), good localization (minimal error while positioning edges), and single response for a single edge. To fulfill these objectives, the edge detection process can be summarized as the following stages.

- We want to find the maxima of the partial derivative of the image function I in the direction orthogonal to the edge direction, and to smooth the signal along the edge direction. Thus Canny's operator looks for the maxima of

$$\frac{\partial^2}{\partial n^2}\left(G(x,y) * I(x,y)\right), \tag{3.23}$$

where,

$$n = \frac{\nabla G * I}{\left|\nabla G * I\right|}, \tag{3.24}$$

and

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2 + y^2}{2\sigma^2} \right] \tag{3.25}$$

However, many implementations of the Canny edge detector in fact approximate this process firstly by convolving the image with a Gaussian to smooth the signal, and then looking for maxima in the first partial derivatives of the resulting signal (using masks similar to the Sobel masks). Thus we can convolve the image with 4 masks, looking for horizontal, vertical and diagonal edges. The direction producing the largest result at each pixel point is marked. Record the convolution result and the direction of the edge at each pixel.

- Perform non-maximal suppression. Any gradient value that is not a local peak is set to zero. The edge direction is used in this process.
- Threshold these edges to eliminate `insignificant' edges. Canny introduced the idea of thresholding hysteresis. This involves having two different threshold values, usually the higher threshold being 3 times the lower. Any pixel in an edge list that has a gradient greater than the higher threshold value is classed as a valid edge point. Any pixels connected to these valid edge points that have a gradient value above the lower threshold value are also classed as edge points. That is, once you have started an edge you do not stop it until the gradient on the edge has dropped considerably.

The result of the Canny edge detection algorithm can be seen in Figure 3.5.

3.3.2 Hough Transform

The Hough transform is a standard tool in image analysis which allows recognition of global patterns in an image space by recognition of local patterns (ideally a point) in a transformed parameter space. The basic idea of this technique is to find curves that can be parameterized like straight lines, polynomials, circles, etc., in a suitable parameter space. Duda and Hart

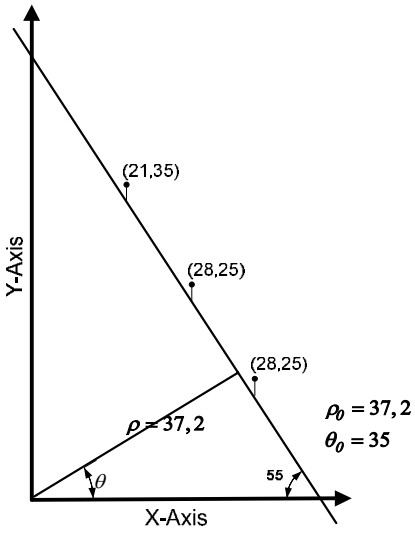
[53] suggested a more convenient way of representing lines, where they use parametric notation. With the length ρ, and the orientation of ρ with respect to x-axis θ all lines can be represented as:

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{3.26}$$

By using the Equation 3.26, mapping all the points on the line in Cartesian coordinate system to polar Hough parameter system results in curves that intersects at a specific polar coordinate ρ and θ. This point to curve mapping is Hough transformation for lines. Representations of a line in Cartesian coordinate and Hough parameter systems can be seen in 3.4.a-b respectively.

Mapped coordinates are recorded in a two-dimensional histogram. The higher count number for a specific bin in this two-dimensional histogram states that, there exists a line whose parameters are ρ and θ that corresponds to the counted bin. To set line length, and to determine end points of the line a search algorithm, which runs from an edge-image pixel to each direction along the found line, is implemented. The contribution of all counts from detected line to two-dimensional histogram is excluded, and the end point selection algorithm runs for other line segments. To reduce search space, [54] introduced a probabilistic Hough transform method that processes only a proportion of the pixels in the edge image. A trade off in the Hough transform is complexity and validity. To reduce complexity probabilistic Hough transform is used. To obtain more valid lines (i.e. no gaps along the line segment) an algorithm runs at the preprocessing step that combine or separate detected lines, and a smaller angle (θ) and a distance (ρ) resolution is set. In Figure 3.6, detected lines are plotted on the real image with white lines.

(a)



(b)

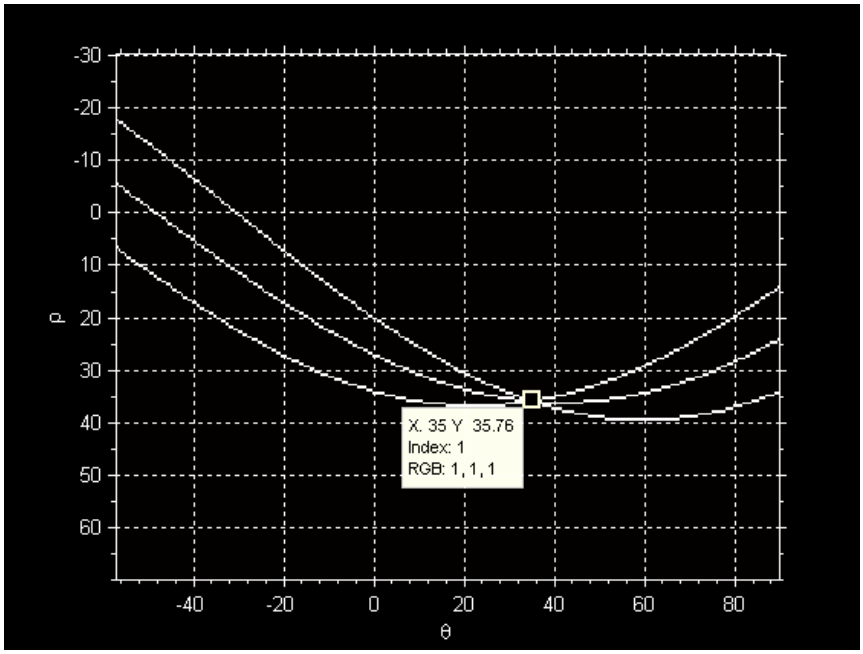


Figure 3.4: a) 3 points on a straight line in the original coordinate plane.

b) 3 curves in the polar Hough parameter plane coinciding at the line parameters (ρ_0) and (θ_0).

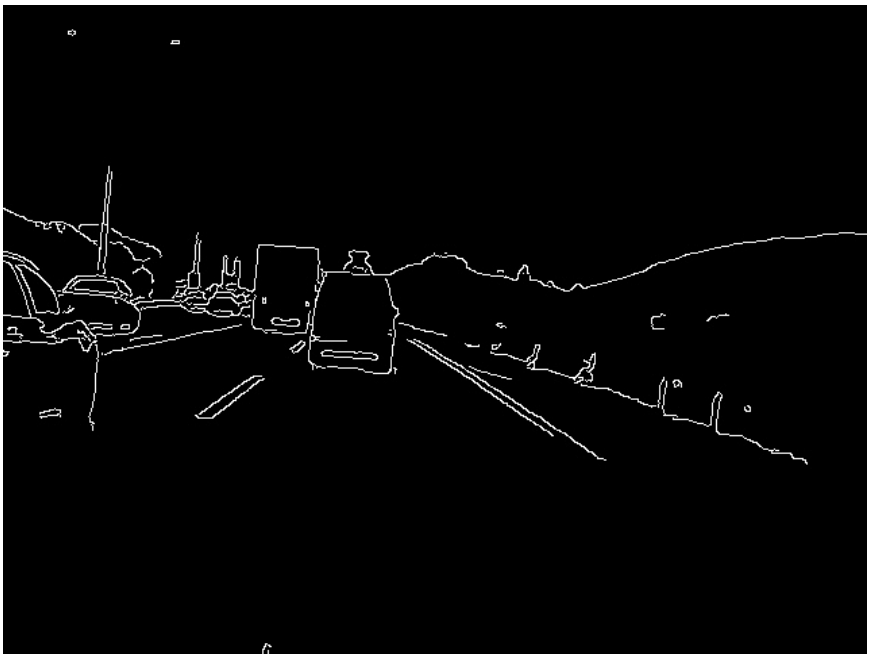


Figure 3.5: Result of the Canny Edge Detection

Figure 3.6: Result of the Hough Transform

3.4 Line Aggregation and Lane Mark Model

3.4.1 Line Aggregation

To track robustly the detected lanes, a preprocessing step in each frame is needed. The purpose of this step is to select the possible lines lying in the ROI determined by the lane mark model parameters which is initialized at the global motion estimation step. Other than ROI, a slope constraint is set, such that the lines having a range of angle from 25 to 85 degrees with the *x*-axis are processed for the lane mark model. Additionally, to reduce combination errors occurred at the line detection step (3.3.2), a classification algorithm that combine or separate interacted lines in the ROI according to lines' positions and alignments is implemented at this preprocessing step. Due to imaging geometry, distance vectors having same magnitudes may imply two different cases. As shown in Figure 3.7, a distance vector ($\vec{d}_{1,2}$) positioned at a higher z-coordinate position may imply two distinct lines belonging to two consecutive collinear lanes, whereas a distance vector at a lower coordinate position ($\vec{d}_{2,3}$)implies that it can be resulted from the same line in a lane mark due to edge detection errors. For this purpose an adaptive decision algorithm is implemented such that distance vectors between consecutive lines having similar slopes are calculated, and the change in the magnitude of these vectors are parameterized with respect to the vectors' positions ($P_{(\bar{x},\bar{y})}$). To parameterize this change a second order polynomial is fitted in a similar way that previously applied in Section 2.2.1.1. Distance vector's position is assumed as the mid-point of consecutive lines' end and start positions:

$$P_{(i,j)} = \left(\frac{x_{i,S} - x_{j,E}}{2}, \frac{y_{i,S} - y_{j,E}}{2} \right), \tag{2.27}$$

where $\|\boldsymbol{i}\|$ and $\|\boldsymbol{j}\|$ are the interacted lines, and are assumed to be lined up at an increasing *z*-direction orientation while deciding lines' start and end points.

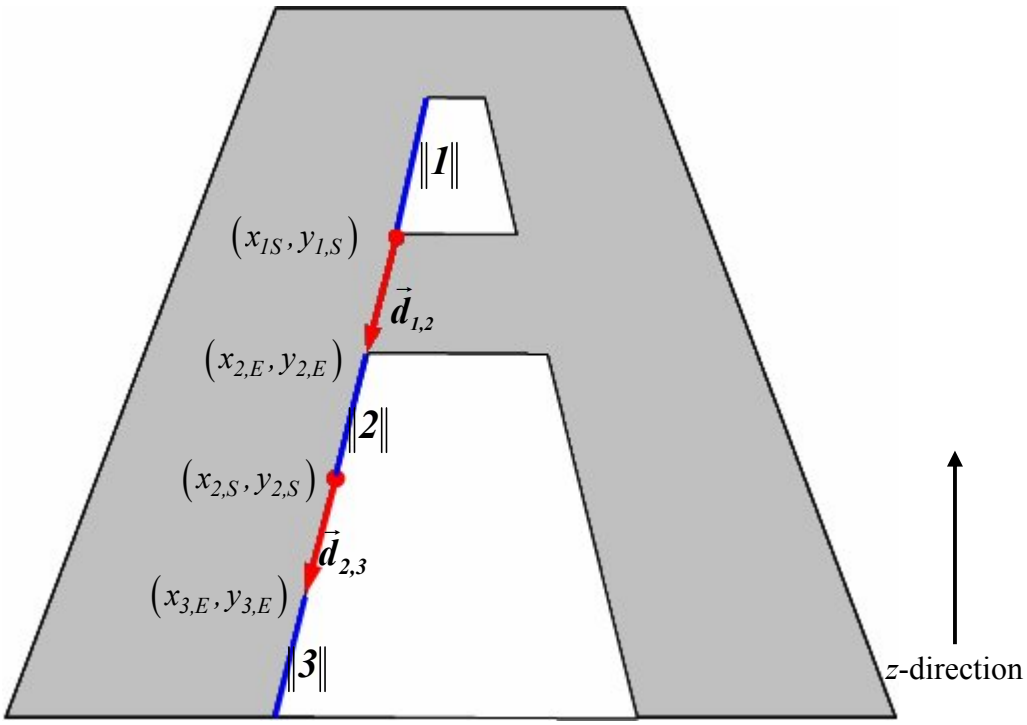


Figure 3.7: Line aggregation algorithm is based on the position of the distance vectors. $\|\boldsymbol{1}\|$ represents the first line, and $\bar{\boldsymbol{d}}_{1,2}$ represents the distance vector between the first and second lines.

3.4.2 Lane Mark Model

In order to robustly track lanes, a lane mark model based on relative positions of interacted lines is defined. This model is based on that, two lines having similar geometrical characteristics, i.e. slope, length, can form a lane mark. First, lines having same slope signs and meaningful shift vectors are compared. Then, triangular inequality criterion is used to form lanes.

To find the magnitude of the shift vector $\vec{\phi}$ between parallel (or nearly parallel) lines we used point to line distance where it is given as:

$$\left|\vec{\phi}\right| = \left|\frac{a.x_0 + b - y_0}{a^2 + 1}\right|. \tag{2.28}$$

where, 'a' is the slope of the line, 'b' is the line constant parameter, '$\vec{v}$' is the line normal, and (x_0, y_0) is the coordinate of the point.

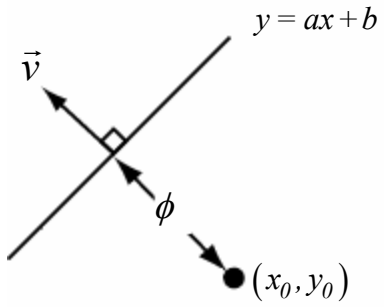


Figure 3.8: Point to line distance.

Distance between lines 'end' and 'start' positions is calculated for each interacted lines. The condition to decide whether two lines form a lane mark or they are consecutive lines

belonging to consecutive lanes is their lengths. As represented in the left side of the Figure 3.9, it is expected that the lines in a lane mark obey the triangular inequality, such that:

$$\left|\vec{l}_2\right| + \left|\vec{l}_3\right| > \left|\vec{D}_{2,3}\right|, \tag{2.29}$$

and the lines belonging to consecutive lanes do not obey the above inequality, such that:

$$\left|\vec{l}_1\right| + \left|\vec{l}_3\right| < \left|\vec{D}_{1,3}\right|. \tag{2.30}$$

In this equation $\left|\vec{l}_i\right|$ represents the length of the i^{th} line '$\|\boldsymbol{i}\|$', and $\left|\vec{\boldsymbol{D}}_{\mathbf{i,j}}\right|$ represents the distance between the end-point of i^{th} line and the start-point of j^{th} line.

Obtained lanes are represented with its center line and this line's slope. Center line is calculated from the side (left and right) lines by averaging line coordinates.

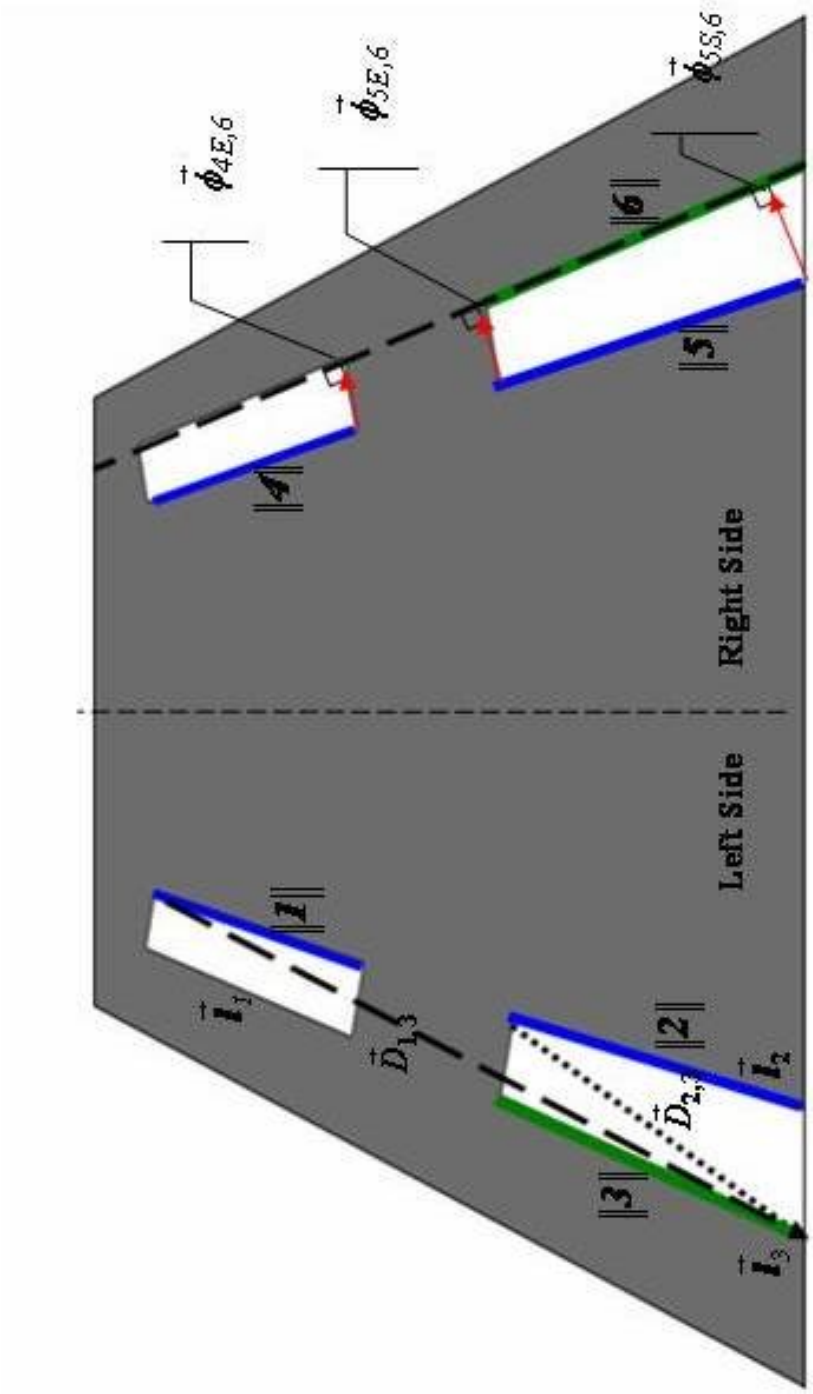


Figure 3.9:
Left Side: Lane decision algorithm is based on the triangular inequality.
Right Side: |∅| is distance between a point and a line trajectory.

3.5 Lane Mark Tracking and Matching

The purpose of the previous section is to obtain lanes that satisfy lane mark constraints in each frame. Tracking of these lanes is essential to describe the motion of the host vehicle and easier than the tracking of arbitrary lines that scatter across the image.

For each lane marks in the left and right sides of the road, four Kalman filters is initialized for the corners of the lane marks in a similar way that explained in Chapter1. Each lane mark is tracked with their left and right (end, start) points as shown in Figure 3.10. This way, a robust tracking system that can handle unwanted cases such as, temporary changes or errors due to misclassification or noise is being implemented.

In each frame, matching is based on the lane mark centers x_C, y_C of the predicted lane mark with the observed lane mark for each side of the road. Matched lanes' parameters, i.e. left and right (start, end) points, are updated, and other parameters are recalculated, i.e. lane slope. Finally, Kalman filters of unmatched lanes are updated with their prediction values.

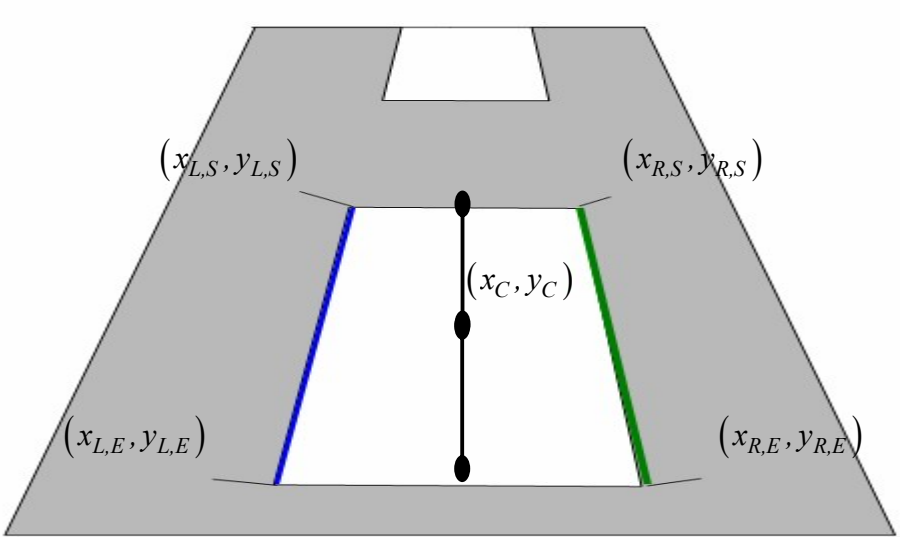


Figure 3.10: Lane mark model. Each lane mark is composed of 2 similar, i.e. slope, and length, line and represented with their mean values.

3.6 Decision Fusion

Tracking lanes in both sides of the road gives some important information about the scene and motion. First, relative motion of a vehicle can be estimated from the lane mark model parameters, i.e. overtake. Second, the relative distance of a vehicle to left or right lane mark can be calculated from the lane mark angles with respect to *x*-axis. For a non-rotating camera setup where camera is mounted nearly at the center of vehicle width (or mounted at a known position), it is possible to estimate the position of a vehicle by comparing image center with the estimated road center along the *x*-axis. Also, relative position of the vehicle on the road can be determined from the lane mark lengths. A higher length value for the right lane mark implies that vehicle is traveling at the nearest lane to the road boundary. Moreover, to reduce search space, a 'Tracking Zone' (TZ) is created by intersecting the ROI with the area that lies between left and right lanes. Lastly, vehicle-to-vehicle distance is calculated by founding the nearest lines that belongs to the TZ, by searching from the bottom of the image to vanishing point (intersection point of lanes).

Two modes are created to analysis vehicle motion: i) in-lane mode and, ii) overtake mode. In the in-lane mode, vehicle to vehicle distance is processing, and in each frame, it is checked that whether overtake mode is started or not. Vehicle to vehicle distance is implemented in this mode. In the overtake mode, vehicle motion is defined according to lane mark slopes, and checked whether end of overtake is reached or not.

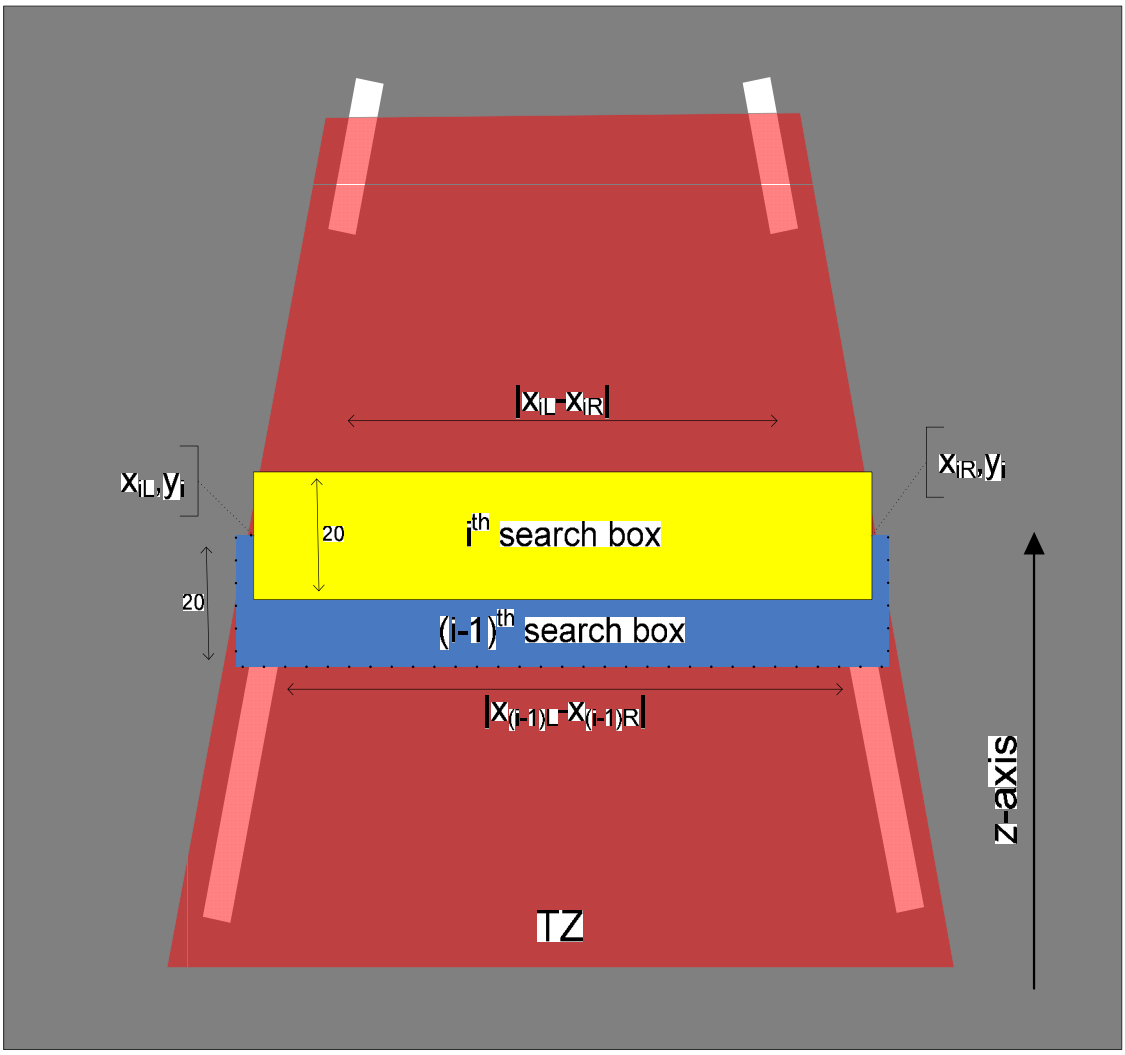


Figure 3.11: Vehicle to vehicle distance is performed through the *z*-axis.

3.6.1 In-Lane Mode

In this mode, vehicle to vehicle distance is calculated by weighted line count for a floating box that scans the TZ of image through the *z*-direction. Since it is expected to obtain more edges at more textured regions, counting of parallel lines with respect to lateral in the ROI gives information about transition from road (relatively more flat, end constant intensity region) to vehicle. To eliminate the effect of unexpected edge occurrences along the *z*-direction in the TZ, count of the vehicles is weighted for the i^{th} box with a weighting parameter w_i where

$$w_i = \frac{1}{|x_{iL} - x_{iR}|}. \tag{3.31}$$

Here x_{iL} and x_{iR} represents the left and right coordinates of the TZ for the i^{th} box location. By weighting the counts with a higher weight parameter, small line occurrences are not taken into account.

3.6.2 Passing Mode

This mode is used to decide when the vehicle starts to overtake, and when it finishes passing. To do such a work, after initialization with global motion estimation, the slopes of left and right lanes are recorded. By comparing lane mark slopes in each frame with the initial slope, whether a vehicle is traveling between the lanes (in-lane mode) or passing, can be determined. Start of left-passing is set to '+'10 degrees of difference between initial left slope with the left lane mark's slope. Similarly, end of left-passing is found if the angle difference is smaller than 5 degrees. On the contrary, for right-passing same procedure is applied for a difference sign taken as '-'.

3.7 Results

Images of a high way with 3 lanes in Istanbul are used to test proposed algorithm. Images were obtained using an onboard camera with input image resolutions of 640 by 480. Algorithms are implemented in C++, and experiments are conducted on a PC with AMD Athlon(tm) 64x2 Dual Core Processor 4200++ , 2.20 GHz speed, and 2.00 GB Ram under Windows OS. Under these conditions, implementation process 300 frames over 33 seconds.

In the experimental studies, test criterions are set to vehicle to vehicle distance, and host vehicle positioning. Test set is composed of 250 frames, and 150 of them are detected as in passing mode. The rest are labeled as in-lane mode, and vehicle to vehicle distance is calculated on these images.

3.7.1 Vehicle to vehicle distance calculation

Since counting with floating box is performed for fixed segments along *z*-direction, distance is measured in multiples of 10 pixels. The leading vehicle is detected by observing the first floating box whose corresponding line count is higher than an occupancy threshold. Search for such a box is performed from the bottom of the image to vanishing point in TZ as mentioned in Section 3.6.1. Table 3.2 presents the calculated distance errors in box segments with respect to inspection values. Numbers in the columns indicate the displacement errors, and signs represent the direction of these errors, i.e. ‘-1’ indicates, an estimation error of 1 segment (10 pixel) nearer from the leading vehicle. Increase in the error is mostly resulted from line detection step. Previously mentioned assumption in Section 3.6.1 - increase in edge numbers at transition from road to vehicle along the *z*-direction, is not satisfied for these frames. ‘+’ error for 2 and 3 is mostly resulted from failure of this assumption. Algorithm founds a further floating box instead of the inspected one.

Table 3.2: Distance calculation performance of the proposed algorithm.

	Distance From the Inspection Values						
Total Frame Number	**Exact match**	**1**		**2**		**3**	
		+	-	+	-	+	-
99	76	8	8	4	-	3	-

3.7.2 Decision of Passing and Vehicle Localization

To obtain the relative position of the host vehicle on the image, positions of the left and right lanes are used over 100 frames. Since camera parameters are not present, rather than real position, relative position of the estimated road center is compared with the inspection values. The root mean square error (RMSE) of a frame sequence, consisting of 45 frames in in-lane mode, is found as 8,0854 pixels, whereas with the addition of 55 frames in passing mode this error increase to 12,003 pixels. Here, the RMSE is taken as:

$$\mathrm{RMSE}(x,y)=\sqrt{\frac{\sum_{i=1}^{n}(x_i-y_i)}{n}} \tag{3.32}$$

where the observation count n is taken as 45 and 100 for the two RMSE calculation and the estimated lane mark position x, and inspection values y are represented as

$$x=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix},\ \text{and}\ y=\begin{bmatrix}y_1\\y_2\\\vdots\\y_n\end{bmatrix}.$$

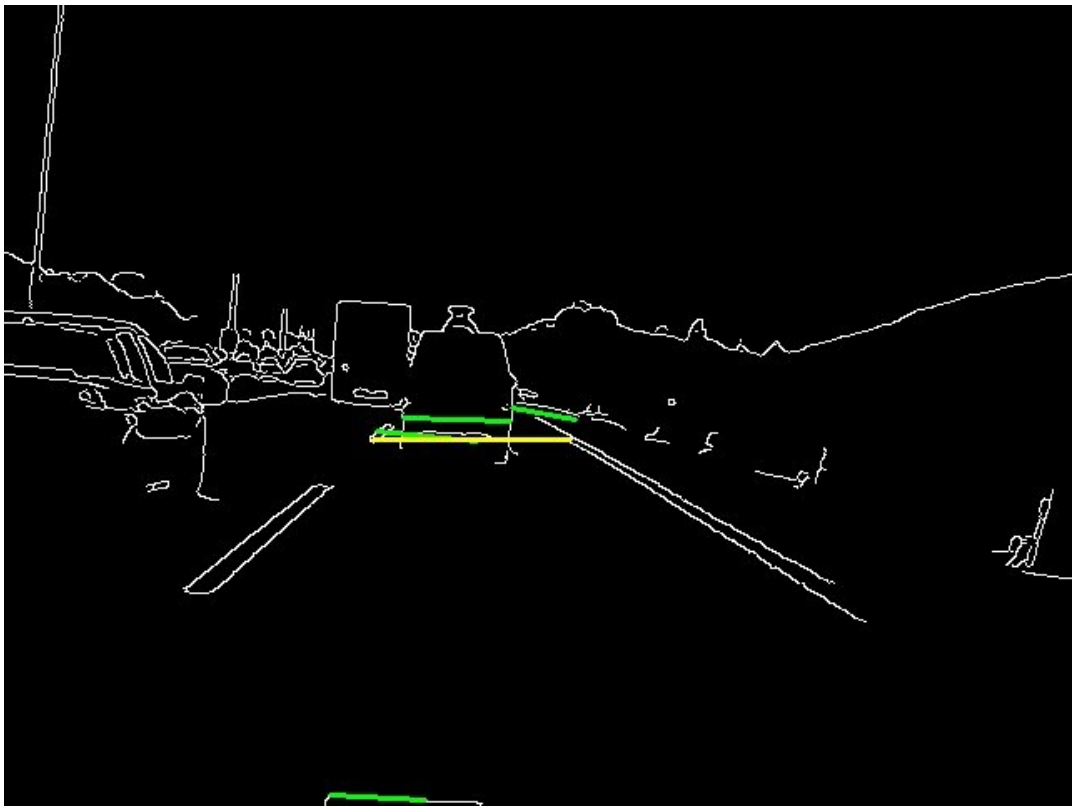


Figure 3.12: Result of the distance algorithm. Yellow line indicates the leading vehicle position, where green lines indicate the founded horizontal lines in the TZ.

In Figure 3.13, in order to visualize the position of the host vehicle, a horizontal magenta bar having a length of the estimated road width at that location is placed at the bottom of the image. Centers of the road and image are marked with red and green dots consecutively. To set a reference, initial position of the road center is represented with a black line. Green dot above the image represents the mode, i.e. green for in-lane, red for passing.

Figure 3.13: Frame number 43.

Initial frame of the sequence is shown in Figure 3.14, and overall algorithm results are shown in Figure 3.15, where yellow and red lines are right and left lane mark's boundaries consequently, and black line between them represents the lane mark model. Red dot at the top of the image shows the passing mode, and black dot represents the side of the passing, i.e. 'left' or 'right'. For the below example, passing is 'left'.

Figure 3.14: Initial Frame

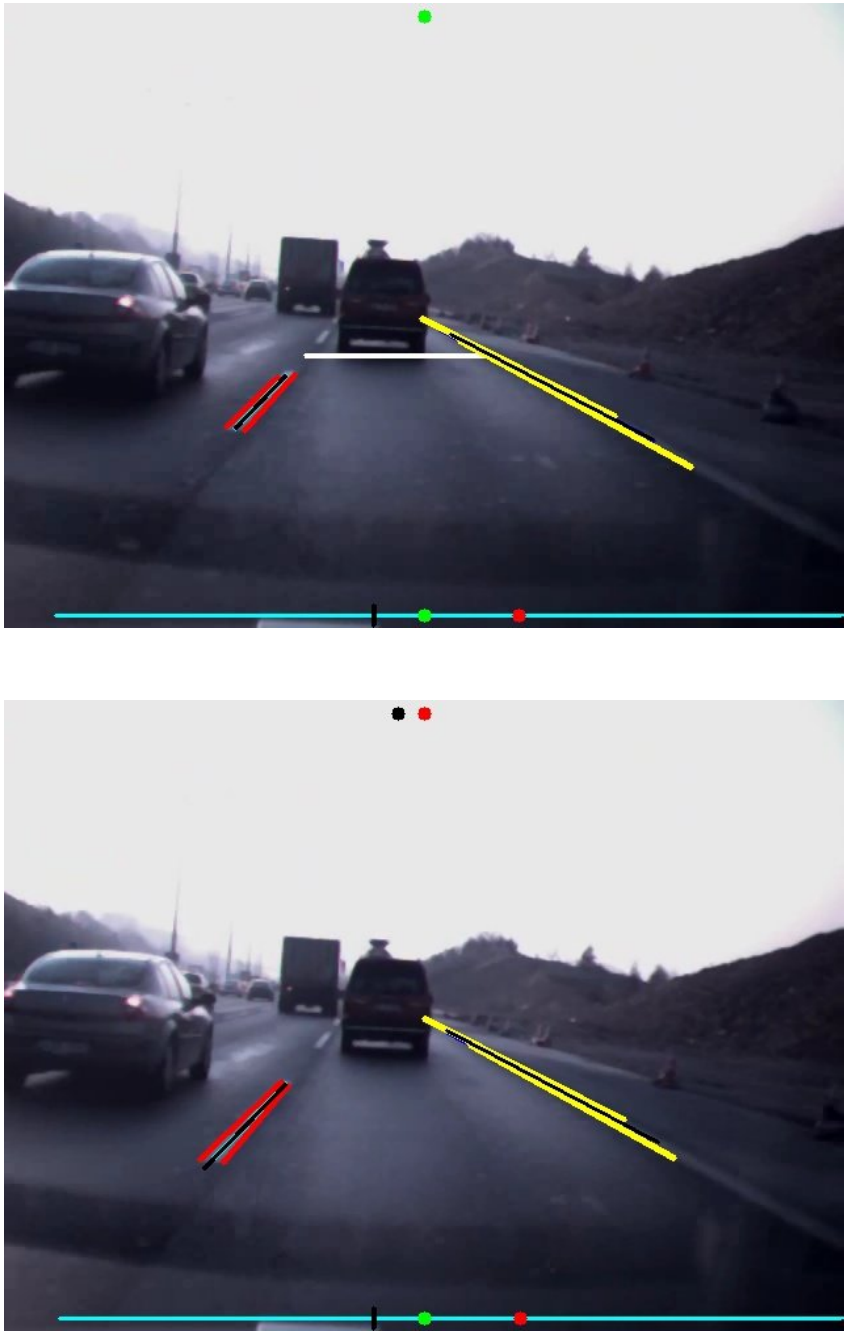


Figure 3.15:

a) Frame 43, before the start of passing

b) Frame 44, the start of passing.

Chapter 4

CONCLUSION

In both traffic monitoring, and automatic driver warning systems, mainly two major tasks of perception is pointed: i) estimation of the road geometry and ii) vehicle and obstacle detection. In the traffic monitoring systems, different methods have been proposed in the previous studies, but there have been few studies considering the road geometry as in this thesis. Temporal frame differencing and background modeling algorithms are implemented for foreground extraction in the literature. Although temporal differencing algorithms are well suited real-time applications, most of the previous implemented applications use probabilistic models to the model background. The reason for this choice is that adaptation to environmental changes such as illumination variations is necessary for outdoor applications. As a result, the second point to question is the trade off between the computational complexity and efficiency of the model used in the detection systems. Another important issue is the feature dependency of the proposed algorithms, where nearly all model based application proposed in literature based on the valid feature selection. Image oriented, discriminant and reliable feature sets such as texture, color, edge, etc.., are extensively used in the detection and tracking systems to analyse the scene. Previously proposed systems handle most of the problems such as occlusion. However, most of them are implemented for close-up recordings.

After considering the outlined issues above, in this study we performed a feasibility study of a road monitoring system where adaptive bounding box size is used to detect and track vehicles according to their estimated distance from the camera. Automatically detected road

mask is used to improve system performance, and an occlusion reasoning algorithm is implemented based on the vehicle motion. The cameras used in this study are positioned at a high location which enables a good view of the road; however it is not sufficient to implement a model-based recognition algorithm for the objects on the road due to the low quality of videos.

Another crucial issue that we mainly focused in this thesis study is the system performance under different environmental conditions such as night time recordings and snowy weather. Firstly, video recordings including different time intervals in a day are used to test overall system performance. Secondly, night time images are used to test under extreme illumination variations, and snowy surrounding images are used for detection under different weather conditions. It has to be noted that cameras used in this work operate 23 frames per second and our overall performance is tested for 7 minutes of video recordings that are taken at 2 different time periods, and image resolution is 640 by 480. Winter recordings are obtained from the web site of "The Fakultät für Informatik". Video resolution used in the weather condition tests is 768 by 576 and frame rate is 30 frame-per-seconds.

The results we obtained in this study illustrate that, the implemented algorithm works robustly under different environmental conditions, and proposed bounding size strategy fit the vehicles well for monitoring applications. However, it is seen that, the tracking algorithm underestimates the count of the vehicles in general; the intuition behind this fact is that for long durations occluded vehicles are assigned as a single vehicle. By extending the tracking time of vehicles, these results can be further improved. The reason for the high difference in algorithm and inspection results at 5^{th} and 7^{th} intervals of CM1 is due to the environment conditions. The proposed algorithm is implemented for stationary setups but cameras are affected from environmental factors such as wind. Moreover, since proposed model forces to assign a Kalman filter to every new bounding box (rectangle) and keeps the other objects' Kalman filters, problems with tracking mostly occur due to foreground segmentation. For example, in over 4500 frames there are only 4 tracking errors due to the nature of the

adaptive bounding box size in CM1; small objects in high threshold regions are omitted while implementing connected-component analysis. On the other hand, 29 vehicles' tracking is lost only in the left lane of the highway in 4500 frames in CM2. This is the result of assigning foreground regions to background due to the high similarity between the objects and the background modeling for low quality images. Moreover, counting strategy needs modifications, since false detections yield the estimated count values higher than the original inspection values in some intervals. To prevent this shortcoming, counting can be performed for the whole image or in a large box, rather than a small region. But this approach requires the knowledge of counted vehicles history, since counted vehicles must be excluded from the total count to prevent recount. Because every vehicle is assigned with a Kalman filter, the tracking of a vehicle along the road is possible for the proposed algorithm, but tracking and recording the history of all vehicles in a scene will increase the computational complexity.

To reduce the computational complexity, extracted road mask is used to determine modeling region. Addition of the road mask improves the speed of the system by a multiple of three (9 fps). Moreover, by reducing the resolution to 480x320, we obtain near real time operation (22 fps), but there occurs some errors in occlusion reasoning algorithm at this resolution.

In the third chapter of the thesis, we try to point out the trade off between the two major methods, feature and model based, which are commonly used in automatic driver warning applications. To compensate feature extraction errors, an innovative two-step aggregation method that reduces the false detections is proposed in our study. The first step is the reliable aggregation of lines into meaningful lane components. Then the second step aggregation is used for lane mark formation. Moreover, a partial tracking approach is followed for host vehicle motion analyse, where lane markings are tracked rather than the whole lane. Lastly, a new initialization step is proposed in this work that eliminates the global motion to detect planar regions on the scene. We assume that the road is planar, and belongs to the global motion estimated region.

To test the proposed automatic warning application, a video data base obtained for drive-safe [55] applications is used. In our study, video sequence composed of 250 frames is used to validate the proposed system. Processed video is recorded at 29 fps, and having resolution 640 by 480. Test criteria are the relative distance of the inspected road center to algorithm result. Proposed algorithm operates in image plane and the distances are calculated in pixels. Calculations based on the relative distances. Moreover, we assume that the error in road center estimation is proportional to the error in vehicle localization. The RMSE of the proposed algorithm is calculated for both in-lane and passing mode. The results show that proposed algorithm can also estimate the position of the vehicle with in a % 1.25 error margin. In order to represent camera orientation, initial position of the estimated road center is marked with black color in Figure 3.13. Further improvements can be made on the proposed system with the introduction of camera parameters, where yaw and pitch angle can be considered, too.

Proposed driver warning system could not be used for real time applications currently. Therefore, to reduce the computational complexity, single frame approach is maintained. Despite the correctly performed detection, tracking errors occur due to the constant velocity assumption used in Kalman filter states. Constant acceleration model is used for state variables to prevent this problem, however the tracking errors were observed in this model, too. With the introduction of camera parameters, proposed warning system can be implemented for the two planes accordingly. Thence, robust detection can be performed in the image plane, and a more adaptive tracking can be performed in the road plane. Also using constant acceleration model with the speed data that is obtained by the tachometer enables more accurate tracking of lane marks. These are the issues should be further investigated.

After the localization test, another test is applied for the vehicle to vehicle distance calculation on the proposed system. In this test, a host vehicle traveling between the lanes is tracked for 100 frames, and distance between the host and targeted vehicle is calculated. Then, algorithm results are compared with the inspection values. Since the proposed method

calculates the distance in terms of segments, results are tabulated with respect to error in segment numbers. It is seen from the test that the algorithm tends to make (+) errors; meaning estimated vehicle position is further away from the targeted vehicle. The reason of obtaining (+) error is mainly due to the line detection inaccuracy. Although algorithm directed to find more edges and lines by setting a smaller resolution at edge detection step, and using smaller accumulator threshold for Hough transform consequently, the observation is not sufficient at the road to vehicle transition. To prevent this shortcoming, algorithm can be extended to a more complex one by adding new features, such as histograms. One possible solution is the histogram calculation in the floating box. Since flat road assumption is valid contemporarily, the histogram information can be used to detect road to vehicle transitions by detecting peak values in the histogram of the box.

BIBLIOGRAPHY

[1] C. Little, "The intelligent vehicle initiative: Advancing 'Human-Centered' smart vehicles," Public Roads Mag., vol. 61, no. 2, pp. 18-25, Sept./Oct. 1997.

[2] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles", Proceedings of the IEEE, vol. 90, iss. 7, pp.1258- 1271, July 2002.

[3] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications", IVC, no. 4, pp. 359-381, April 2003.

[4] http://tkm.ibb.gov.tr/kameraAna.aspx

[5] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," IEEE Trans. on Intelligent Transportation Systems, vol. 1, no. 2, pp. 119-130, Jun. 2000.

[6] B. Li and R. Chellappa, "A generic approach to simultaneous tracking and verification in video," IEEE Trans. on Image Processing, vol. 11, no. 5, pp. 530-544, May 2002.

[7] K. P. Karmann and A. von Brandt, "Moving object recognition using an adaptive background memory" Time-Varying Image Processing and Moving Object Recognition, The Netherlands, 1990.

[8] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, “A real-time computer vision system for measuring traffic parameters,” in Proc. of IEEE Conf. Computer Vision and Pattern Recognition, pp. 496-501, Puerto Rico, June 1997.

[9] Jong-Ho Choi, Kang-Ho Lee, Kuk-Chan Cha, Jun-Sik Kwon, Dong-Wook Kim, and Ho-Keun Song, “Vehicle tracking using template matching based on feature points”, Information Reuse and Integration, IEEE Conf. pp. 573 – 577, Sept. 2006.

[10] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, “Moving target classification and tracking from real-time video,” in Proc. IEEE Workshop Appl. Comput. Vis., pp. 8–14, 1998.

[11] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, “Detection and classification of vehicles,” IEEE Trans. Intell. Transp. Syst., vol. 3, no. 1, pp. 37–47, Mar. 2002.

[12] K. Baker and G. Sullivan, “Performance assessment of model-based tracking” Proc. of the IEEE Workshop on Applications of Computer Vision, Palm Springs, CA, pp. 28-35, 1992.

[13] Z. Kim and J. Malik, “High-Quality vehicle trajectory generation from video data based on vehicle detection and description”, Proc. IEEE Intelligent Transportation Systems Conference, pp. 176-182, 2003.

[14] Lei Xie, Guangxi Zhu, Yuqi Wang, Haixiang Xu, and Zhenming Zhang, ”Real-time vehicles tracking based on Kalman filter in a video-based ITS”, Communications, Circuits and Systems, 2005.

[15] J.W. Hsieh, S.-Hao, Y.S. Chen, and W.F. Hu, “Automatic traffic surveillance system for vehicle tracking and classification” , Proc. IEEE Intelligent Transportation Systems Conference, vol. 7, pp. 175-187, June 2006.

[16] C. Gentile, O. Camps, and M. Sznaier, “Segmentation for robust tracking in the presence of severe occlusion”, IEEE Transactions On Image Processing, vol. 13, no. 2, February 2004.

[17] Y. Sugaya and K. Kanatani, “Outlier removal for motion tracking by subspace separation”, 8th Symposium on Sensing via Image Information (SSII2002), Yokohama, Japan, pp. 603 608, July 2002.

[18] H.T. Nguyen and A.W.M. Smeulders, “Fast occluded object tracking by a robust appearance filter”, In Pattern Analysis and Machine Intelligence, IEEE Transactions, pp. 1099- 1104, August 2004.

[19] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, “Occlusion robust tracking utilizing spatio-temporal Markov random field model”, International Conference on Pattern Recognition, ICPR, Barcelona, vol. 1, pp. 142-147, 2000.

[20] H. Jun-Wei, Y. Shih-Hao, C. Yung-Sheng, and H. Wen-Fong, “Automatic traffic surveillance system for vehicle tracking and classification”, IEEE Trans. Intell. Transp. Syst, vol. 7, no. 2, June 2006.

[21] C. Stauffer and W.E.L. Grimson. “Adaptive background mixture models for real-time tracking” IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99) - Volume 2 p. 2246, 1999.

[22] W. Enkelmann, G. Struck, and J. Geisler, "ROMA—a system for model - based analysis of road markings", Proceedings of IEEE Intelligent Vehicles, Detroit, pp. 356–360, 1995.

[23] D.J. LeBlanc, G.E. Johnson, P.J.T. Venhovens, G. Gerber, R. DeSonia, R. Ervin, C.F. Lin, A.G. Ulsoy, and T.E. Pilutti, "A road departure prevention system", IEEE Control Systems December, 1996.

[24] J. Collado, C. Hilario, A. de la Escalera, and J.M. Armingol, "Detection and classification of road lanes with a frequency analysis", Proc. IEEE Intelligent Vehicles Symposium, pp.78-83, 6-8 June 2005.

[25] Z. Kim, "Realtime lane tracking of curved local road", Proc. IEEE Intelligent Transportation Systems Conference, pp. 1149-1155, 2006.

[26] Y. Wang, D. Shen, and E.K. Teoh, "Lane detection using spline model", Pattern Recognition Letters no.21, pp. 677–689, 1994.

[27] K. Kluge and S. Lakshmanan, "A deformable-template approach to lane detection", IEEE Proceedings of Intelligent Vehicles, pp.54-59, 1995.

[28] C. Kreucher and S. Lakshmanan, "LANA: a lane extraction algorithm that uses frequency domain features", IEEE Trans. on Robotics and Automation no. 15, p. 54–59, 1999.

[29] Z. Wennan, C. Qiang and W. Hong, “Lane detection in some complex conditions”, Intelligent Robots and Systems, pp. 117–122, Oct. 2006.

[30] E.D. Dickmanns, B. Mysliwetz and T. Christians, “An integrated spatiotemporal approach to automatic visual guidance of autonomous vehicles”, IEEE Transactions on Systems, Man, and Cybernetics 1990.

[31] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng and K.-C. Fan, “Lane detection with moving vehicles in the traffic scenes”, IEEE Trans. Intelligent Transportation Systems, vol. 7, Issue 4, pp. 571-582, Dec. 2006.

[32] J C. McCall and M.M. Trivedi, “Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation”, IEEE Trans. on Intelligent Transportation Systems, vol. 7, pp. 20-37, 2006.

[33] D. Schreiber,B. Alefs and M. Clabian, “Single camera lane detection and tracking”, Proc. IEEE Intelligent Transportation Systems, pp. 302-307, 2005.

[34] L. Guo, J.Wang and K. Li, “Lane keeping system based on THASV-II Platform”, IEEE Vehiccular Electraonics and Safety, pp. 305-308, Dec. 2006.

[35] K. Shimizu and N. Shigehara, “Image processing system used cameras for vehicle surveillance”, IEE International Conference on Road Traffic Monitoring, pp. 61–65, Feb. 1989.

[36] M. Bertozzi and A. Broggi, “GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection”, IEEE Trans. on Image Processing, vol.7, 1998.

[37] M. Xie, L. Trassoudaine, J. Alizon, and J. Gallice, “Road obstacle detection and tracking by an active and intelligent sensing strategy,” Machine Vision and Applications vol. 7 pp.165-177, 1994.

[38] E.D. Dickmanns, “Vehicle guidance by computer vision”, Concise Encyclopedia of Traffic and Transportation Systems, pp. 597-602, 1991.

[39] A. Broggi, P. Cerri, and P.C. Antonello, “Multi-resolution vehicle detection using artificial vision”, IEEE Intelligent Vehicles Symposium, 2004.

[40] X. Li, Z.-Q. Liu, and K.-M. Leung, “Detection of vehicles from traffic scenes using fuzzy integrals”, Pattern Recognition, pp 967-980, 2002.

[41] H. Moon, R. Chellapa, and A. Rosenfeld, “Performance analysis of a simple vehicle detection algorithm”, Image and Vision Computing, pp. 1-13, 2002.

[42] K. She, G. Bebis, G. Haisong, and R. Miller, “Vehicle tracking using on-line fusion of color and shape features”, IEEE Proc. Intelligent Transportation Systems, pp. 731-736, Oct. 2004.

[43] J. M.Ferryman, S. J. Maybank, and A. D. Worrall, “Visual surveillance for moving vehicles”, Proc. IEEE Workshop on Visual Surveillance, 1998.

[44] K. P. Karmann, A. V. Brandt, and R. Gerl, “Moving object segmentation based on adaptive reference images”, Proc. Fifth European Signal Process. Conf, pp. 951 954, Sep. 1990.

[45] F. Spindler and P. Bouthemy, “Real time estimation of dominant motion in underwater video images for dynamic positioning”, IEEE Int. Conf. Robotics and Automation, pp. 1063-1068, 1998.

[46] R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, “Hierarchical model-based motion estimation”, European Conference on Computer Vision, pp. 237-252, 1992.

[47] J.-M. Odobez and P. Bouthemy, “Robust multiresolution estimation of parametric motion models in complex image sequences”, Proc. 7th European Conf. on Signal Processing, Sept. 1994.

[48] http://www.irisa.fr/vista/Motion2D/about.html

[49] P.J. Hubert and Robust statistics, Wiley, 1981.

[50] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", The Canadian Cartographer, pp. 112-122, 1973.

[51] J.Canny, “A computational approach to edge detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, no. 6, Nov 1986.

[52] P.V.C. Hough, “Machine analysis of bubble chamber pictures”, International Conference on High Energy Accelerators and Instrumentation, CERN, 1959.

[53] R. Duda and P.Hart , “Use of the Hough transform to detect lines and circles in pictures”, Communication of the ACM, vol. 15, pp 11-15, 1972.

[54] N. Kiryati ,Y. Eldar and A.M. Bruckstein , “A probabilistic Hough transform”, Pattern Recognition vol.24, pp. 303-316, 1991.

[55] DRIVE-SAFE: Signal Processing and Advance Information Technologies for Improving Driver/Driving Prudence and Accident Reduction

VITA

ERHAN BAS was born in Nevsehir, Turkey on August 2, 1982. He received his B.Sc. degree in Electrical and Electronics Engineering and his double-major in Physics from Middle East Technical University, Ankara, Turkey, in 2005. From August 2005 to August 2007, he worked as a teaching and research assistant in Koç University, Istanbul, Turkey. At Koç University, he focused on Automatic Vehicle Detection and Counting on Static Camera, and Traffic and Road Analysis on On-board Camera which have been supported by IBB and DPT consequently. He has submitted a paper about Traffic Flow Analysis and Vehicle Counting in the ITS 2007 (Istanbul, Turkey).