UNIVERSITY OF CALIFORNIA, SAN DIEGO

Human Attention and Intent Analysis
using Robust Visual Cues in a Bayesian Framework

A dissertation

submitted in partial satisfaction of the requirements for the degree

Doctor of Philosophy

in

Electrical Engineering

(Intelligent Systems, Robotics, and Control)

by

Joel Curtis McCall

Committee in charge:

Professor Mohan M. Trivedi, Chair
Professor James Hollan
Professor Kenneth Kreutz-Delgado
Professor David Kriegman
Professor Bhaskar Rao

2006

The dissertation of Joel Curtis McCall is approved, and it is acceptable in quality and form for publication on microfilm:

____

____

____

____

____
Chair

University of California, San Diego

2006

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Professor Mohan Trivedi for the enthusiasm, support, motivation, and insight that he has given to me unreserved from the beginning of my graduate studies. I would also like to thank the members of my committee, Professors James Hollan, Kenneth Kreutz-Delgado, David Kriegman, and Bhaskar Rao, for their advice and constructive criticism.

I am also very grateful for the love and support of my wife, Jill Griggs McCall. Without her encouragement, none of this would be possible. I would also like to thank my parents, Judith and Timothy McCall, and my brother, Patrick McCall, for their love and advice.

I would also like to acknowledge the members of the Computer Vision and Robotics Research (CVRR) Laboratory for their assistance in setting up and conducting experiments, discussions and insight relevant to this thesis, and overall scholarship and camaraderie. Specifically I would like to thank Mr. Ofer Achler for his assistance in the development of the Intelligent Vehicle Test bed that provided the basis for the data collection and experimentation contained in this thesis, Dr. Tarak Gandhi for useful discussions and insight into various aspects of intelligent vehicles, Dr. Douglas Fidaleo for his insight and feedback concerning facial affect recognition, Dr. Dario Salvucci for his insight into predicting driver behavior and intent, Mr. Shinko Cheng, Ms. Junwen Wu, Mr. Stephen Krotosky, Mr. Erik Murphy-Chutorian for their help in data collection as well as many useful discussions.

The text of Chapter II, in part, is a reprint of the material as it appears in: J. M^{c}Call and M. M. Trivedi, “Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, March 2006. I was the primary researcher of the cited material and the co-author listed in this publication directed and supervised the research which forms a basis for this chapter.

The text of Chapter III, in part, is a reprint of the material as it appears in: J. M^{c}Call and M. M. Trivedi, “Facial Action Coding Using Multiple Visual Cues and a Hierarchy of Particle Filters,” in *Proceedings of IEEE Workshop on Vision for Human Computer Interaction in Conjunction with CVPR 2006*, New York, New York, June 2006, and J. M^{c}Call and M. M. Trivedi, “Pose invariant affect analysis using thin-plate splines,” in *Proceedings of International Conference on Pattern Recognition*, August 2004, pp. 958–964. I was the primary researcher of the cited materials and the co-author listed in these publication directed and supervised the research which forms a basis for this chapter.

The text of Chapter IV, in part, is a reprint of the material as it appears in: J. M^{c}Call and M. M. Trivedi, “Driver Behavior and Situation Aware Brake Assistance for Intelligent Vehicles,” in *Proceedings of the IEEE*, submitted for review, and J. M^{c}Call, D. Wipf, M. M. Trivedi, and B. Rao, “Lane change intent analysis using robust operators and sparse bayesian learning,” *IEEE Transactions on Intelligent Transportation Systems*, in press. I was the primary researcher of the cited materials.

The text of Appendix A, in part, is a reprint of the material as it appears in: J. M^{c}Call, O. Achler, and M. M. Trivedi, “Design of an Instrumented Vehicle Testbed for Developing Human Centered Driver Support System,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004. I was the primary researcher of the cited materials.

This research was sponsored by UC Discovery Grants, digital media innovations and Nissan Motor Co., Ltd.

VITA

1977	Born, Long Beach, California
1999	Bachelor of Science, University of California, Berkeley
1999–2001	Hardware Engineer, Cadence Design Systems, Inc.
2001–2002	Teaching Assistant, Department of Electrical and Computer Engineering University of California, San Diego
2003	Master of Science, University of California, San Diego
2002–2006	Research Assistant, University of California, San Diego
2006	Doctor of Philosophy University of California, San Diego

PUBLICATIONS

J. M^{c}Call and M. M. Trivedi, "Driver Behavior and Situation Aware Brake Assistance for Intelligent Vehicles," in *Proceedings of the IEEE*, submitted for review.

M. M. Trivedi, T. Gandhi, and J. M^{c}Call, "Looking-In and Looking-Out of a Vehicle: Selected Investigations in Computer Vision based Enhanced Vehicle Safety," *IEEE Transactions on Intelligent Transportation Systems*, in press.

J. M^{c}Call and M. M. Trivedi, "Facial Action Coding Using Multiple Visual Cues and a Hierarchy of Particle Filters," in *Proceedings of IEEE Workshop on Vision for Human Computer Interaction in Conjunction with CVPR 2006*, New York, New York, June 2006.

J. M^{c}Call and M. M. Trivedi, "Human behavior based predictive brake assistance," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, June 2006.

J. M^{c}Call, D. Wipf, M. M. Trivedi, and B. Rao, "Lane change intent analysis using robust operators and sparse Bayesian learning," *IEEE Transactions on Intelligent Transportation Systems*, in press.

J. M^{c}Call and M. M. Trivedi, "Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, March 2006.

J. M^{c}Call, D. Wipf, M. M. Trivedi, and B. Rao, "Lane change intent analysis using robust operators and sparse Bayesian learning," in *IEEE International Workshop on Machine Vision for Intelligent Vehicles*, June 2005.

J. M^{c}Call and M. M. Trivedi, “Performance evaluation of a vision based lane tracker designed for driver assistance systems,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, June 2005.

J. M^{c}Call, O. Achler, M. M. Trivedi, P. Fastrez, D. Forster, J. B. Haue, J. Hollan, and E. Boer, “A collaborative approach for human-centered driver assistance systems,” in *Proceedings of the 7th IEEE Conference on Intelligent Transportation Systems*, October 2004, pp. 663–667.

J. M^{c}Call and M. M. Trivedi, “Visual context capture and analysis for driver attention monitoring,” in *Proceedings of IEEE Conference on Intelligent Transportation Systems*, October 2004, pp. 332–337.

J. M^{c}Call and M. M. Trivedi, “Pose invariant affect analysis using thin-plate splines,” in *Proceedings of International Conference on Pattern Recognition*, August 2004, pp. 958–964.

J. M^{c}Call and M. M. Trivedi, “An integrated, robust approach to lane marking detection and lane tracking,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004, pp. 533–537.

J. M^{c}Call, O. Achler, and M. M. Trivedi, “Design of an instrumented vehicle testbed for developing human centered driver support system,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004, pp. 483 – 488.

J. M^{c}Call, S. Mallick, and M. M. Trivedi, “Real-time driver affect analysis and televiewing system,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2003, pp. 372–377.

ABSTRACT OF THE DISSERTATION

Human Attention and Intent Analysis

using Robust Visual Cues in a Bayesian Framework

by

Joel Curtis McCall

Doctor of Philosophy in Electrical Engineering

(Intelligent Systems, Robotics, and Control)

University of California, San Diego, 2006

Professor Mohan M. Trivedi, Chair

Human-computer interaction is beginning to permeate all aspects of our lives. As we develop more and more interactive computer systems, it is important to develop proper tools and frameworks for making this interaction more efficient. These types of interactive systems can also increase the safety of everyday tasks. We propose a human attention and intent analysis system based on a probabilistic framework using visual cues to help increase the efficiency and safety of everyday tasks. We will specifically look at the driving task. By combining cues about the vehicle interior and driver, the vehicle state, and the vehicle surround, we can make estimates of the driver's focus of attention and intent. Research contributions will be made in the areas of head pose and facial affect analysis, lane detection and tracking, fusing multiple cues to generate estimates of both attention and intent, and overall system integration. A Bayesian framework allows us to effectively combine multiple modalities of cues from visual information as well as other sensors to generate estimates which take into account the uncertainty of both the observations and the underlying process as well as prior knowledge about the parameters we are estimating. This can help us assess critical situations and feedback information faster and more efficiently than systems that do not take into account the driver's attention or intent. We will also show statistical results demonstrating the accuracy of such a system in real-world conditions using data collected from a 28 different drivers.

Chapter I

Introduction

As computers become more and more pervasive in our lives, it is increasingly important to develop tools for improved interaction between man and machine. One tool useful for such interaction is attention and intent analysis. The ability to judge human state and attention gives a computer an invaluable input for assessing the importance of various pieces of information as well as proper ways to display that information. Predicting intent can help make computers more efficient and, as we will discuss next, safer. One important area of research where more and more human-computer interaction is being created is Intelligent Vehicles. Safety systems that interact efficiently and predictively have the potential to make a skill we perform daily much safer.

Far too often we are inclined to multitask while driving. This can include finishing our morning hygiene regiment, carrying out a cell phone conversation, or just becoming lost in thought. This driving task becomes so automated and routine in our minds that often we put ourselves on "autopilot" and have difficulty even remembering the details of our drives. This inattention is quite evident in statistical studies of vehicle accidents. Inattention was listed as a contributing factor in 35% of an estimated 700,000 single vehicle off-roadway crashes; 65% of an estimated 300,000 rear-end collisions; and 50% of an estimated 100,000 lane-change related collisions [1].

The repetitive nature of the driving task can also lead to reduced driver vigilance and drowsiness. Aside from driver inattention, these types of situations can also lead to errors in perception. Studies of crashes involving lane changes and lane merges have shown that 91% of these types of crashes are due to perceptual errors [2]. This can include a misjudgement of an approaching vehicles speed, misjudgement of a gap size between vehicles, or looking but failing to see a vehicle.

Based on these collision statistics, it is important to take into account the driver's attentive state and intentions in determining the criticality of the driver's situation. Knowing this information has the potential to generate earlier predictions of hazardous situations with greater accuracy [3, 4]. For intended lane changes, predicting the driver's intent can allow for an earlier warning in certain critical situations such a risky lane changes. The driver's attentive state is also coupled tightly with the driver's intent. For example, by knowing that a driver is intending to perform a lane change, we can infer that the driver's attention is directed towards the lane to which he is changing. If potentially critical situations arise in areas in which the driver is not focused, it might be necessary to bring these areas to the driver's attention. Monitoring driver attention can also help to predict unintended actions. Vehicle maneuvers which deviate from the predicted driver's intentions or occur while the driver is in an inattentive state might require a warning to be given to the driver.

It is therefore the focus of this dissertation to develop a system for predicting driver attention and intent using a probabilistic framework that can incorporate necessary cues from a variety of sensor modalities. By incorporating a wide variety of cues taken from rectilinear cameras, omnidirectional cameras, near-infrared cameras, vehicle sensors, and global positioning system (GPS) sensors, this system has the potential to provide robust and accurate predictions of the driver's attentive state and intentions. The specific cues we will be examining for this system fall into one of three categories: Vehicle interior, vehicle state, and vehicle exterior. Cues obtained from the vehicle's interior can give us a wealth of information about the driver and passenger's in the vehicle. Head pose, facial affects, physiological information, and body movements are important factors in predicting attention and intent [3, 5]. The vehicle state is also equally important to capture. Steering and pedal corrections can provide information about the driver's attention to the driving task [6]. Vehicle state information is also important for predicting the vehicle trajectory. Information from the vehicles exterior such as lane trajectories and obstacle maps are important for assessing critical situations and events which may require the driver's attention.

I.A Problem Statement and Challenges

It is the focus of this dissertation to explore new algorithms and techniques for analyzing human attention and intent. Because of the enormous amount of activities and intents that we can partake in, for this dissertation we will focus on those associated with the driving task. Specifically, we will be exploring the use of a variety of robust visual cues for inferring driver attention and intent. Placed into a Bayesian framework, we can infer the most probable attentive state or intended action. The driving environment can be divided into the vehicle surround, the vehicle state, and the vehicle interior. Each of these areas are important for attention and intent monitoring and each present their own challenges in inferring driver attention and intent.

Inside the vehicle we are focused mainly on the driver and occupant activities. Visual cues are extremely useful for inferring driver activities and from this intent, but challenges such as sudden changes in lighting, hard shadows, camera jitter, and occlusions make the analysis more difficult. To help solve these problems, a variety of robust visual cues fused together to generate the most probable description of a scene are can be used. Near infrared (IR) cameras with IR illumination can help in low light situations, but they preclude the use of color-based cues.

Outside the vehicle we are focused on the localization of the vehicle with respect to the lane boundaries and other vehicles. Variations in lighting, environment and weather conditions, variations in road surfaces and markings, and occlusions pose the largest challenges. Again these challenges can be mitigated using robust visual cues and a probabilistic framework that allows for measurement error.

Along with challenges arising from visual cues, challenges posed from variations in driver behavior also add to the difficulties in analyzing driver attention and intent. Specifically, different driver can perform the same action in different ways, making generalization more difficult. These variations have an impact on the sensing systems both inside and outside the vehicle. To help overcome this problem, large databases of natural driving behavior must be created and analyzed. As we will discuss in section A.C, the data collection itself presents its own challenges.

I.B Research Contributions

This dissertation presents research contributions in three key domains: the concept domain, the algorithm domain, and the system architecture domain. In the concept domain, we present a framework for inferring human attention and intent and specifically apply this to the driving scenario. In conjunction with this, we also present a framework for lane detection and tracking robust to real-world driving conditions. A framework for facial feature and landmark tracking based on an hierarchy of particle filters is also presented.

In the algorithm domain, key contributions include: algorithms for robust road feature extraction, algorithms for fusion of multiple cues for facial feature tracking, algorithms for inferring driver attention and intent, and algorithms for identifying critical situations based on real-world data.

In the system architectural domain, key contributions include: a holistic system for driver assistance, a test bed for real-world data collection, new metrics and experiments for driver attention and intent analysis based on real-world data.

I.C Outline of the Dissertation

This dissertation is separated into chapters based on the various components of the overall driver assistance system we are presenting. Chapter II introduces a system for robust road marking detection and tracking. Experimental results are shown that demonstrate the system performance in a wide variety of environmental conditions including different daylight conditions, atmospheric conditions, road conditions, and traffic conditions. Chapter III explores driver affect analysis and the low level processing necessary to create a robust facial feature tracking and analysis system. We show results based on various databases as well as in-vehicle analysis. Chapter IV presents the overlying architecture for driver intent analysis. In this chapter we also provide a wide variety of performance evaluations based on receiver-operator-characteristics (ROC) curves for a variety of driving behaviors. Each aspect of the system is trained and evaluated using an extensive database containing real-world driving situations. Appendix A discusses requirements for and design of an intelligent vehicle test bed.

Chapter II

Vehicle and Road Localization

II.A Introduction

Within the last few years, research into intelligent vehicles has expanded into applications which work with or for the human user. Human factors research is merging with intelligent vehicle technology to create a new generation of driver assistance systems that go beyond automated control systems by attempting to work in harmony with a human operator. Lane position determination is an important component of these new applications. Systems which monitor the driver's state [7], predict driver intent [8, 9], warn drivers of lane departures [10], and/or assist in vehicle guidance [11, 12] are all emerging [13]. With such a wide variety of system objectives, it is important that we examine how lane position is detected and measure performance with relevant metrics in a variety of environmental conditions.

There are three major objectives of this chapter. The first is to present a framework for comparative discussion and development of lane detection and position estimation algorithms. The second is to present the novel “Video Based Lane Estimation and Tracking” (VioLET) system designed for driver assistance. The third is to present a detailed evaluation of the VioLET system by performing an extensive set of experiments using an instrumented vehicle testbed. To this end the paper is arranged in the following manner. In section II.B we will first explore the system objectives, environmental variations, and sensing modalities involved in creating a lane position tracker. In section II.C we will introduce a common framework for lane position tracking systems which we will use to provide comparisons between existing systems based on the objectives, conditions, and sensing systems described in the introduction. Next, in section II.D, we will present the VioLET system, a lane position detection and tracking system with its design based upon a driver assistance system for use in a highway road environment. Finally, in section II.E, we will evaluate the VioLET system with both (a) a wide variety of performance metrics which are relevant to the system objectives and (b) a wide range of environmental variations and driving contexts.

The contributions of this research extend to five areas:

1. The introduction of a fully integrated lane estimation and tracking system with specific applicability to driver assistance objectives. By working closely with human factors groups to determine their needs for lane detection and tracking we developed a lane tracking system for objectives such as driver intent inferencing [3] and behavioral analysis [14].

2. The introduction of steerable filters for robust and accurate lane marking extraction. As will be described in section II.D, steerable filters provide an efficient method for detecting circular reflector markings, solid-line markings, and segmented-line markings under varying lighting and road conditions. They help to provide robustness to complex shadowing, lighting changes from overpasses and tunnels, and road surface variations. Steerable filters are efficient for lane marking extraction because by computing only three separable convolutions we can extract a wide variety of lane markings.

3. The incorporation of visual cues (lane markings and lane texture) and vehicle state information to help generate robust estimates of lane curvature as described in section II.D.3. By using the vehicle state information to detect instantaneous road curvature, we can detect curvature in situations where roadway lookahead is limited.

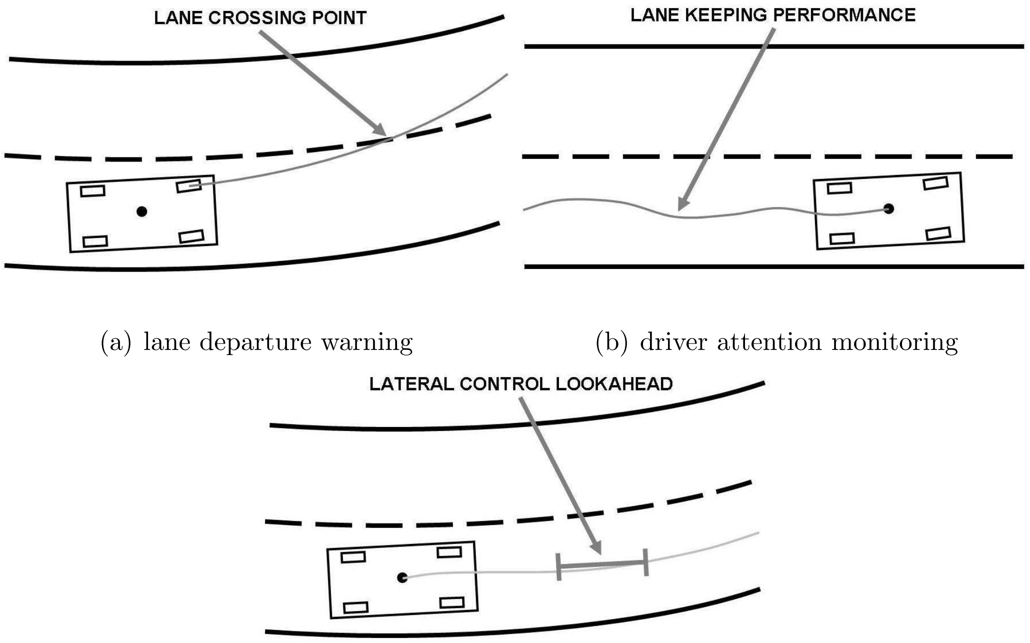
4. The experiment design and evaluation of the VioLET system. This experimentation was performed using multiple quantitative metrics over a wide variety of test conditions on a large test path using a unique instrumented vehicle. We also present a justification for our choice of metrics based on our work with human factors applications as well as extensive ground-truthed testing from different times of day, road conditions, weather, and driving scenarios.

5. The presentation of an up-to-date and comprehensive analysis of the current state-of-the-art in lane detection research. We present a comparison of a

wide variety of methods, pointing out the similarities and differences between methods as well as for what objectives and environmental conditions various methods are most useful.

II.B Lane Position Detection: Objectives, Environments, and Sensors

II.B.1 System Objectives



(a) lane departure warning

(b) driver attention monitoring

(c) vehicle control

Figure II.1: Illustrations of systems which require lane position and key performance metrics associated with the system objectives.

In this chapter we will look at three main objectives of lane position detection algorithms as illustrated in figure II.1. These three objectives and their distinguishing characteristics are:

- Lane Departure Warning Systems

  For a lane departure warning system, it is important to accurately predict the trajectory of the vehicle with respect to the lane boundary. [15, 16]

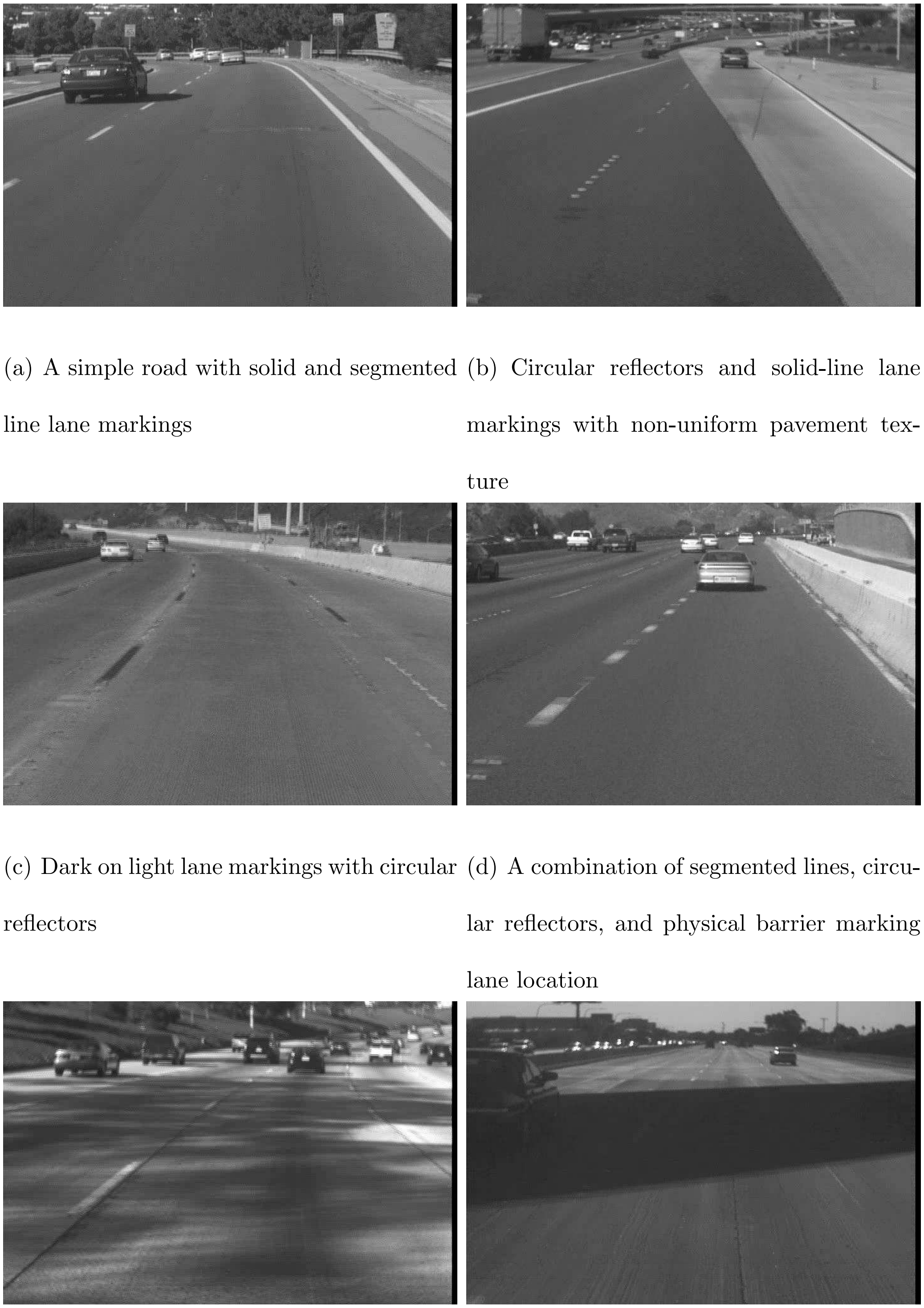
- Driver Attention Monitoring Systems

  For a driver attention monitoring system, it is important to monitor the drivers attentiveness to the lane keeping task. Measures such as the smoothness of the lane following are important for such monitoring tasks. [7]

- Automated Vehicle Control Systems

  For a vehicle control system, it might be required that the lateral position error at a specific lookahead distance, as shown in figure II.1c, be bounded so that the vehicle is not in danger of colliding with any objects. [17]

For each objective it is important to examine the role that the lane position sensors and algorithms will take in the system and design the system accordingly. Also, evaluation of these sensors and algorithms must be performed using the proper metrics. Components of lane position sensors and algorithms that work well for certain objectives and situations might not necessarily work well in others. Examples of these situations will be shown in section II.C.



(a) A simple road with solid and segmented line lane markings

(b) Circular reflectors and solid-line lane markings with non-uniform pavement texture

(c) Dark on light lane markings with circular reflectors

(d) A combination of segmented lines, circular reflectors, and physical barrier marking lane location

(e) Highly cluttered shadows from trees obscuring lane markings

(f) Freeway overpass causing extreme lighting changes and reducing road marking contrast

Figure II.2: Images depicting the variety of road markings and conditions for lane position detection and tracking.

Environmental Variability

In addition to the system objective in which the lane position detection will be used, it is important to evaluate the type of environmental variations that are expected to be encountered. Road markings and characteristics can vary greatly not only between regions, but also over nearby stretches of road. Roads can be marked by well defined solid lines, segmented lines, circular reflectors, physical barriers, or even nothing at all. The road surface can be comprised of light pavement, dark pavement, or even combinations of different pavements. An example of the variety of road environments can be seen in figure II.2, all of the images in the figure were taken from roads within a few miles of each other to show the environmental variability within even small regions. In this figure, (a) shows a relatively simple scene with both solid-line and segmented-line lane markings. Lane position detection in this scene can be considered relatively easy because of the clearly defined markings and uniform road texture. Item (b) shows a more complex scene in which the road surface varies and markings consist of circular reflectors as well as solid lines. Item (c) shows a road marked solely with circular reflectors. Item (d) shows a combination circular marking and segmented-line marking as well as a physical barrier. Items (e) and (f) show complex shadowing obscuring road markings. Along with the various types of markings and road, weather conditions and time of day can have a great impact on the visibility of the road surface, as seen in figures II.2e-f and II.3.

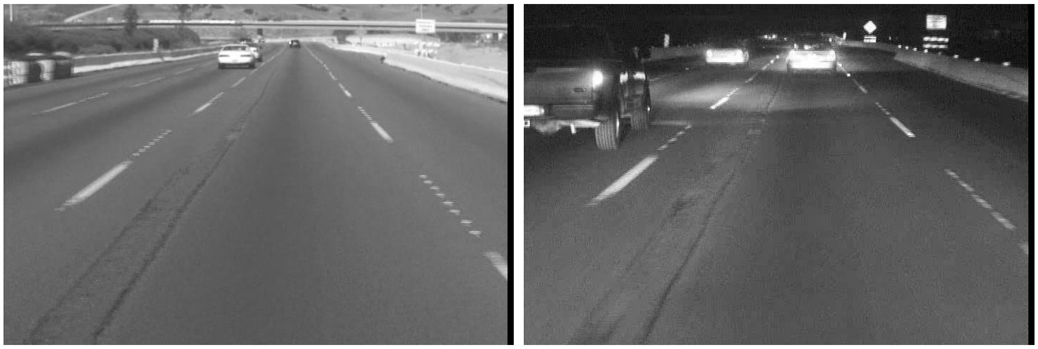


Figure II.3: Images of the same stretch of road shown in the daytime and nighttime.

Sensing Modalities

Various sensors have been studied to perform lane position determination. Examples of these include:

- camera and vision sensors
- internal vehicle state sensors
- line sensors
- LASER RADAR sensors
- global positioning system (GPS) sensors

While LASER RADAR sensors, line sensors, and GPS sensor can perform extremely well in certain situations, vision sensors can be utilized to perform well in a wide variety of situations. LASER RADAR sensors are useful in rural areas for helping to resolve road boundaries [18], but fail on multi-lane roads without the aid of vision data. Line sensors, while accurate for current lateral position, have no look-ahead and cannot be used well for trajectory forecasting, which is needed to compute metrics such as time to lane crossing (TLC) [15]. GPS, especially differential GPS, can provide accurate position resolution, but this requires infrastructure improvements to achieve these accuracies and rely on map data which may be outdated and inaccurate. Vision sensors can provide accurate position information without the need for external infrastructure or relying on previously collected map data. In the situations where vision sensors do not perform well (i.e. extreme weather conditions or off-road conditions), the vision data can be fused with other sensor modalities to provide better estimates. This makes vision sensors a good base on which to build a robust lane position sensing system. Because of these reasons, this article will focus mainly on vision sensors augmented by vehicle state information obtained from the in-vehicle sensors.

II.C Survey of Lane Position Detection and Tracking Systems

In this section we will take a look at the current state of the art in lane position detection and tracking as well as provide a critical comparison between algorithms. Broad surveys of intelligent vehicles have examined many of the lane position sensing algorithms available [19,20]. While these papers are useful for broad examinations of vision research for intelligent vehicles, they are limited in the detail they can provide on lane position sensing because of their broad nature. It is our intent to provide a more in-depth survey of the current methods for lane position sensing. In order to cover such a large expanse of research which has taken place in the last 15 to 20 years, we will group the algorithms discussed here into categories related to the contributions of the algorithms.

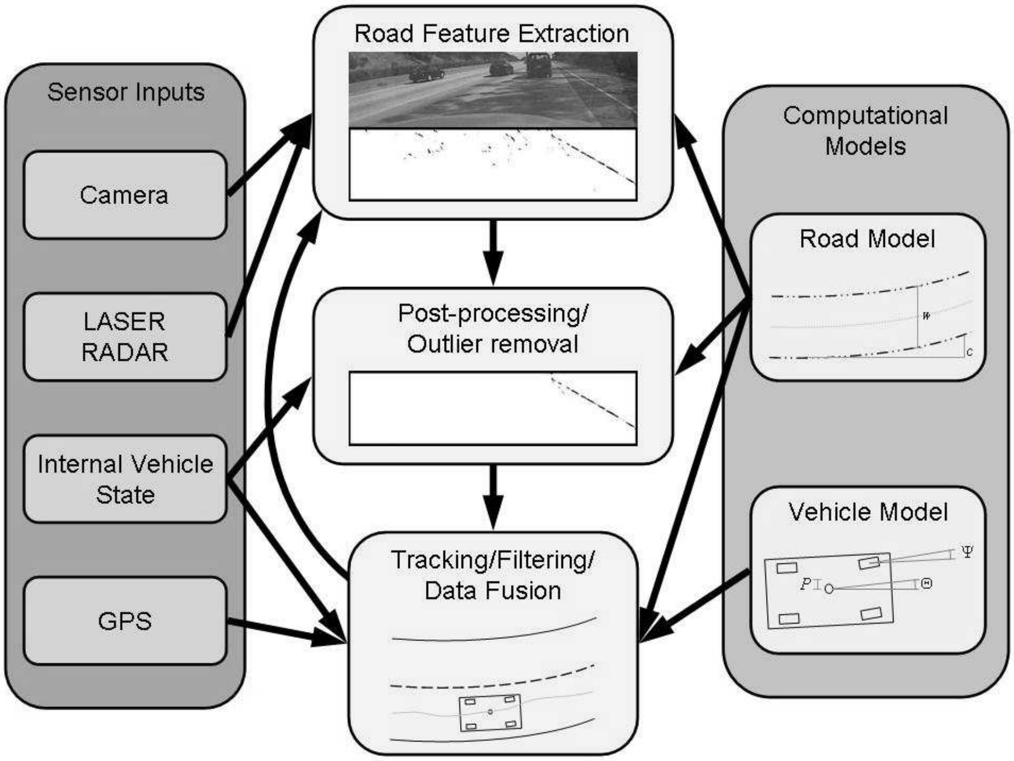


Figure II.4: A generalized flow chart for lane position detection systems combining multiple modalities an iterative detection/tracking loop and road and vehicle models.

After taking an extensive look at the types of lane position tracking algorithms that have been developed, we have noticed similarities in the way that they are structured. Namely, almost all lane position tracking algorithms follow a similar flow. This common system flow is diagramed in figure II.4. First, a model for the road and vehicle is proposed. This can be something as simple as straight lines or more complex clothoid [21] or spline models [22]. Next, a sensing system is used to gather information about the vehicle's environment. Others have used GPS and other sensors to augment lane position estimates [23] and fuse the sensor modalities to work in difficult-to-interpret situations like city driving [11]. However, in this article we will focus on vision sensors combined with vehicle data for reasons described in section II.B.1. Features are then extracted from the sensing system. A few examples of these features are edges, motion vectors and textures. These features are then used in combination with the road model to create an estimate of the lane position. Finally, a vehicle model can then be used to refine these estimates over time given the vehicle data and vision sensing data. This general flow can vary slightly between systems as objectives of these systems change. For example, Taylor et al. [17] propose various control strategies which are tightly coupled with the lane position tracking. Certain exceptions to this flow also exist. Most notable is the ALVINN system [24]in which the neural network directly incorporates the feature detection into the control algorithm with no tracking feedback.

II.C.1 Road Modeling

Road modeling can greatly increase system performance by helping to eliminate false positives via outlier removal. A variety of different road modeling techniques have been used. This variety of techniques stems from the wide variety of roads. Bertozzi and Broggi [25] assumed that the road markings form parallel lines in an inverse-perspective-warped image. Others have used approximations to flat roads with piecewise constant curvatures [21, 26]. More recently, deformable contours such as splines have been used to parameterize roads [22,27]. Maps constructed using differential GPS systems have also been used to provide detailed road models in urban environments [11].

The best choice of road model depends on the type of system and intended environment in which the lane position tracker will be used. For example, complex road models such as spline-based road models might not be a suitable choice for a lane position control system designed to work on highways, which have a relatively simple structure. Furthermore, a stable control system might only require about a 10 meter lookahead [17], making a simple linear road model satisfactory. In a lane departure warning system it is required to calculate the trajectory of the vehicle a few seconds ahead. At freeway speeds, this can require accurate road modeling for 30-40 meters or more ahead of the vehicle to catch TLC of around one second. In this situation, a parabolic or spline-based road model would be better. This is because an accurate curvature model is necessary for vehicle trajectory forecasting.

II.C.2 Road Marking Extraction

Road marking extraction is a key component to lane position detection. Road and lane markings can vary greatly, making the generation of a single feature extraction technique difficult. Edge based techniques can work well with solid and segmented lines, and can even be extended to attempt to compensate for circular reflectors [28]. However, edge based techniques can often fail in situations such as those in figures II.2b, II.2e, and II.2f which contain many extraneous lines. Frequency based techniques, such as the LANA system [29], have been shown to be effective in dealing with extraneous edges. However, they may still be confused by complex shadowing as seen in figure II.2e. The LANA system in particular is restricted to diagonal edges, limiting its effectiveness during lane change maneuvers when the camera is directly above the lane. Other techniques, such as the RALPH system [30], base the lane position on an adaptive road template. These methods generally assume a constant road surface texture and therefore can fail in situations such as in figure II.2b.

Similarly to road modeling, a good choice of a road marking detection also depends greatly on the type of system and environment in which the lane position detection is to be performed. If the system is to be used only on certain types of roads only in specific regions, it might not be necessary to detect all possible variations of road markings. For certain system scenarios, such as autonomous vehicle control, it might not be necessary to find specific road markings at all as long as a safe path or lead vehicle to follow [30] can be found.

II.C.3 Postprocessing

Postprocessing is necessary to improve estimates based on a priori knowledge of the road and extracted features. One of the most common postprocessing techniques used is the Hough transform [31, 32], but other techniques used include enhancing or attenuating features based on orientation [28] or likelihood [26,29] and culling features based on elevation using stereo vision [27]. Dynamic programming has also been used on extracted line segments to help remove outliers more effectively than Hough transforms [33]. Apostoloff et al. [34] performed cue scheduling to help determine which of multiple features should be extracted, processed, and fed into the position tracking module.

In general, postprocessing is one of the most important steps as it ties together the feature extraction stage with the tracking stage by generating a robust estimate of actual lane position based on the extracted features. Most postprocessing techniques make assumptions about the road and vehicle. We will examine these assumptions later in section II.C.5

II.C.4 Vehicle Modeling and Position Tracking

The two most common tracking techniques used in lane position detection systems are Kalman filtering [17, 21] and particle filtering [34, 35]. More complex nonlinear systems have also been used with success [36]. In these systems, feature extraction and position tracking are often combined into a closed loop feedback system in which the tracked lane position defines an a priori estimate of the location and orientation of the extracted features.

Similarly with road models, the choice of vehicle models can vary depending on the primary system objective. For objectives such as vehicle control, complex vehicle models might help to improve stability and perform precise movements. Lane departure warning systems are often designed for high-speed, low-curvature highways. In these situations, a linear approximation to the vehicle model does not significantly affect performance.

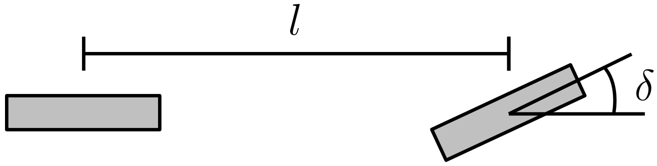


Figure II.5: Bicycle model parameterization commonly used to estimate and predict vehicle dynamics

The most common model for the vehicle dynamics is the bicycle model [37]. In this model (Figure II.5, the system is simplified by assuming that the forces acting on the vehicle are the same as those acting on a vehicle with two wheels separated by a baseline l. Furthermore, it is assumed that only the front wheel is used to steer the vehicle (parameterized by δ.) Often, only the two-dimensional planar case and ignore pitch and roll angles. The continuous time, linearized dynamics of this system are shown in equation II.1, where C_f and C_r represent the front and rear cornering stiffness, L_f and L_r represent the lateral distance between the center of mass and the front and rear tires, v_x represents the vehicle speed, m represents the vehicles mass, and I_z represents the vehicles inertia.

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_f+C_r}{mv_x} & \frac{-L_fC_f+L_rC_r}{mv_x} \\ \frac{-L_fC_f+L_rC_r}{I_zv_x} & -\frac{L_f^2C_f+L_r^2C_r}{I_zv_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_f}{m} \\ \frac{L_fC_f}{I_z} \end{bmatrix} \delta \tag{II.1}$$

II.C.5 Common Assumptions and Comparative Analysis

A significant improvement to the accuracy of lane position estimation can be made by applying a few assumptions based on the structured nature of road surfaces. These assumptions include:

a) The road/lane texture is consistent.

b) The road/lane width is locally constant.

c) Road markings follow strict rules for appearance or placement.

d) The road is a flat plane or follows a strict model for elevation change.

Existing algorithms tend to use at least one or more of these assumptions. These assumptions improve overall results; however, it is important to understand where these assumptions might fail as the lane position tracking is likely to be used for one of the objectives explored in section II.B.1. Any sort of critical failure in these systems could prove disastrous.

The assumption of constant road texture can greatly improve results as the entire road surface is usable as a feature rather than just road markings. In situation in which road markings are scarce or missing, road texture can provide an estimate for lane position [30]. As stated above, roads that have been modified to add lanes or exits (as in figure II.2b) can cause erroneous position estimates.

The assumption that the road or lane width is locally constant can greatly enhance performance by allowing the fusion of left and right hand side boundaries. 3D reconstruction can be performed based on a known constant road width [38]. This assumption is usually valid for most stretches of highway road. However, this is generally not a good assumption for city driving or highways near merging lanes or off-ramps. Unfortunately, merging lanes often are often critical situations in which you would like to have a robust lane position estimate.

Road markings are often assumed to be light solid lines on a dark road surface. However, this is not always the case and as can be seen in figure II.2d, which contains dark lines on a light road surface as well as circular reflectors. Making assumptions about lane marking appearance can greatly degrade performance in places where those assumptions about the road infrastructure are not valid.

Often it is assumed that the road surface is a flat plane or follows a constant curvature elevation model. This is accurate most of the time and allows monocular vision systems to easily transform points on the image plane to 3D points in world coordinates. However, for situations such as changing elevations on curves, these road model assumptions can lead to an incorrect estimation of road curvature. It is important to examine the amount of error in curvature the system can handle before choosing a road model.

Up to this point we have examined the various modules that make up a lane position tracking system, previous research related to each of these modules, and the importance of the modules and the assumptions made about them to the primary objective of the system. It is also important to take a look at systems as a whole and how they compare in performance based on their objectives, environments and sensing systems. Table II.1 serves to help summarize and compare various lane position detection and tracking algorithms in relation to the objectives of the system in which they are deployed.

The objectives of many systems, especially the earlier developed systems, were geared towards autonomous vehicle control. The VaMoRs system [21] uses multiple processors and both wide angle and telephoto lenses for vehicle guidance. A linear vehicle model and 3D road model were used. The system was tested on a rural road with hills. For autonomous control, the systems can adjust vehicle speed allowing more time for computation; this is a valid assumption unique to the autonomous control objective. The YARF system [39] uses multiple features and robust estimation to help improve performance of the autonomous driving task. Differences in the detected versus expected features are used to identify situations in which the road structure is changing. Taylor et al. [17] show a vehicle control system which they analyzed using a variety of control schemes. Using these different control schemes, they tested their system on a oval test track and measured performance based on the vehicles offset from the center line. The DARVIN system [11] fuses dGPS information with vision information for supervised autonomous driving in an urban environment. The use of higher accuracy GPS system provides the benefit of having more exact knowledge of the road structure. Differential GPS also provides a good a priori knowledge about the vehicle's location which can be improved upon using vision algorithms. However, the use of dGPS makes the system more reliant on a constantly updating infrastructure system and only provides up-to-the-minute knowledge on the vehicles position. Changes in the road structure, such as construction zones, would need to be relayed to the vehicle for the road model to retain it accuracy. The GOLD system [25] combined lane position tracking with obstacle detection for autonomous freeway driving. A special function finds lane markings in an inverse perspective road image based on brightness differences between a pixel and its neighbors to the left and right. A 3000 Km test run was performed and images were shown demonstrating robustness to occlusions and shadows. More recently the Springrobot [31] used an adaptive, randomized Hough transform for processing detected edges.

While the systems mentioned above have focused mainly on the autonomous control objective, others have focused on the lane departure warning and driver assistance objectives. Kwon and Lee [10] developed a system for lane departure warning based on a modular architecture that allowed fusion of multiple features. The lane position and rate of departure was then fed into a heuristic departure warning function. Testing was performed based on the overall system performance and quantified in a number of metrics including the detection rate, false alarm rate, missed detection rate, and alarm triggering time. The LOIS system [40] was also used in a lane departure warning system. In this system, edge magnitude and orientation was used along with a maximum a posteriori estimator to provide lane position. They showed results from a test run with a standard deviation of error of around 13cm. Risack et al. [41] demonstrate a lane departure warning system based on the TLC measure. As with most of the other systems, performance was measured for the system as a whole with little quantitative results related to the lane position tracking.

Another major difference between the various systems that have been developed stems from the types of environments for which these systems were designed.

Ma et al. [18] present a system which fuses RADAR information with vision to navigate rural roads. The RADAR images improved performance for weather conditions such as snow, which obstructs the camera view. The DARVIN system mentioned above used GPS to allow navigation through urban areas. The vast majority of systems however, are designed for highway environments. This is important for the commercial sector in which a large amount of research has been performed [13].

This analysis and comparison of these systems with respect to their primary objective and intended environment enables us to see some of the merits and deficiencies of these systems. We have seen that improvements to performance can be made by applying feature extractors that use multiple cues or can be used to extract multiple types of road markings. Assumptions about the road and vehicle models have also been shown to greatly increase performance. However, care needs to be taken that assumptions made about the road environment that are assumed to apply to a wide range of environments are not actually limited only to specific regions. Often testing is performed by the examination of a few key frames or simple tests taken in only a few environments. It was these realizations that led us to develop a lane tracking system designed for driver assistance functions and capable of performing well under a wider variety of environments. It is also important to provide a thorough evaluation of the system to enable a better comparison of performance between various environments. This includes evaluating the system at different times of the day with varying road markings and textures as well as taking a close look at special case scenarios, such as tunnels, to get an accurate, quantitative measure of performance.

Table II.1: A comparison of various lane position detection and tracking techniques.

System	Use[1]	Road Model	Feature Extraction	Postprocessing	Tracking	Evaluation	Comments
VaMoRs (1992) [21]	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) [39]	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) [24, 42]	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) [30]	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

Continued on Next Page...

Table II.1 continued. . .

System	Use[1]	Road Model	Feature Extraction	Postprocessing	Tracking	Evaluation	Comments
GOLD (1998) [25]	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) [40]	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) [29]	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) [17]	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.

Continued on Next Page. . .

Table II.1 continued...

System	Use[1]	Road Model	Feature Extraction	Postprocessing	Tracking	Evaluation	Comments
Ma et al. (2000) [18]	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) [35]	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) [10, 36]	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) [11]	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.

Continued on Next Page...

Table II.1 continued. . .

System	Use[1]	Road Model	Feature Extraction	Postprocessing	Tracking	Evaluation	Comments
Lee et al. (2003) [43, 44]	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) [34]	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. II.2b)
Kang et al. (2003) [33]	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.

Continued on Next Page. . .

Table II.1 continued. . .

System	Use[1]	Road Model	Feature Extraction	Postprocessing	Tracking	Evaluation	Comments
Nedevschi et al. (2004) [27]	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 II.E.2	

II.D The Video Based Lane Estimation and Tracking (VioLET) System For Driver Assistance

Breaking down the design into the sections illustrated in figure II.4 helps to create a lane position detection and tracking system focused on one or more of the system objectives described in section II.B.1 and capable of handling a variety of the environmental conditions explored in section II.B.1. By examining the system one piece at a time and understanding how that choice might affect overall system performance, we can optimize our system for our objective of driver assistance.

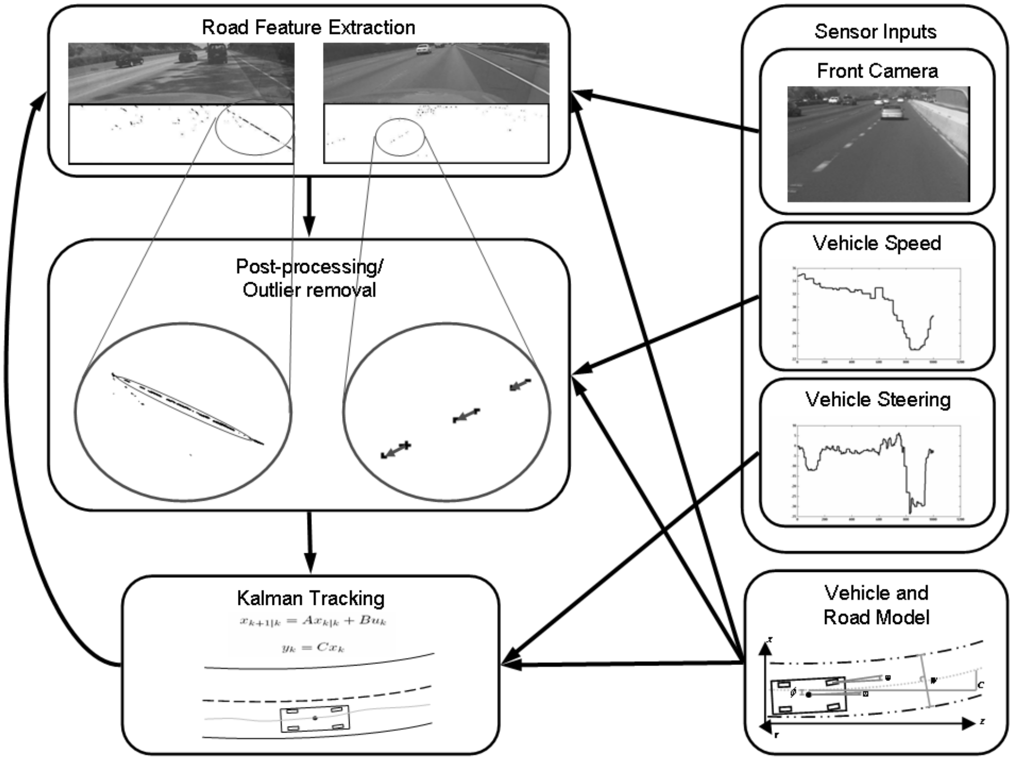


Figure II.6: System flow for VioLET, a driver assistance focused lane position estimation and tracking system.

The primary objective of the VioLET system is driver assistance. This is a rather broad objective, so some clarification is necessary. It is our intention for the system to provide accurate lateral position over time for the purposes of lane departure warning and driver intent inferencing. The intended environment for the lateral position detection is daytime and nighttime highway driving under a variety of different roadway environments. These environments include shadowing and lighting changes, road surface texture changes, and road markings consisting of circular reflectors, segmented lines, and solid lines. The VioLET system follows a similar flow to the generic system flow described in section II.C. The system specific flowchart is diagramed in greater detail in Figure II.6. In this section we will describe each of the system modules and the motivation behind their development.

II.D.1 Vehicle and Road Modeling

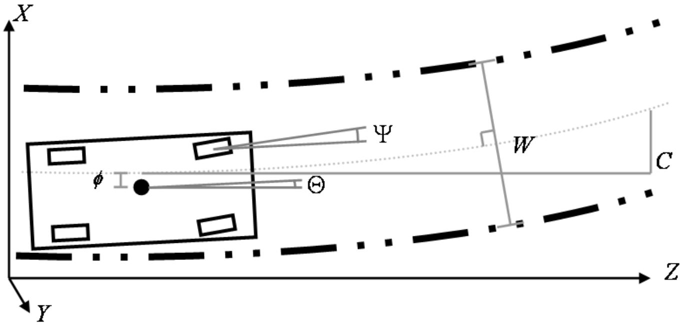


Figure II.7: Vehicle and road models used in the system. We are using a constant curvature road model and linearized vehicle dynamics for use in a Kalman filter.

Our system objective requires a road and vehicle model that retains accuracy for distances of at least 30-40 meters. This is required because, in critical situations in which driver assistance systems are useful, a prediction of the vehicle trajectory at least one second ahead of the vehicle is necessary. A simple parabolic road model, as shown in figure II.7, incorporates position, angle and curvature while approximating a clothoid model commonly used in the construction of highway roads [21]. In the figure, X_s represents the lateral offset along the center of the road, Z_s represents the distance in front of the vehicle, ϕ represents lateral position, θ represent the lane angle, C represents lane curvature, Ψ represents the steering angle, and W represents the lane width. Equation II.2 describes the road down the center of the lane while equation II.3 describes the road at the lane boundaries. l takes the value of 1 for the left lane and -1 for the right lane. Lane width is assumed locally constant, but is updated via a Kalman filter described in section II.D.5. The vehicle dynamics are approximated using a bicycle model similar to that used in [17].

$$X_s(Z_s) = \phi + \theta Z_s + C Z_s^2 \tag{II.2}$$

$$X_{border}(Z_s) = \phi + \theta Z_s + C Z_s^2 + \frac{lW}{2(\theta + C Z_s)^2 + 2} \tag{II.3a}$$

$$Z_{border}(Z_s) = Z_s - \frac{lW(\theta + C Z_s)}{2(\theta + C Z_s)^2 + 2} \tag{II.3b}$$

II.D.2 Road Feature Extraction

Road feature extraction is a difficult problem for a variety of reasons. For our objective and intended environment, it is necessary to have a robust estimate of road features given a variety of road marking types and road environments. Making the problem even more difficult is the necessity for fast algorithms for feature extraction. To this end, we have found features extracted by using steerable filters provide robust results for multiple types of lane markings and are able to be decomposed into simple convolutions and arithmetic capable of being implemented in a digital signal processor.

Steerable filters have a number of desirable properties that make them excellent for a lane position detection application. First, they can be created to be separable in order to speed processing. By separating the filters into an X and Y component, the convolution of the filter with an image can be split into two convolutions using the X component and Y component separately. Second, a finite number of rotation angles for a specific steerable filter are needed to form a basis set of all angles of that steerable filter. This allows us to see the response of a filter at a given angle and therefore to tune the filter to specific lane angles or look at all angles at once. This property is useful because circular reflectors will have high responses in all directions while line markings will have high responses in a single direction.

The steerable filters used for the circular reflectors and lane detection are based on second derivatives of two-dimensional Gaussians.

$$G_{xx}(x,y) = \frac{\partial^2}{dx^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} = -(\frac{2x}{\sigma^2} - 1)\frac{2}{\sigma^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} \tag{II.4}$$

$$G_{xy}(x,y) = \frac{\partial^2}{dxdy} e^{\frac{-(x^2+y^2)}{\sigma^2}} = \frac{4xy}{\sigma^4} e^{\frac{-(x^2+y^2)}{\sigma^2}} \tag{II.5}$$

$$G_{yy}(x,y) = \frac{\partial^2}{dy^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} = -(\frac{2y}{\sigma^2} - 1)\frac{2}{\sigma^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} \tag{II.6}$$

It has been shown that the response of any rotation of the G$_{xx}$ filter can be computed using the equation II.7 [45].

$$G2^{\theta}(x,y) = G_{xx}\cos^2\theta + G_{yy}\sin^2\theta - 2G_{xy}\cos\theta\sin\theta \tag{II.7}$$

Taking the derivative of equation II.7, setting it equal to 0, and solving for θ, we can find the values of that correspond to the minimum and maximum responses. These responses can be computed by the formulas given in equations II.8 and II.9.

$$\theta_{min} = \arctan(\frac{G_{xx} - G_{yy} - A}{2G_{xy}}) \tag{II.8}$$

$$\theta_{max} = \arctan(\frac{G_{xx} - G_{yy} + A}{2G_{xy}}) \tag{II.9}$$

where,

$$A = \sqrt{G_{xx}^2 - 2G_{xx}G_{yy} + G_{yy}^2 + 4G_{xy}^2} \tag{II.10}$$

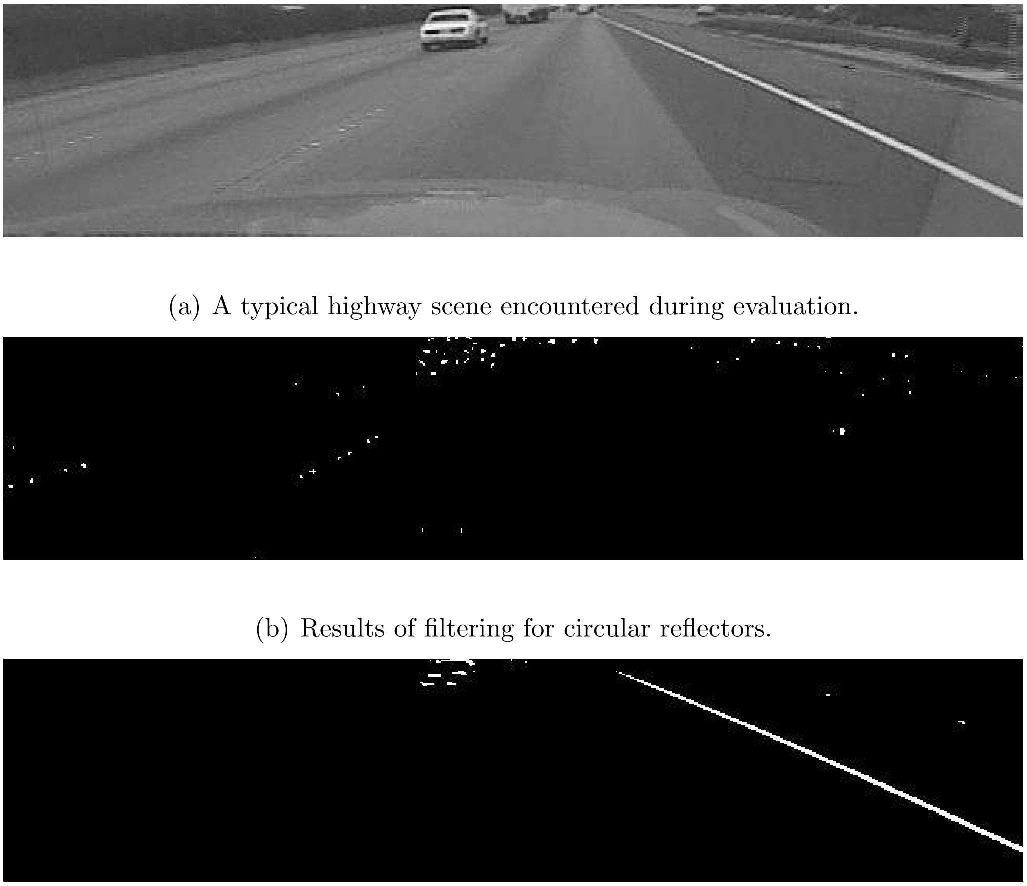
$$G2^{\theta_{min/max}} = G_{yy} - \frac{2G_{xy}^2}{G_{xx} - G_{yy} \pm A} \tag{II.11}$$

Using the equations II.7, II.8, and II.9, we can find the values and angles of the minimum and maximum responses, or the response at a given angle. This is useful for detecting circular reflectors because, for small circular objects, the minimum and maximum responses will be very similar. In order to detect circular reflectors, we can therefore threshold the filtered image for minimum responses that are above a certain value as well as within a certain range of the maximum value. These minimum and maximum response values can be computed efficiently using equation

II.11. For detecting lanes, the response in the direction of the lane should be near the maximum, and the minimum response should be low. Also, applying a threshold to the difference between the response in the direction of the lane marking and the minimum response, we can detect lanes of a specific angle. Figure II.8a shows a typical highway scene with lane markings consisting of both circular reflectors and solid lines. Figure II.8b shows the image after being filtered and thresholded by the minimum response value. Figure II.8c shows the response to lines in the orientation of the current lane parameters. The filter kernel size was chosen to be roughly three times the expected lane marker width. Filtering on the inverse perspective warped image allows a single kernel size to be used over the entire area of interest.

These results show the usefulness of the steerable filter set for relatively normal highway conditions. This filtering technique is also very useful for dealing with shadowed regions of road. Figure II.9 below shows a road section that is shadowed by trees and the filter response for the lane when it is tuned for that lane angle.

The same technique can be applied to images that are transformed into world coordinates using the inverse perspective equations described in equation II.12. In this equation, T and R represent the transformation and rotation of the camera respectively. The world coordinate Y is assumed zero because of the flat plane road model. Example images of the inverse perspective warped image as well as results for circular and lane marking detection are shown in figures II.10 and II.11. The advantages to this technique include the ability to use a single filter size for a wider

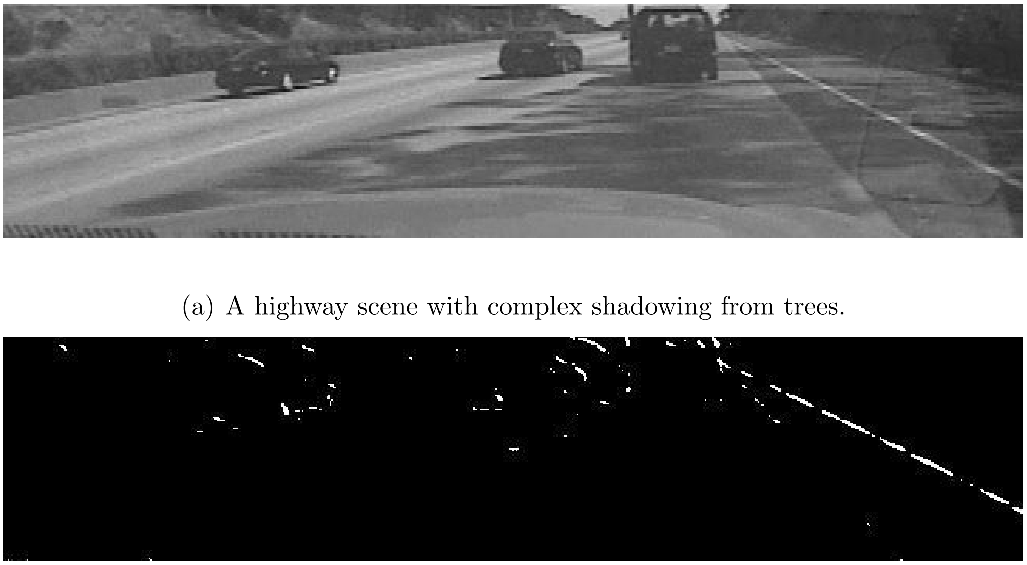


(a) A typical highway scene encountered during evaluation.

(b) Results of filtering for circular reflectors.

(c) Results from filter for a line tuned to the lane angle.

Figure II.8: Application of Steerable filter road marking recognition for circular reflectors and solid lines on a highway.



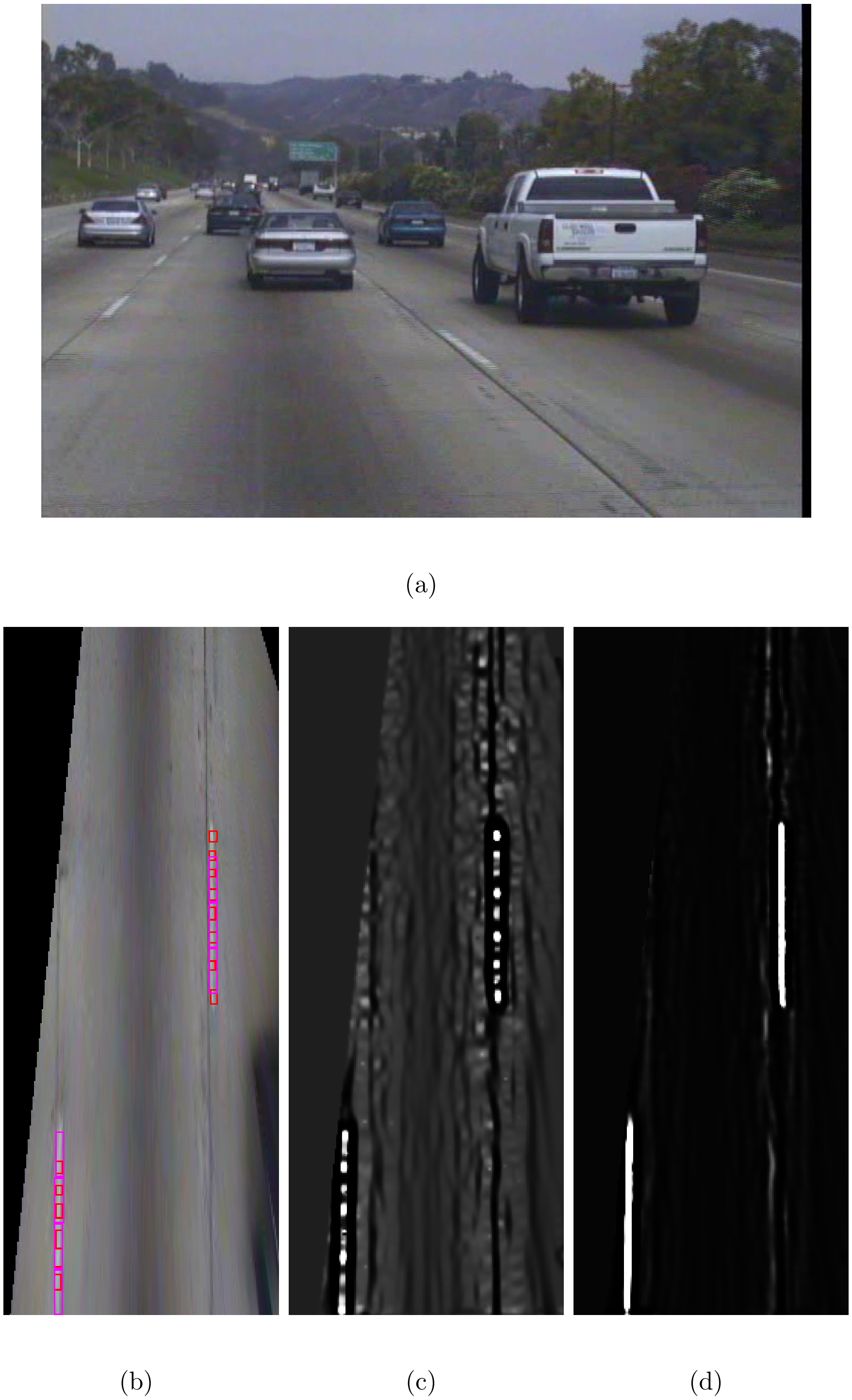
(a) A highway scene with complex shadowing from trees.

(b) Detection results for lines tuned to the lane angle.

Figure II.9: Filter results when lane markings are shadowed with complex shadows and non-uniform road materials.

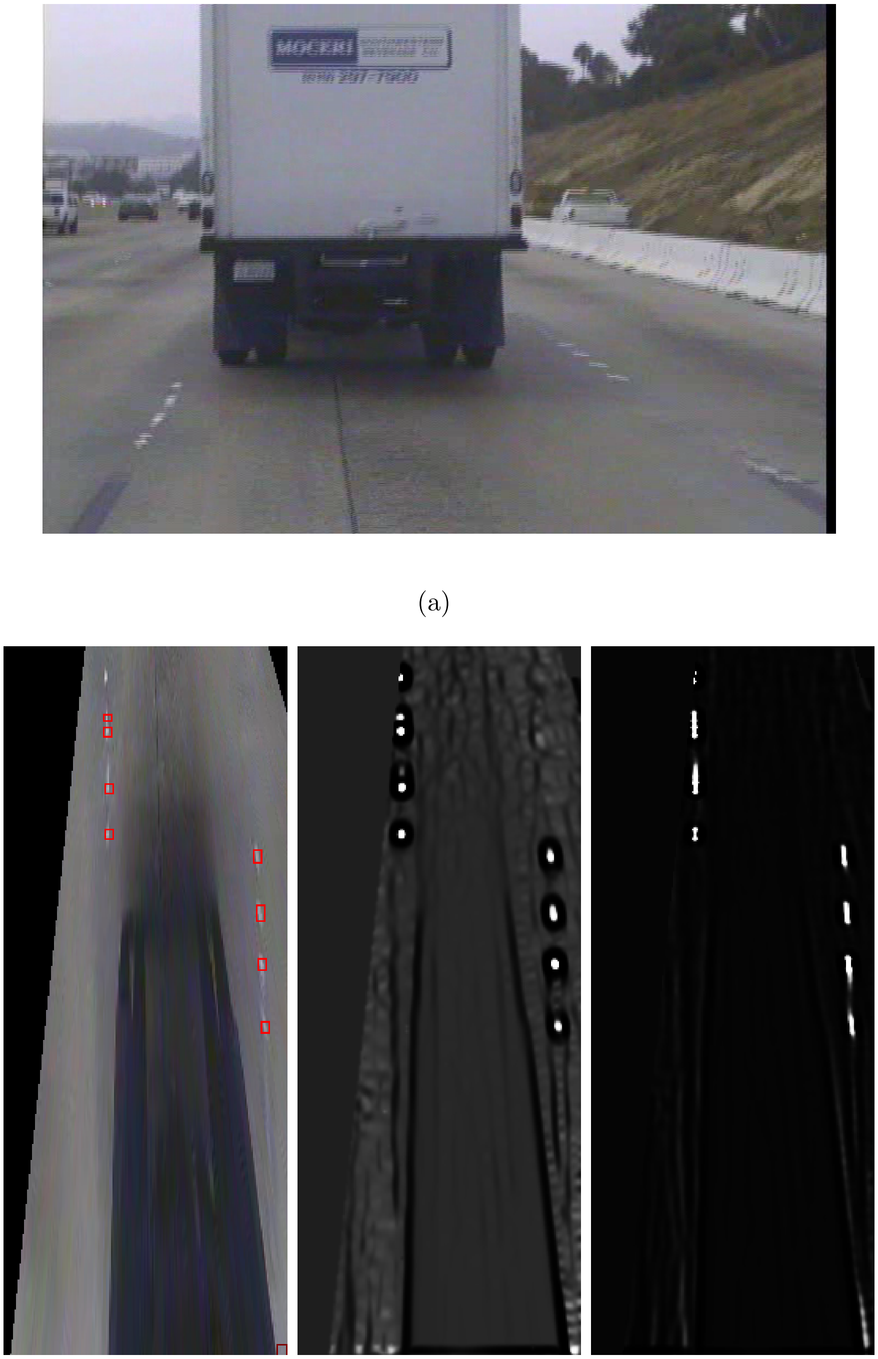
range of distances from the vehicle and the ability to easily modify the resolution of the image to maintain a balance between the level of detail and processing time. We will explore the utility of the inverse perspective image and transformation further in the following sections.

$$\begin{bmatrix} x_{\text{image}} \\ y_{\text{image}} \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix}, \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X_{\text{world}} \\ 0 \\ Z_{\text{world}} \\ 1 \end{bmatrix} \tag{II.12}$$


(a)

(b) (c) (d)

Figure II.10: Image from a road scene containing solid line markings with embedded circular reflectors (a) as well as the image transformed using the inverse perspective transformation (b), which is subsequently filtered for circular reflectors (c) and solid line markings (d).



(a)

(b) (c) (d)

Figure II.11: Image from a road scene containing circular reflector markings (a) as well as the image transformed using the inverse perspective transformation (b), which is subsequently filtered for circular reflectors (c) and solid line markings (d).

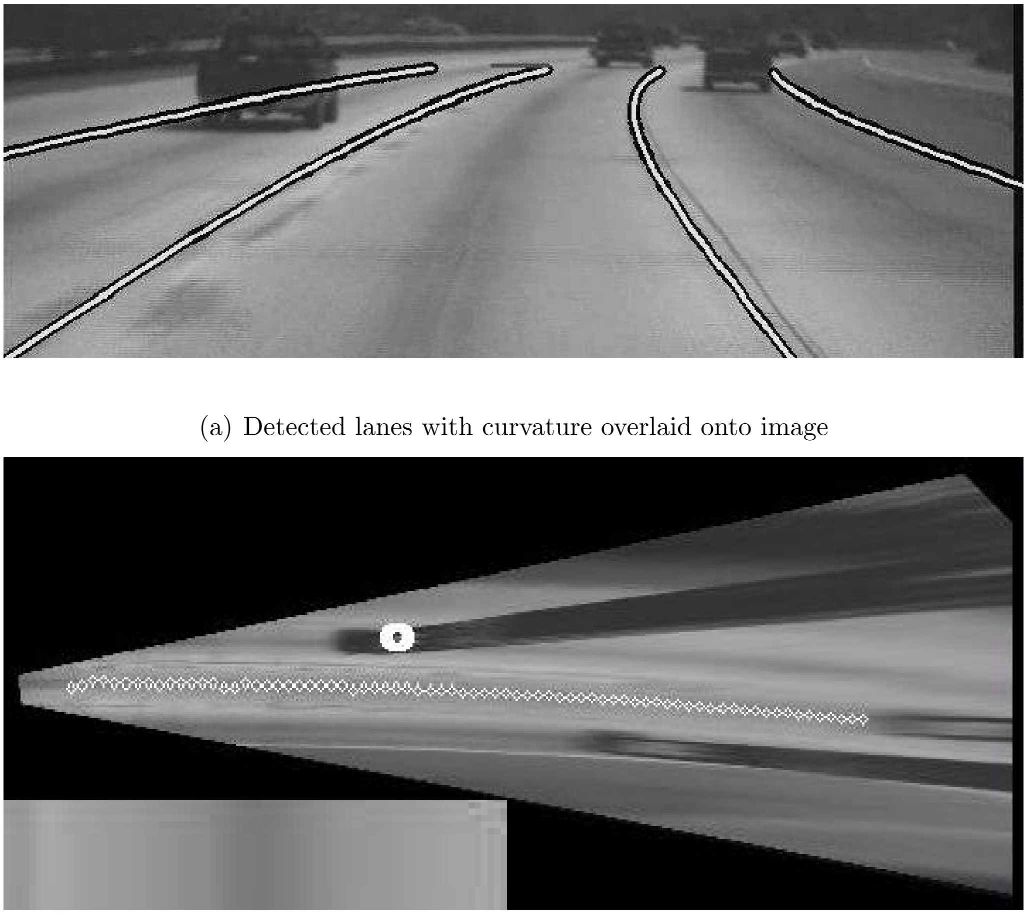
II.D.3 Road Curvature Estimation

Some sections of road within our intended environment are marked solely by circular reflectors as is seen in figure II.2f. These circular reflectors are too small to be seen with the cameras used in our configuration at distances greater than about 20 meters. In these situations an adaptive template is used to measure curvature beyond the range of what is detectable by road markings alone. Curvature detection is performed by matching a template of the current road to the road ahead, then fitting the detected results to the lane model described in section II.D.1. The adaptive template is generated per pixel using a weighted average of the intensity values of the previous template and the intensity values of the lane area for the current image. The intensity values for the lane area are found by applying an inverse perspective warping to the image and cropping a rectangular area centered on the current estimate of the lane position a few meters ahead of the vehicle. The weighting can be adjusted to allow faster or slower response times and is initialized using the intensity values of the initial frame. The template is then matched to the road ahead by minimizing the squared error in intensity values of the inverse perspective warped image. The error is minimized laterally at equally-spaced distances ahead of the vehicle to get an estimate of the lateral position of the road at specific distances ahead of the vehicle. The final curvature estimate is generated by minimizing the squared error between the parabolic road model and the measured road positions. While this method works well on most roads with little traffic, template matching techniques such as these fail in cases of poor road texture and occlusion. For this reason, curvature is also estimated using the vehicles yaw rate and the second derivative of position. These are estimated using the Kalman filtering equations described in section II.D.5. This provides a robust estimate on lane curvature by combining the vehicle state information with visual cues from the lane tracking system to determine instantaneous curvature when road look-ahead is not sufficient.

Figure II.12 shows the results of the curvature detection system. Figure II.12a shows a forward looking view with the detected lane positions overlaid onto the image. Figure II.12b shows aerial photography for that specific section of road. The vehicles trajectory is depicted in this figure using a green curve for future trajectory and a red curve for past trajectory. Figure II.12c shows the inverse perspective warping of the forward looking camera with the detected lane points shown as small white circles. The template is shown in the lower left hand corner of figure II.12c.

II.D.4 Postprocessing and Outlier Removal

In order to perform robust tracking in situations such as in figures II.2 and II.3, post-processing on the filter results is performed. First, only the filter candidates within the vicinity of the lanes are used in updating the lanes. This removes outliers from other vehicles and extraneous road markings. Because the algorithm uses a local search about the lanes for candidates, it requires initialization. In testing, it was sufficient to initialize the lane tracker position and trajectory to



(a) Detected lanes with curvature overlaid onto image

(b) Inverse perspective warping showing curvature detection (small white dots) and template (lower left corner)

Figure II.12: Curvature detection in the VioLET lane tracking system.

zero (corresponding to the center of the lane).

Furthermore, for each lane, the first and second moments of the point candidates are computed. Straight lane markings should be aligned so that there is a high variance in the lane heading direction and a low variance in the other direction. Outliers are then removed based on the eigenvalues and eigenvectors of the computed covariance matrix. The aspect ratio of the eigenvalues and the direction of the eigenvector associated with the major axis of the concentration ellipse are used to eliminate outliers.

For circular reflectors, the speed of the vehicle is used to calculate the expected location of the reflector. This is performed using the inverse perspective equations described in II.12. Circular reflector detections which do not move as predicted by the ground plane are removed as they generally correspond to false detections. These false detections commonly stem from things such as specular highlights on vehicles and other small circular textures which do not move with the ground plane.

II.D.5 Position Tracking

Position tracking for our objective of driver assistance is vitally important. Position tracking can provide improved results in noisy situations and generate other useful metrics important for the primary system objective. Kalman filtering provides a way to incorporate a linearized version of the system dynamics to generate optimal estimates under the assumption of Gaussian noise. Kalman filtering also provides estimates of state variables which are not directly observable, but may be useful for the system. It is important to have metrics such as rates of change of position robustly estimated not only from lane angles, which may contain errors for vehicle pitch or camera calibration, but from lane position estimates over time.

$$x_{k+1|k} = Ax_{k|k} \tag{II.13}$$

$$y_k = Mx_k \tag{II.14}$$

where

$$x = [\phi, \dot{\phi} \approx \Theta, \dot{\Theta}, \Phi, W]^T \tag{II.15}$$

$$A = \begin{bmatrix} 1 & v\Delta t & \frac{(v\Delta t)^2}{2} & \frac{(v\Delta t)^3}{6l} & 0 \\ 0 & 1 & v\Delta t & \frac{(v\Delta t)^2}{2l} & 0 \\ 0 & 0 & 1 & \frac{v\Delta t}{l} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{II.16}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.5 \\ 1 & 0 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{II.17}$$

The Kalman filter state variables are updated using the lane position and angle estimates along with measurements of steering angle and wheel velocity. These measurements are then used to update the discrete time Kalman filter for the road and vehicle state. I our system, we convert the linearized model presented in equation II.1 to a discrete time system and simplifying the system by assuming a relatively large vehicle mass and inertia. We have also included the yaw rate Φ and lane width W into the system. The system and measurement equations as well as the Kalman update equations at time k are detailed in equations II.13 to II.17. The variables used in these equations are the same as those described in section II.D.1 and figure II.7. The measurements consist of the left and right lane marking positions, the lane angle, and the yaw rate of the vehicle. Curvature is currently calculated and filtered separately. This is calculated separately using the steering angle and road curvature. The initial values for the estimation-error covariance and state-noise covariance were determined by empirical testing. Adding a control input to the Kalman equations allows us to effectively use steering and yaw rate information from the vehicle similar to that described in Southall et. al. [35]

The measurement vector y_k (equation II.14) consists of the vehicle position, the lane angle, and the lane width. These measurements are found using a combination of a Hough transform and the lane marker detection statistics. For solid lane markings, the Hough transform provides a robust means of determining the location and angle of individual lane markings. When a valid line cannot be found using a Hough transform, as in the case of lanes marked with circular reflectors, the statistics of the lane markings are used to determine the position and angle of the lanes. These statistic are described in section II.D.4. This estimation is performed for both the left and right lane markings. These estimates are then used to determine estimates of the lane position, angle, and width using a weighted sum.

II.E Experiments and Performance Evaluation

Lane detection systems have been studied quite extensively and several metrics for the evaluation of lane position error have been proposed [16, 46]. However, most proposed algorithms have shown limited numerical results or simply selected images of the algorithm results. While these images provide information on the performance on road marking extraction in specific contexts, they fail to account for errors involved in transforming image coordinates to world coordinates and cannot be used to quantitatively compare different algorithms. In order to adequately measure the effectiveness of a lane position detection and tracking system in a specific context or system, specific metrics must be used. In this section we will explore the usefulness of a variety of performance metrics and show how the algorithm described in this paper performs based on these metrics in a variety of test conditions.

II.E.1 System Test-bed Configuration and Test Conditions

The video input to the system is taken from a forward looking rectilinear camera for our test results, but can be taken from any number of cameras on our test bed vehicle. The test bed is pictured in figure VI.1. Some of the key capabilities of the LISA-Q intelligent vehicle test bed include:

- Eight NTSC hardware video compressors for simultaneous capture.
- Controller-Area-Network (CAN) interface for acquiring steering angle, pedals, yaw rate, and other vehicle information.
- Built-in 5 beam forward looking LASER RADAR range finder.
- WAAS enabled GPS.
- Integration into car audio and after-market video displays for feedback and alerts.

More information on this test bed is provided in Chapter Appendix A. Information about the vehicle's state, including wheel velocities and steering angle, are acquired from the car via the internal CAN bus.

Testing was performed on highways in southern California. These highways contained road conditions shown in figures II.2 and II.3. Namely this includes:

- lighting changes from overpasses.
- circular lane markers, solid-line lane markers, and segmented-line lane markers.
- shadowing from trees and vehicles.
- changes in road surface material.

A camera directed downwards at the road on the side of the vehicle provided a good view for generating positional ground truth data. The cameras used in the system were calibrated for their intrinsic parameters using the Matlab Camera Calibration Toolbox [47].

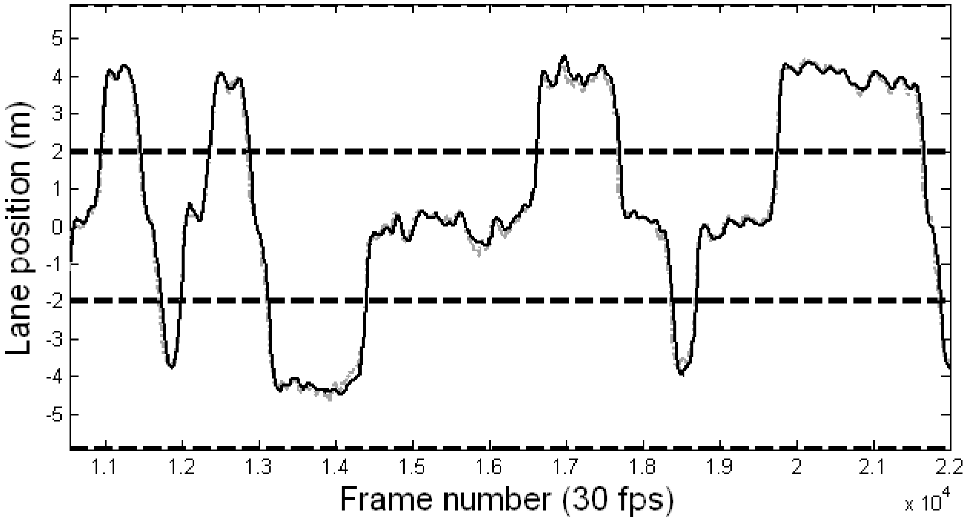


Figure II.13: Detected lateral position in meters (solid black) superimposed on ground truth (dashed grey) plotted vs. frame number with dashed lines marking the position of lane boundaries for an 11,000 frame (slightly over 6 minute) sequence.

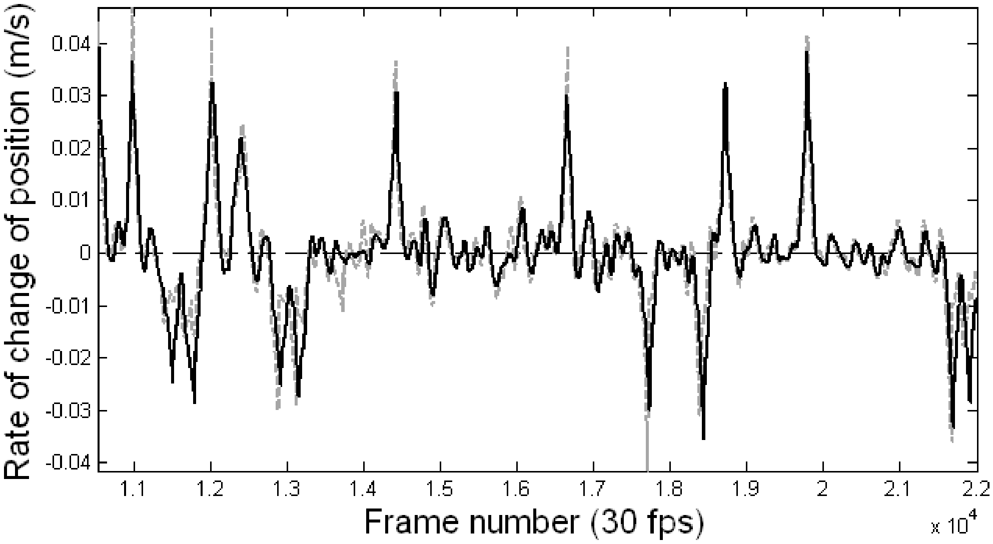


Figure II.14: Detected departure rate in m/s (solid black) superimposed on ground truth (dashed grey) plotted vs. frame number with dashed line marking the abscissa for the same sequence shown in figure II.13.

II.E.2 Choice of Metrics For Objective Specific Performance Evaluation

One of the most common metrics for lane position performance evaluation is mean absolute error. While this provides a good estimate of the performance of a lane position tracker for system objectives such as control and driver intent, it lacks usefulness in quantifying the accuracy for other objectives like road departure warning in which the TLC and rate of approach to the road boundary are important. For this reason it is important to use a variety of performance metrics when evaluating a system rather than just one.

Other statistical metrics have been proposed which are based on examining the distribution of the detected markings [16]. These include the Angular Deviation Entropy and Angular Deviation Histogram Fraction and their magnitude-weighted counterparts. While these serve as good online metrics for evaluating relative system performance for different stretches of road, they are not as useful for determining system performance relative to a known ground truth.

Several metrics have been proposed to evaluate the performance of driver lane change intent and road departure warning systems. These systems are related because they deal with forecasting the vehicles trajectory. Most of these involve looking at the system as a whole and measuring false positives, false negatives, or the time it takes to trigger an alarm [8,10,46]. However, because the systems involve the collection of data other than just lateral position, it is difficult to decouple the lane position performance from the system performance using these types of metrics. In order to generate an accurate prediction of performance in a trajectory forecasting objective, it is necessary to examine the accuracy of the parameters used to generate this forecast. In this situation, we expect the metrics of error distribution of the rate of change of lateral position to provide good indicators of system performance. The rate of change of lateral position metric was chosen over the time-to-lane-crossing metric for two reasons. First, the rate-of-change metric has been shown to be useful in driver assistance [44] and driver intent [3] applications. Second, the time-to-lane-crossing metric is prone to large errors stemming from small errors in vehicle position and lateral velocity. Furthermore, generating a ground truth for the time to lane crossing is complicated by the need for a well known road and vehicle model for the entire stretch of road on which the testing is being performed.

II.E.3 Evaluation and Quantitative Results

In order to provide a more complete test of our system, we chose to quantify the error using three different metrics. The three metrics we chose are mean absolute error in position, standard deviation of error in position, and standard deviation of error in rate of change of lateral position.

Results were analyzed according to the metrics discussed in section II.E.2 under the environmental variations described in section II.E.1. More specifically, data was collected from portions of the roughly 65 kilometer route at four different times of the day: dawn, noon, late afternoon/dusk, and night. Scenes from each

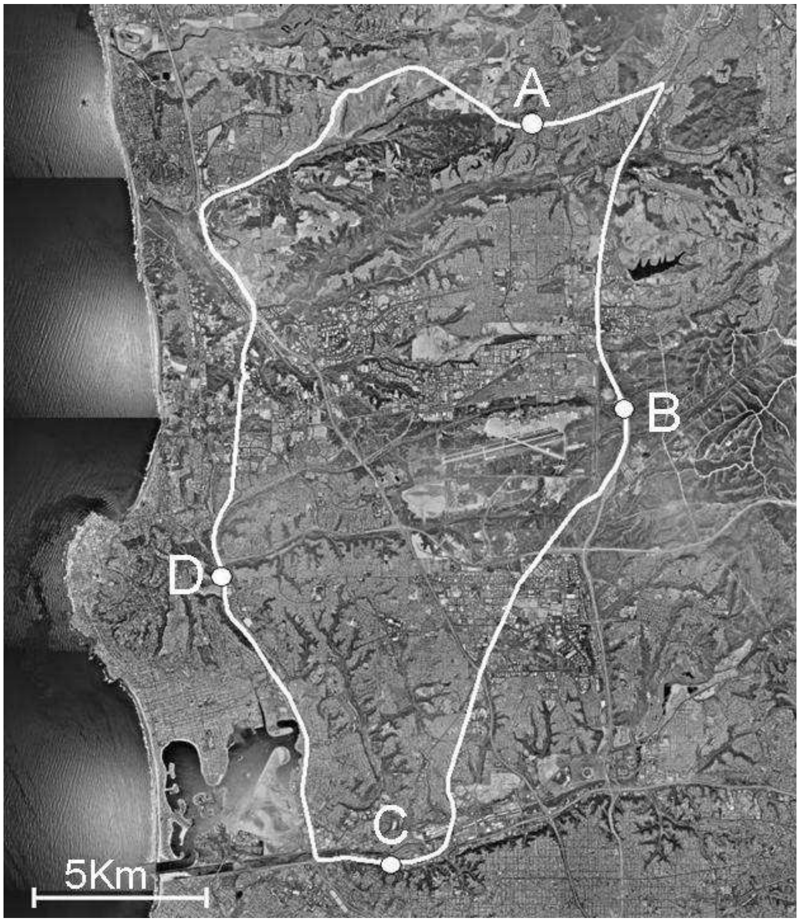


Figure II.15: The 65Km route in San Diego used in the evaluation. The route is overlayed on aerial photography. Points A, B, C, and D are sections of road used in the evaluation (photography courtesy USGS).

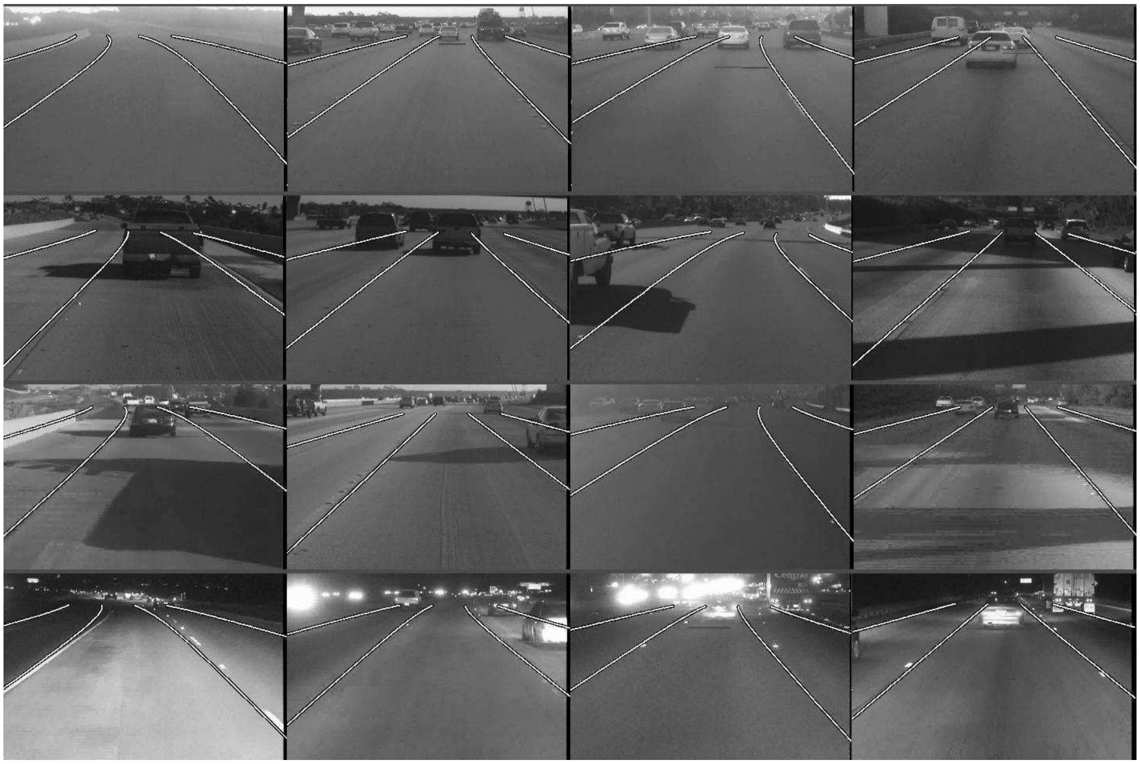


Figure II.16: Scenes from dawn (row 1), daytime (row 2), dusk (row 3), and nighttime (row 4) data runs for each of the four sections of road. These scenes show the environmental variability caused by road markings and surfaces, weather, and lighting.

of these corresponding to the points A, B, C, and D in figure II.15 along with an aerial view of the individual points are shown in figure IV.21. Section A consists of solid-line and segmented-line markers, while sections B and C contained a mixture of segmented lines and circular reflectors. Section D is marked by circular reflectors. Ground truth for one-thousand frame sequences was found for each of these locations on the route and each of the four times of day, making a total of sixteen thousand testing frames covering many different highway types heading in different directions at different times of the day. These results are shown in shown in tables II.2, II.3, and II.4. Per frame outputs and ground truth for selected data sets can be seen in figures II.19-II.21.

After examining the results, it is interesting to note that the system actually performs better at night and dawn than during day and dusk. At night, this can be attributed to the larger contrast in road markings due to their reflective nature as well as the lack of complex shadows formed by trees and vehicles during the daytime. complex shadows hampers the systems ability to detect circular reflectors in scenes such as that shown in figure II.9. At dawn, a morning fog reduced contrast somewhat, but also helped eliminate shadows. The low contrast of the dawn run required a change in the thresholds used in feature extraction. Future work will include making the system more adaptive to general lighting changes.

Furthermore, the difference in departure rate performance between daytime and nighttime driving would point to increased number of successful detections (i.e. those not eliminated by outlier removal). The comparatively smaller gain

Table II.2: Results from the Standard Deviation of Error performance metric evaluated under various lighting and road conditions.

	Standard Deviation of Error (cm)				
	Dawn	Noon	Dusk	Night	Total
Set A	4.5400	11.5700	8.1062	7.9710	8.4221
Set B	8.6041	14.8687	7.9457	3.8871	9.6612
Set C	11.1815	13.5135	29.9347	23.2722	20.8885
Set D	5.1547	10.7514	12.1687	8.3031	9.4761
Totals	7.8460	12.7784	17.1246	13.1261	**13.1377**

Table II.3: Results from the Mean Absolute Error performance metric evaluated under various lighting and road conditions.

	Mean Absolute Error (cm)				
	Dawn	Noon	Dusk	Night	Total
Set A	3.6497	8.6429	5.5313	6.4720	6.0740
Set B	6.8463	10.6362	5.6768	3.0417	6.5503
Set C	8.1815	10.8677	20.4727	12.9471	13.1173
Set D	4.1713	8.4701	9.8390	6.5232	7.2509
Totals	5.7122	9.6542	10.3800	7.2460	**8.2481**

Table II.4: Results from the departure rate performance metric evaluated under various lighting and road conditions.

	Standard deviation of error in departure rate metric (cm/s)				
	Dawn	Noon	Dusk	Night	Total
Set A	0.11107	0.31885	0.18453	0.20120	0.21710
Set B	0.21364	0.32725	0.24030	0.06224	0.23149
Set C	0.26277	0.29709	0.70780	0.39743	0.45173
Set D	0.10842	0.29767	0.19349	0.21076	0.21344
Totals	0.18627	0.31049	0.39693	0.24836	**0.29595**

in standard deviation performance over mean absolute error might suggest that the tracking at night performed better overall, but still contained cases where the tracking was off. This is because the mean absolute error metric is less influenced by the small amounts of data points which contain a larger amount of error.

Comparing performance for different type of lane markings, we can see that section A, which contained solid and segmented line markings, performed better than the other sections, which at points were marked only with circular reflectors. However, this difference is less noticeable than the variations caused by lighting and traffic.

As the system's intended use is in a driver safety system, it is critical that we analyze the situation in which the lane tracking did not perform well. Section C can be seen to be the most difficult section of road based on the results. Looking deeper into the cause of these errors we can see points where the tracking is lost for a period of time and then catches back on again. An example of one of these events occurs near frame 82,900 of figure II.20. Figure II.17 shows this tracking error resulting from occlusion of the road by a vehicle. This section of road is near the intersection of three different freeways and therefore generally contains more traffic and more people merging and changing lanes. In most of the situations where the tracking was lost, vehicles changed lanes directly in front of the test bed. The specular highlights and lines of the vehicles caused false positives in the feature extraction. Another important event occurred near frame 24,650 of figure II.19. At this point the video signal was lost due to saturation of the CCD while exiting a tunnel causing the vertical sync to be misaligned. It is also important to note that while these types of failures might be critical for a autonomous vehicle, whereas a driver assistance system can warn the driver when the current lane detections do not fit well with previous data and the system is not functioning properly.

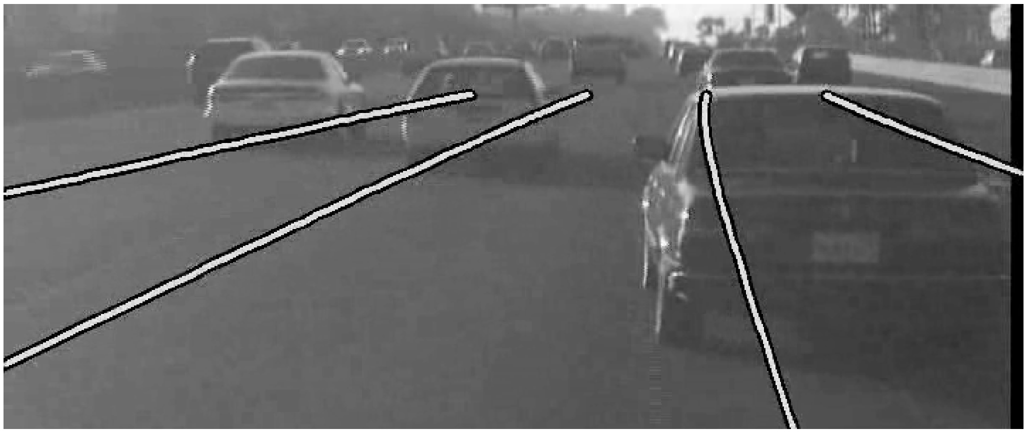


Figure II.17: Error due to occlusion of the road by a vehicle on the dusk dataset on road segment C.

II.E.4 Lane Keeping vs. Lane Changing Performance

Furthermore, the different driving scenarios require different performances for different types of driving. Specifically, lane departure warning systems need to accurately detect when the driver is close to the edge of the lane. This makes it important to test the performance during lane changes. Lane keeping control systems might require good performance only near the center of the lane, where the system is designed to operate. We therefore also measured performance during lane change maneuvers and compared this with performance during lane keeping.

Table II.5 show the results for the lane keeping driving context versus

Table II.5: Results from various metrics associated with lane keeping vs. lane changing.

	Metrics		
	Std. Dev. of Error (cm)	Absolute Mean Error (cm)	Std. Dev. of Error in Rate of Change (cm/s)
Lane Keeping	10.4263	8.2661	0.1625
Lane Changing	16.0634	12.4149	0.4653
Combined	13.0060	9.9446	0.3639

the lane changing driving context.This is an important distinction because some systems, such as lane departure warning systems, are required to operate well during lane changing situations. The data samples used in this evaluation were collected during the day and include circular reflector markings, dash line markings, clear roads, shadowed roads, and overpasses. This data is shown in figures II.13 and II.14. From the table we can see that there is performance degradation when changing lanes. This is also evident in the relative higher errors at the peaks and troughs of figure II.14 which correspond to higher lateral velocities during lane chages. These types of errors are possibly associated with errors in the vehicle model, errors in yaw rate sensing, and delays associated with Kalman filtering.

II.E.5 Special Case Scenario Testing

Often it is the case that the situations which can be considered the most critical are less likely to occur. Figure II.18 shows cutscenes from a situation in which the road surface is obfuscated by complex shadows and a tunnel sequence that contains tight quarters and extreme lighting changes. In this section we will

Table II.6: Results for the special case scenarios of tunnels and complex shadows.

	Error Metrics		
	Std. Dev. of Error (cm)	Absolute Mean Error (cm)	Std. Dev. of Error in Rate of Change (cm/s)
Tunnel	22.7438	17.8740	0.51141
Complex Shadows	10.3869	15.4488	0.45682

analyze the performance of our lane position tracker in these two situations.

First we will examine the performance while traveling through tunnels. Unfortunately, tunnels are not common in the area where testing was performed so only one tunnel in a more urban setting was used in the testing. The results from the evaluation are quantified in table II.6. At the end of the tunnel was a sharp left hand turn for which tracking failed as our road model was not designed for such use. Figure II.21 shows the detected position superimposed on ground truth.

The second special case was traveling through complex shadows. Often trees, overpasses, cars, and other object can cast shadows with sharp edges and complex shapes. These can pose problems for lane tracking systems because they form extraneous edges, obscure the road texture, and otherwise complicate feature detection. The results from the evaluation can be seen in table II.6. This obscuring of the lane markings by complex shadows only slightly degrades performance.

II.F Conclusion

In this chapter we have presented a detailed analysis of the use of lane position in a variety of system objectives, road environments, and sensing systems.

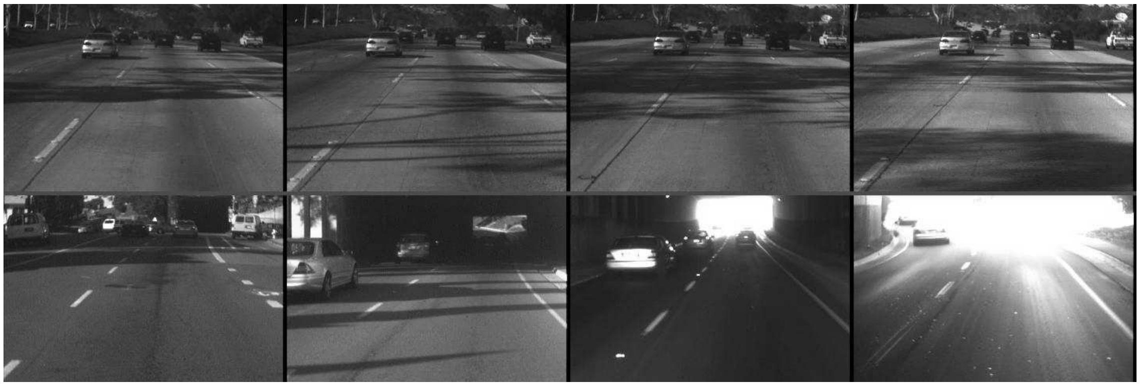


Figure II.18: Scenes from the special case scenarios of complex shadowing (top row) and tunnels (bottom row). These scenes highlight the extreme variability that can occur within short sections of road.

We then presented a framework for comparative discussion and evaluation of existing lane tracking systems. This led to our presentation of the novel VioLET lane tracking system. A system designed for driver assistance vehicles operating in a wide range of environments. The VioLET system introduces steerable filters to the lane detection and tracking problem by allowing greater robustness to complex shadowing and lighting changes, while at the same time maintaining a computational simplicity necessary for fast implementations. Using both vehicle state information as well as visual cues (lane markings and lane texture) we created robust estimates of lane curvature both with and without lookahead visibility. Finally we provided a detailed analysis of our system with an extensive set of experiments using a unique instrumented vehicle test bed. This evaluation allowed us to compare performance between different types of road markings at different times of the day, different types of driving situations, and special case scenarios that can be critical points in driver assistance systems.

Along with providing metrics for evaluating system performance, it is also important to note the dependence of various driver assistance systems to the metrics provided by the underlying sensing algorithms. An example of this is providing the departure rate to a lane departure warning system can enhance the performance of such systems. Systems have been designed to provide good performance based on a subset of data than is required to accurately predict the exact motion of the vehicle within the lane [44]. With this in mind, we plan on identifying and exploring further the types of metrics useful for various driver assistance systems. An example of this work can be seen in [7] and [3].

Specific examples of the types of system objectives that this lane position tracker was designed to be used are those described in Huang et al. [48], Gandhi et al. [49], and Trivedi et al. [50]. These systems are designed to capture the complete vehicle context including vehicle surround, vehicle state, and driver state. By capturing the complete vehicle context, we open the possibility of developing driver assistance systems focused on the driver and his or her intended actions [3].

The text of Chapter II, in part, is a reprint of the material as it appears in: J. M^{c}Call and M. M. Trivedi, "Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, March 2006. I was the primary researcher of the cited material and the co-author listed in this publication directed and supervised the research which forms a basis for this chapter.

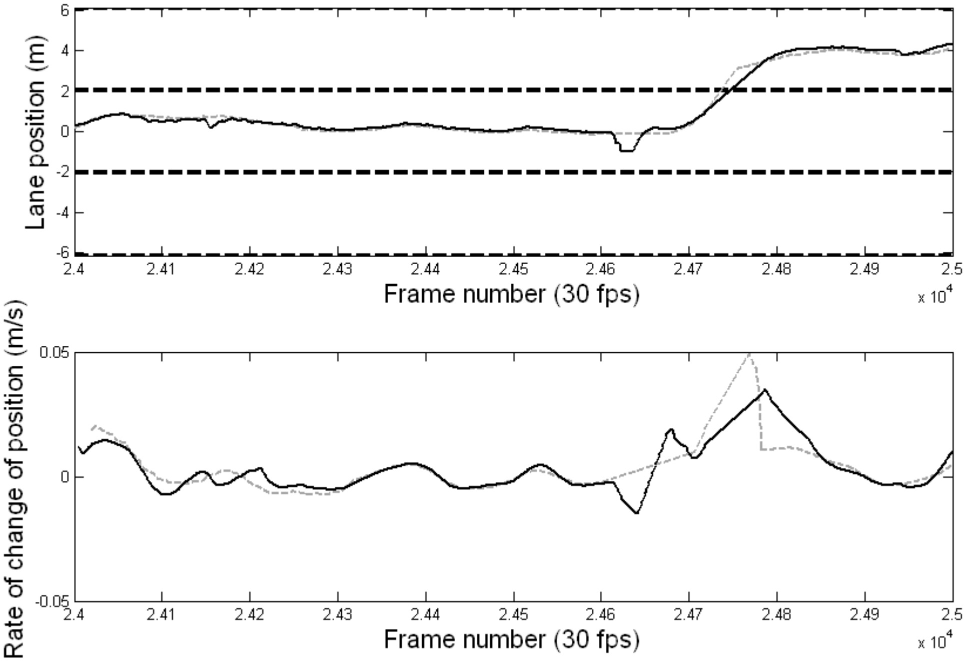


Figure II.19: Lateral position (top) and rate of departure (bottom) for road section A at noon.

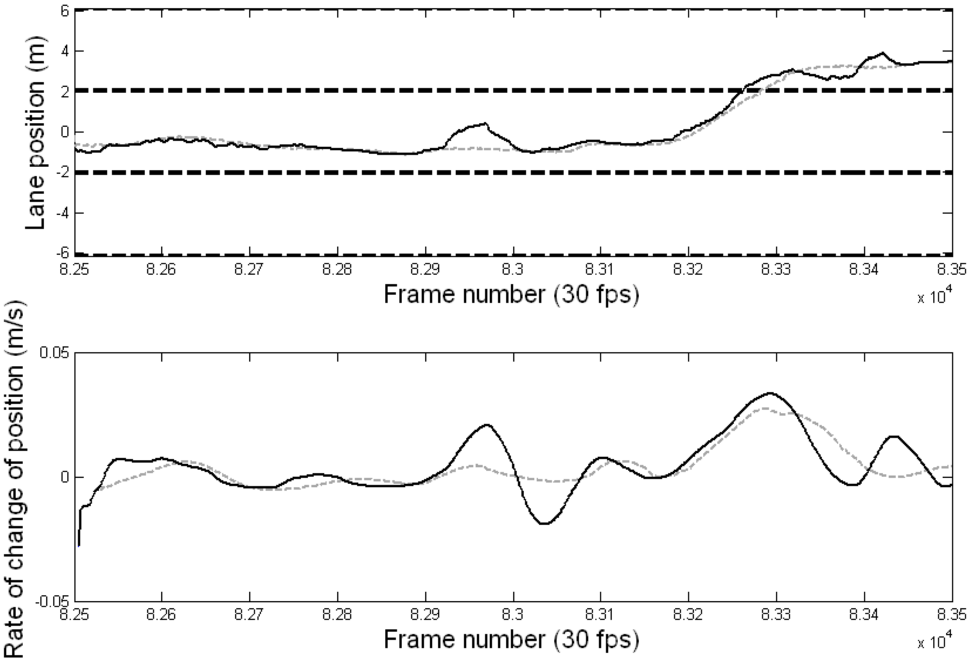


Figure II.20: Lateral position (top) and rate of departure (bottom) for road section C at dusk.

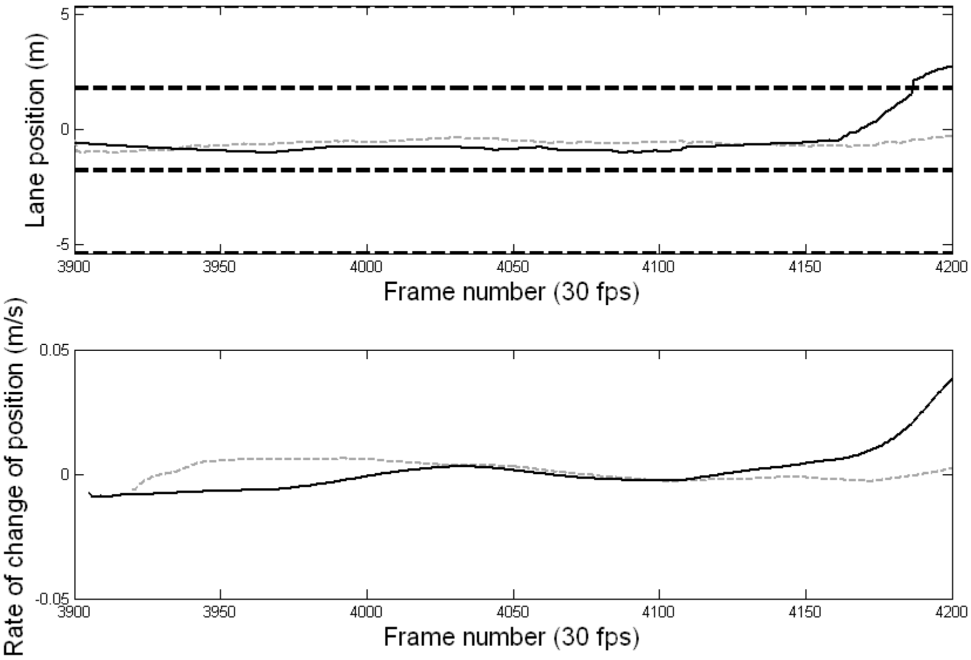


Figure II.21: Lateral position (top) and rate of departure (bottom) for the special case scenario of traveling through a tunnel. High road curvature at the end of the tunnel results in a loss of tracking.

Chapter III

Driver Affect Analysis

III.A Introduction

Analysis of facial expressions by machine vision systems is an important research area for many applications. Applications ranging from user interfaces to intelligent vehicles and spaces can be greatly enhanced with the incorporation of expression analysis [51]. It has been shown that there are six universally common facial expressions for displaying the emotions of Anger, Disgust, Fear, Happiness, Sadness, and Surprise [52]. However, A more descriptive alphabet of human emotion can be constructed by looking at individual components of facial expressions; specifically, we implore facial action units as presented by Ekman et al [53] as a method for identify emotion and mental state. We will help to identify these facial action units by examining the motion of individual facial landmarks such as corners of the eyes, mouth, eyebrows, etc. Classifying individual action units allows us to explore other areas of human computer interactions by broadening the number of emotional states that can be captured. This thereby expands the contexts in which such systems can be used. As an example, intelligent vehicle systems for driver departure warning can be enhanced with the ability to predict fatigue and attentiveness [51].

Facial expression and affect analysis is something we instinctively perceive; however, many factors contribute to the difficulty in machine identification of facial actions and expressions. These factors include difficulties imposed by lighting conditions, varying emotional factors that lead to facial expressions, variations in the expressions between persons, head poses and head movements [54].

In this chapter we propose a novel framework that integrates the problem of face and facial landmark detection and tracking using a variety of cues. By constructing a probabilistic model based on a hierarchical pyramid of facial region trackers, we can more robustly predict individual facial landmark locations while at the same time provide robustness to occlusion. In our pyramid of facial region trackers, we first detect and track faces in a given image. Using this observed knowledge of the location of faces within the image, we can more easily and accurately detect smaller regions of the face. This continues until we have identified specific facial landmarks. We will discuss this approach in more detail in section III.B. The tracking mechanisms used in each layer of the hierarchy are constructed using particle filtering [55]. Particle filtering provides a method for tracking complex distributions over time in an efficient manner as well as combining observations from a variety of cues. By fusing multiple cues we can create robustness to a variety of environmental and lighting conditions. For example, changes in lighting might cause cues such as facial motion to yield poor results as algorithm assumptions are violated, but cues retrieved from facial structure are often more robust to such conditions.

III.A.1 Related Research in Facial Affect Analysis

In order to robustly identify facial actions, one must first identify the subjects face as well as properly register the location of specific facial landmarks. From this point on a variety of techniques can be employed to analyze facial expressions and facial action codes.

Bassili [56] has shown the humans are capable of identifying expressions when presented simply with points placed at facial landmarks. Optical flow and facial motion has also been shown to be useful in the automatic identification of facial affects and expressions [57–59]. However, these methods often break down when rigid body motion, due to head movement, is present. Attempts to solve this problem of separating the rigid motion from the non-rigid motion have been made using model-based estimation [57].

Non-rigid feature tracking has been explored in a variety of ways ranging from methods based purely on optical flow and probabilities, to methods involving the construction detailed facial models and perturbing them according to facial landmark movement. Black and Yacoob parameterized various facial feature motions with affine and similar transformation models [60]. Another approach to solve this problem is to input more complex feature vectors into classification systems.

Systems developed using Graph Matching [61], Neural Networks [62], and Support Vector Machines [63] have been shown to be effective, but require more complex classification schemes.

Appearance cues have also been shown to be very important for facial affect identification. Active appearance models [64] use a linear combination of detailed 3-dimensional models to estimate head pose and appearance. The fitting is performed by aligning a model generated image with the input image. Generic appearance-based object detectors based on Haar wavelets and boosted classifiers have also been shown to be useful in detecting faces [65] and facial features [66]. These types of single frame detectors have also been included as observations to tracking systems [67].

Particle filtering provides a robust method for tracking objects and features by estimating the probability of the object's current state given previous states and observations [55]. It accomplishes this by using Monte Carlo sampling techniques to approximate probability density functions without any assumptions of Gaussianity. Others have used combinations of particle filters for tracking facial features and constraining them to fit particular models of facial movement [68].

In our work, we build upon this previous research by developing a hierarchical structure of particle filters which operate on a variety of different observations. This hierarchy allows for the coarse to fine tracking of facial landmarks. Region tracking results obtained from coarser levels of detail are used to condition the probability estimates of the finer detailed face regions. The observations at each level of the particle filter hierarchy include structural information, relative orientations, locations, and sizes to other filters in the hierarchy; general appearance information, generated by Haar wavelet based object detection; and specific appearance information, generated by adaptive templates of facial regions and landmarks.

III.B Real-time Affect Analysis using Hierarchical Particle Filtering

In this section we will introduce the components that make up our Real-Time Affect Analysis System (RAAS). This system is composed of multiple levels of particle filters related through a hierarchical structure that allows the propagation of landmark location estimates from a coarse resolution to a fine resolution. Figure III.2 consists of a diagram showing the structure of this system. The individual components will be described in the next two sections of this chapter.

III.B.1 Particle Filtering Overview

Particle filtering is a convenient method for estimating the probability density of the current state of an object given the object's previous states as well as current and past observations. This is expressed mathematically as $p\left(x_t|x_{0:t-1}, y_{0:t}\right)$, where x_t is the object state at time t and $y_{0:t}$ are the observations from times 0 through t. This distribution can be estimated using a properly weighted sample distribution. By choosing a transition prior of $p\left(x_t|x_{t-1}\right)$, the weight update procedure

```
If Resampling, generate new samples $\mathbf{s}_t$ from the samples $\mathbf{s}_{t-1}$ using a multinomial distribution with coefficients corresponding to the weights $\mathbf{w}_{t-1}$. Reset weights $w_{i,t-1} = \frac{1}{N} \forall i \in \{1, ..., N\}$.
Update each sample $i$ using the transition prior such that
    $s'_{t,i} = p\left(x_t | x_{t-1} = s_{t,i}\right)$
Re-weight the samples based on the current observations using the equation
    $w_{t,i} = w_{t-1,i} p\left(y_t | x_{t,i} = s'_{t,i}\right)$
```

Figure III.1: Particle Filtering Algorithm

for each sample, indexed by i at time t, simply becomes $w_{t,i} = w_{t-1,i} \cdot p\left(y_t | x_{t,i}\right)$ [69]. The filtering algorithm used in each of the levels of our system is the same as that presented in [55] and shown in figure III.1.

III.B.2 Creating a Hierarchy of Particle Filters

One limitation of particle filtering is its susceptibility to the "curse of dimentionality," [70] where the number of samples required for an accurate estimation grows exponentially with the dimension of the state space. For facial features, the dimensionality for the complete characterization of facial motions is quite large, requiring the state space to be split into multiple filters which can be constrained based on a facial model. Others have done this by separating facial landmarks into groups and then tracking the groups individually under a constrained facial model [68].

By using a dynamic Bayesian network framework, we can condition the finer detailed tracking on results achieved from tracking regions at a coarse level.

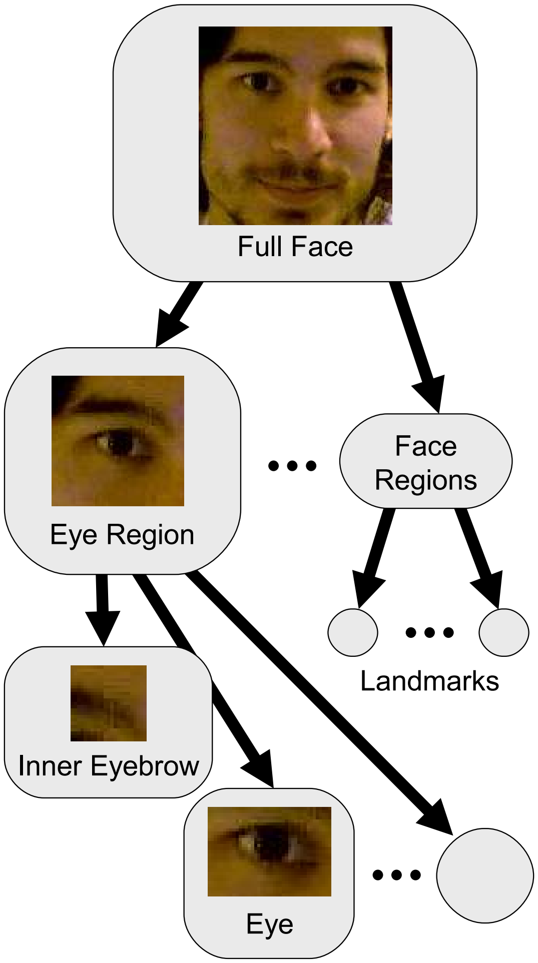


Figure III.2: Bayesian network for facial landmark tracking. The structure for the right eye landmark tracking is emphasized. The associated probabilities are propagated in time using particle filtering to make the network dynamic.

The structure of our Bayesian network is shown in figure III.2. Each node in the network is made dynamic by propagating the probabilities in time using particle filtering. Each node also has it's own observations as we will describe later. As an example, the subset of the network for location of eye region landmarks highlighted in the figure is expressed in mathematical terms in Equation III.1. Where EB, E, ER, F are the parameterizations for the eyebrow, eye, eye region, and face respectively.

$$p\left(EB, E, ER, F\right) = p\left(EB|ER\right) p\left(E|ER\right) p\left(ER|F\right) \tag{III.1}$$

Using this framework, we can adjust the particle filtering mechanism in each of the nodes by conditioning them on their respective parent nodes. Our particle filter now tries to find the probability $p\left(x_t|x_{0:t-1}, y_{0:t}, z_t\right)$ where z_t represents the parent node's parameters. This effectively adds an additional observation to the nodes particle filter. Equation III.2 expresses the sample re-weighting step (step 3 in figure III.1) using this modified filter.

$$w_{t,i} = w_{t-1,i} \cdot p\left(z_t, y_t|x_{t-1,i}\right) \tag{III.2}$$

The parameters we chose to represent each node consists of the x and y image coordinates of the region, the size of the region, the aspect ratio of the region, and the rotation of the region along the image axis. Our state transition probability was chosen to model additive noise with zero mean. The variance was set to account for non-linear facial movements and was determined through empirical testing. More complex models could be trained from facial motion data. In the next section we will explain the node observations in more detail.

III.C Fusion of Multiple Observations

Particle filtering provides a useful probabilistic framework for incorporating multiple observations. Using the “naive” Bayesian assumption that the observations are independent of each other, the observation probability density function can be factored into the product of each of the density functions of the observations o contained in the set of all observations Obs. Strictly, the naive Bayesian assumption doesn’t hold; however, in practice systems based on this assumption have been shown to work quite well [71]. This density function is shown in equation III.3.

$$p\left(y_t|x_t\right) = \prod_{o \in Obs} p_o\left(y_{t,o}|x_t\right) \tag{III.3}$$

The manner in which we choose these probability density functions can provide flexibility and robustness to the system. For example, certain observations, such as those generated from adaptive templates, initially might not provide good information. In this case, we can assign a high variance Gaussian distribution or even a uniform distribution to the density function so that it has little or no impact on the sample re-weighting. We can perform similar adjustments in cases where certain features are occluded by relying on structural information and observational outputs of the other filters in the current and higher levels in our particle filter hierarchy.

The observations at each level of the particle filter hierarchy include general appearance information, generated by Haar wavelet based object detection; person-specific appearance information, generated by adaptive templates of facial regions and landmarks; and global structure cues; generated by the higher order filter of the hierarchy. In the sections below we will describe these cues currently used in the RAAS system.

III.C.1 Haar Wavelet Based Cues

Object detection using a cascade of boosted Haar wavelet based classifiers has been shown to provide highly accurate results with little false positives [65]. Others have used the output of Haar wavelet classifiers in filtering systems [67]. Similarly, in our system, we construct the observation probability by constructing a mixture of equally weighted Gaussian distributions centered on each detected object. The classifier cascades were trained on faces sampled from the FERET database [72] which we hand labeled with the locations of individual facial landmarks. The cascades for the top level particle filter were trained against randomly selected non-face background images obtained from the internet. The cascades for lower level particle filters were trained against background images of facial regions excluding the region or landmark being detected.

To deal with occlusion of regions of the face, multiple Haar wavelet observers can be combined a la the “naive” Bayesian assumption stated earlier at

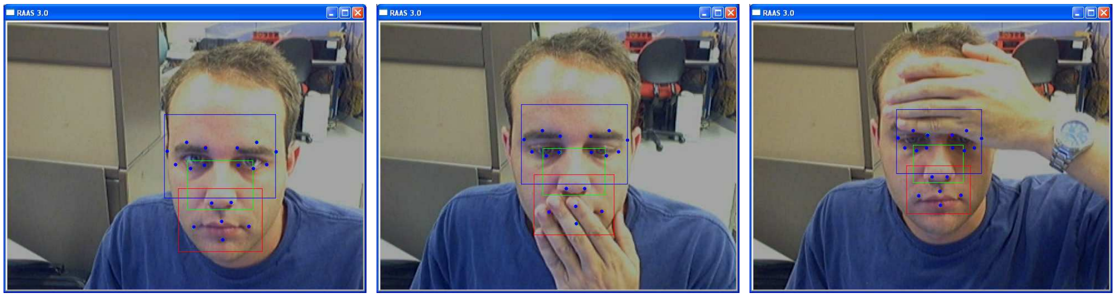


Figure III.3: Face Detection Results with Occlusion

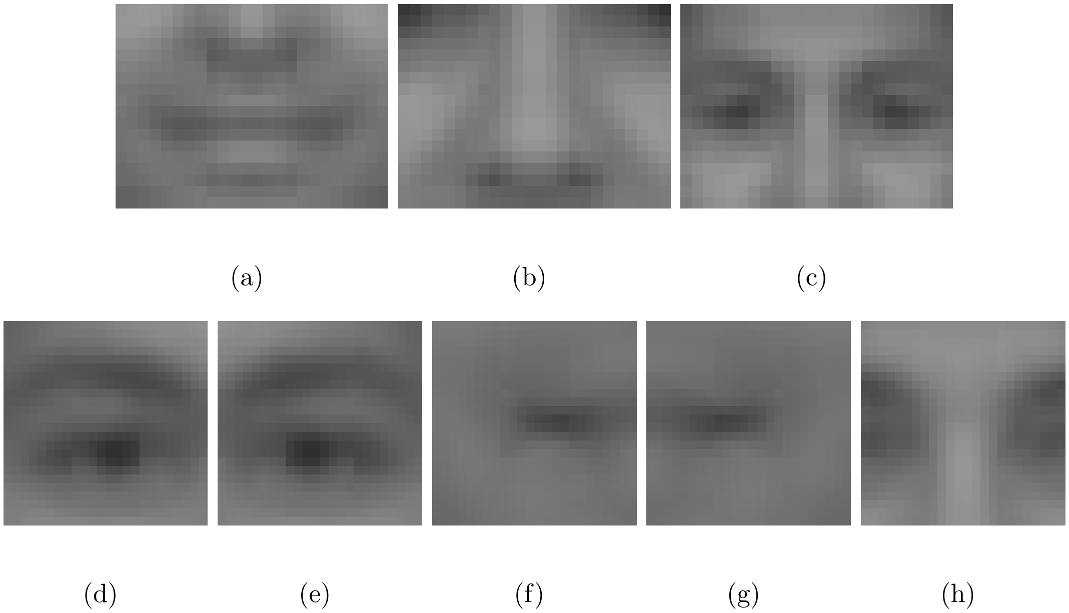
each level of our hierarchy. For example, the top level face detection filter uses Haar wavelet detectors for the eye region, nose region, and mouth region separately, allowing for partial occlusion of the face and providing more robustness to false positives generated by any single detector. Figure III.3 demonstrates this robustness to occlusion. This approach of using a naive Bayesian classifier on different facial regions is similar to that developed by Schneiderman and Kanade [73]. This is expressed mathematically in equation III.4.

$$\begin{aligned} p\left(y_{t,face}|x_t\right) &= p\left(y_{t,eyes}|x_t\right) \\ &\quad p\left(y_{t,nose}|x_t\right) \\ &\quad p\left(y_{t,mouth}|x_t\right) \\ &\quad \prod_{o \in Obs} p_o\left(y_{t,o}|x_t\right) \end{aligned} \tag{III.4}$$

III.C.2 Adaptive Template Cues

Person-specific appearance cues can also be used rather than relying solely on appearance models generated from facial database which might not include the current observed subject. In our system, we construct adaptive templates based on a IIR filtering of the detected facial regions and landmarks. These appearance models are initialized using the average appearance of the corresponding facial regions and landmarks seen in the FERET database used in constructing the general appearance cues. After each frame, the template is updated by taking a weighted sum of the current region or landmark appearance and the adaptive template itself. The weighting represents the responsiveness of the template to change.

The observation density function for the adaptive template is assumed to be a zero-mean Gaussian in the sum-of-squared pixel error between the sample and the template. By measuring the minimum of the sum-of-squared error over all of the samples, we can get a measure of the performance of the adaptive template. This is useful in determining the variance of the observation probability. Higher variances will place less emphasis on the associated observations. This is because the higher the variance, the greater the entropy of the observation function and the closer the density gets to uniform over the samples. Observations having a uniform distribution over all of the samples have no impact on the weight updates as all of the weights are updated by the same value. Figure III.4 shows the initial templates for a few of the face regions and landmarks.


(a) (b) (c)

(d) (e) (f) (g) (h)

Figure III.4: Face Region Templates: (a) Mouth Region, (b) Nose Region, (c) Eye Region, (d) Right Eye, (e) Left Eye, (f) Right Mouth Corner, (g) Left Mouth Corner, (h) Nose Bridge

III.C.3 Higher Level Cues

As demonstrated in equation III.2, we can view the information from the higher level regions as observation to the lower regions. Assuming conditional independence between the various node observations and the higher level observations, we can write our node observation density function (step 3 in figure III.1) as

$$\begin{aligned} p\left(z_t, y_t | x_t\right) &= p\left(z_t | x_t\right) \\ & \quad p\left(y_{t,haar} | x_t\right) \\ & \quad p\left(y_{t,template} | x_t\right) \end{aligned} \tag{III.5}$$

where z_t represents the parameters of the higher level region. The density function $p\left(z_t | x_t\right)$ is assumed gaussian and learned from training data associated with the relative location of the higher level region to the currently tracked region.

III.D Thin-plate Splines for Robust Feature Extraction

Thin-plate splines provide a good method to parameterize a warping transformation based on a set of fixed points. It effectively generates a minimal energy solution to a point constrained warping. This lends itself quite nicely to facial affect analysis because the facial affects can be thought of as the deviation of facial action units from a neutral zero-energy position. We show that by selecting landmark points that correspond to separate action unit areas, a good statistic for affect analysis can be generated. The thin-plate spline model is also easily separated into an affine portion that describes rigid head movement and a nonlinear portion that describes the warping induced by facial expressions. The classifier then does not need to train for rigid body head motion, allowing for reduced training sets and simplified classification systems. The formulation of the thin-plate spline model shows this separation of the affine from the nonlinear. We used the same derivation as Bookstein in his paper on principle warps [74]. This model is initialized from the location of the facial feature points in the neutral position.

$$Z = -U = -r^2 \log r^2 \tag{III.6}$$

$$r^2 = x^2 + y^2 \tag{III.7}$$

$$I_f = \arg\min_f \iint_{R^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 \tag{III.8}$$

Using equations III.6 and III.7, it can be shown that the function f (equation III.9) is a solution to the minimization problem shown in equation III.8

$$f = t_1 + a_x x + a_y y + \sum_{i=1}^{N} w_i U \left(\left\| \begin{bmatrix} x_{m,i} \\ y_{m,i} \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix} \right\| \right) \tag{III.9}$$

In equation III.9, $x_{m,i}$ and $y_{m,i}$ are the respective i$^{\text{th}}$ x and y coordinates from our model. The warping parameters W, T,and A can be calculated by equation III.10 [74].

$$\begin{bmatrix} W & T & A \end{bmatrix} = L^{-1}Y \tag{III.10}$$

where L, P, and K are defined as follows and Y contains the current positions of the tracked points padded with zeros.

$$L = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \tag{III.11}$$

$$P = \begin{bmatrix} 1 & x_{m,1} & y_{m,1} \\ 1 & x_{m,2} & y_{m,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{m,n} & y_{m,n} \end{bmatrix} \tag{III.12}$$

$$K_{i,j} = U \left(\left\| \begin{bmatrix} x_{m,i} \\ y_{m,i} \end{bmatrix} - \begin{bmatrix} x_{m,j} \\ y_{m,j} \end{bmatrix} \right\| \right) (1 - \delta_{i,j}) \tag{III.13}$$

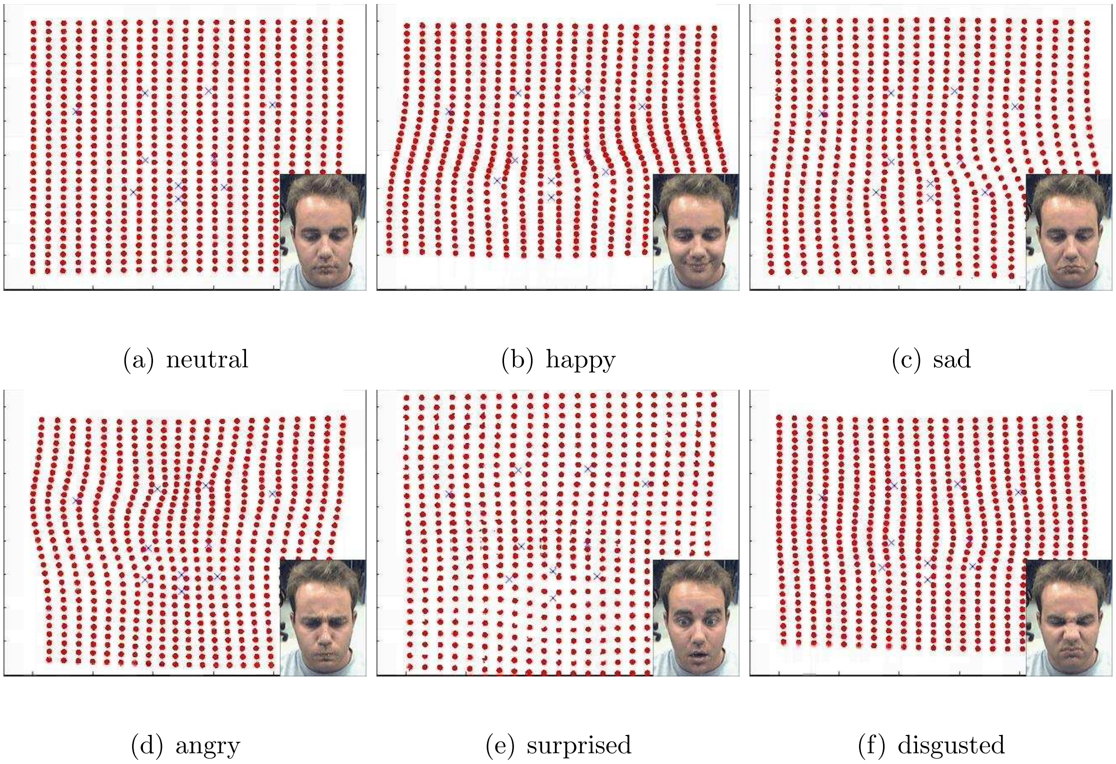
Since P and K are computed from the neutral model, L-1 only needs to be computed once when the neutral face is initialized. This allows for the fast calculation of the nonlinear warping parameters W as well as the affine warping parameters

A. It is also important to note that even though the affine warping parameters have been separated from the nonlinear parameters, the nonlinear parameters are still dependent on the affine parameters. This can be corrected easily calculating the inverse of the linear portion of the affine transform and multiplying it with the nonlinear warping parameters W. This effectively removes the dependence on the affine transformation from the nonlinear parameters. This calculation to remove the affine dependency from W in solution S is shown in equation III.14.

$$S = A^{-T}W \tag{III.14}$$

Thus thin-plate splines provide us with an efficient model for facial affect characterization by providing a closed form solution to the minimum energy warping separated into affine and nonlinear portions. This method does not require iterative techniques or lengthy operations; simply two matrix multiplications and one 2x2 matrix inverse calculation (the inverse of L is precomputed) is sufficient to generate a result. The figures below show examples of the tracked points undergoing a thin-plate spline warping.

Affine warping can give a good approximation for rigid head motion under a perspective projection for points in a plane. The error in this approximation is only due to the amount of error stemming from a perspective rather than orthogonal projection. This error can be minimized by a proper choice of camera and lens. By removing this affine component from the feature vector and by selecting facial feature points that are nearly coplanar, we can achieve invariance to rigid body



(a) neutral

(b) happy

(c) sad

(d) angry

(e) surprised

(f) disgusted

Figure III.5: Facial expression feature points (blue Xs) and grid to illustrate warping for neutral, happiness, sadness, anger, surprise, and disgust

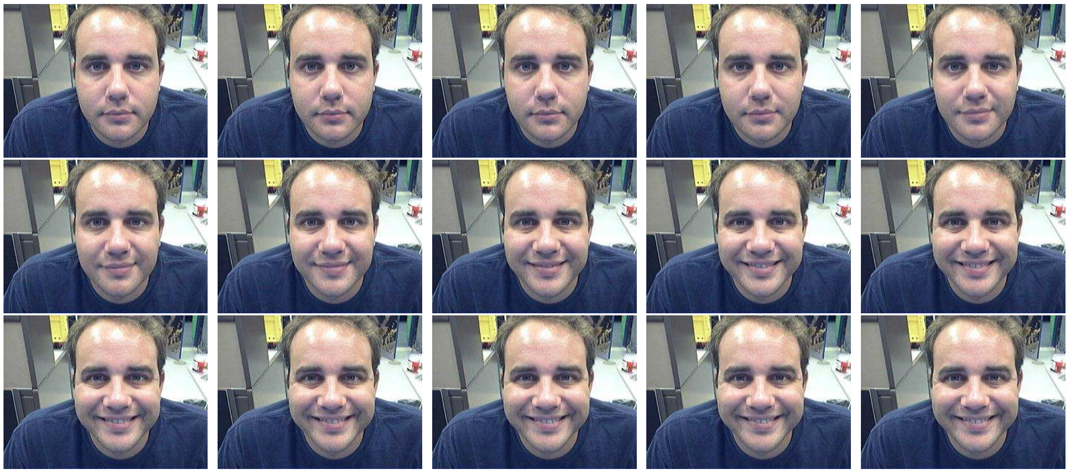


Figure III.6: Fifteen successive frames showing a transition from neutral to happiness

transformations such as head rotations and translations. A measure of the strength of a particular expression can also be calculated from the thin-plate spline warping parameters. This allows us to not only to distinguish that an expressions is being performed, but also how strong the expression is. The bending norm serves this purpose and is calculated by the equation III.15.

$$B = \text{trace}\left(WKW^T\right) \tag{III.15}$$

Figure III.6 shows the first 15 frames from a happy expression while figure III.7 shows a graph of the bending norm values versus time for this sequence.

III.E Facial Action Code Detection

As discussed in the previous section, thin-plate splines provide a good feature vector for facial expression classification. This method also works well with

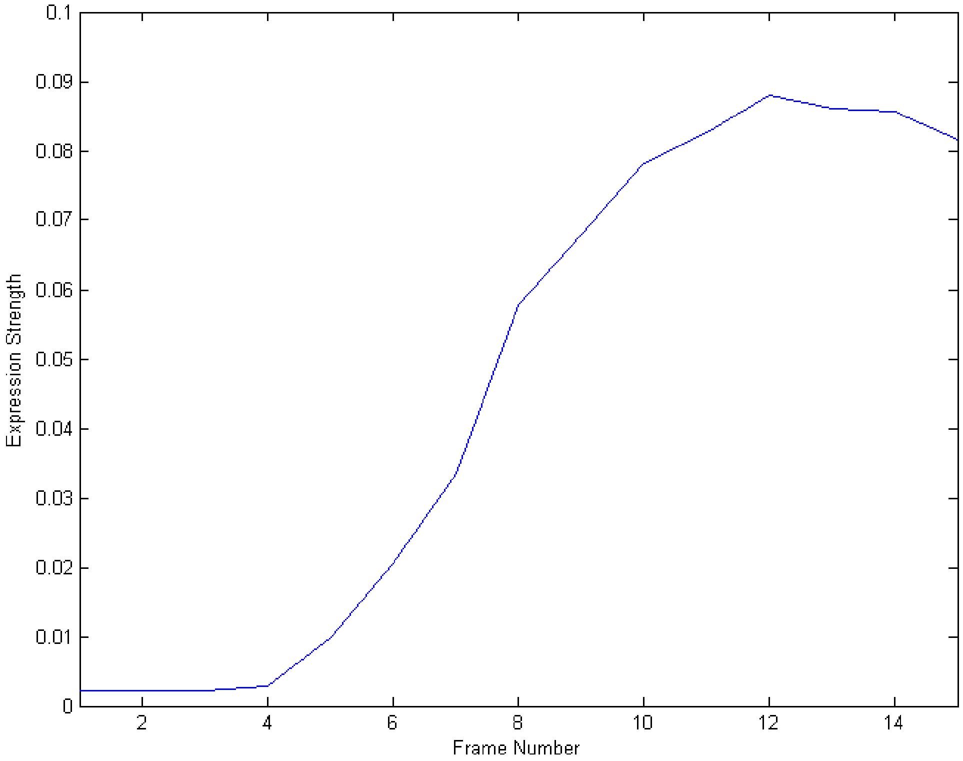


Figure III.7: A plot of the bending norm showing the strength of the expression for each frame.

our system in that the tracked landmarks can be used for control points in the thin-plate spline warping. Thin-plate splines furthermore have the usefulness of parameterizing the warping into affine and non-linear portions which is useful for creating robustness to rotation and translation. In our system, we generate a feature vector based on this non-linear warping and input this feature into an AdaBoost classifier [75] using a decision stump as the weak learner.

III.F Results for Facial Action Code Detection

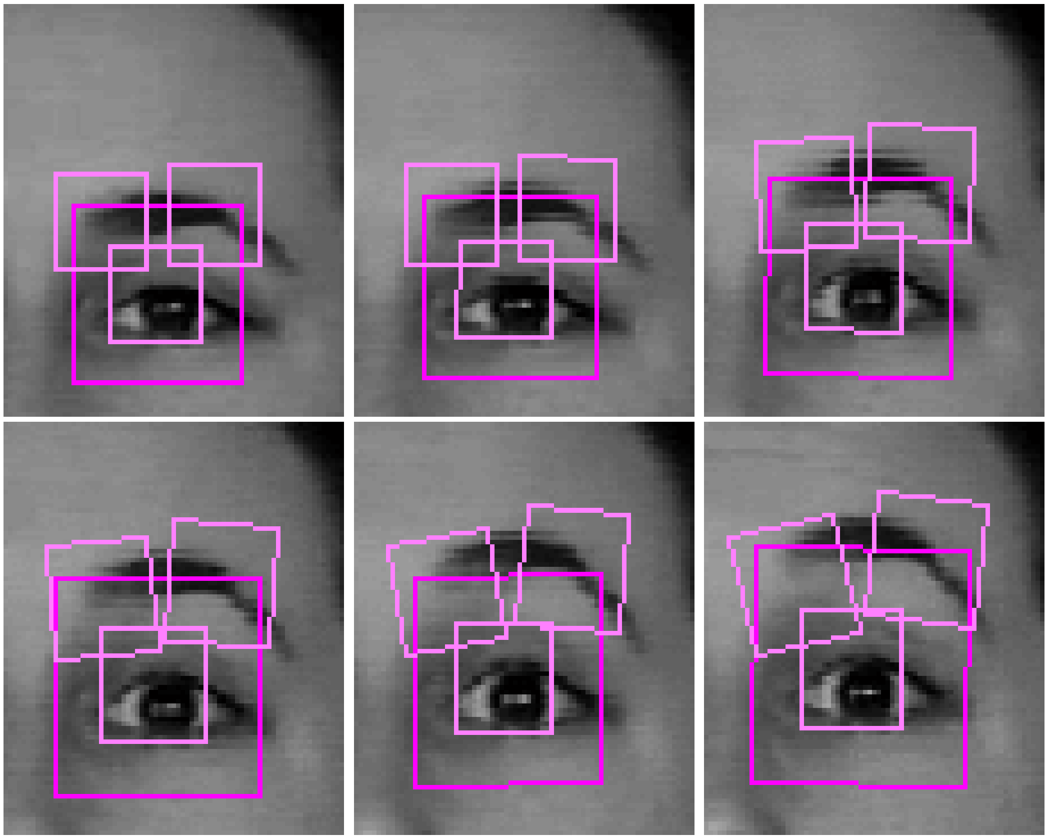


Figure III.8: Tracking results for the eye region and subregions. The lighter boxes denote subregions of the eye region.

Testing of the system was performed using the Cohn-Kanade AU-Coded Facial Expression Database [76]. The database consists of 97 subjects each performing a variety of coached facial expressions. Because of the difficulty in generating natural facial expression in a laboratory setting, it is easier to test system performance using Facial Action Codes (FACS) [77]. As described in section III.A, FACS can also be considered the building blocks for facial expressions as they can be attributed to specific muscles or muscle groups within the face. The database contains ground truth information about the specific FACS which are present in each of the sequences. Figure III.8 shows the results of the tracking on one of the test subject performing the surprised expression.

Table III.1: Test Results for Facial Action Code Classification

	Facial Action Code					
	1	2	4	5	6	7
Description	inner brow raised	outer brow raised	brows lowered	upper eyelids raised	cheeks raised	lower eyelids raised
Overall Accuracy	86.3%	88.9%	80.2%	84.0%	80.9%	79.6%
Detection Rate	88.2%	85.3%	76.0%	76.6%	76.0%	71.4%
False Alarm Rate	15.7%	7.5%	15.6%	8.6%	14.3%	12.2%
Training Sequences (# positive)	457 (132)	457 (85)	457 (140)	457 (67)	457 (100)	457 (93)
Testing Sequences (# positive)	20 (10)	20 (10)	20 (10)	20 (10)	20 (10)	20 (10)

Of the 97 subjects in the database, the system successfully initialized on all but one subject. Table III.1 shows the results for some of the FACS associated with the upper portion of the face. The results were obtained by training on a randomly generated set of sequences and testing on the remaining set of sequences. This procedure was repeated 100 times and the results were averaged. A sequence is considered a positive sequence if the specific action unit is expressed in any part of the sequence, a negative sequence otherwise. The metrics shown in the table include the overall accuracy (percentage of correct test sequences), the detection rate (percentage of positive sequences detected), and the false alarm rate (percentage of negative sequences incorrectly identified). All sequences (both training and testing) were tracked using the method described in this chapter to generate the feature vectors for classification.

Others have achieved higher recognition rates (88.5% for AUs 1, 2, 4, 5, 6, 7 tested above [78]) using additional cues beyond just landmark locations for classifying facial actions. However, we believe that our system provides a better framework for incorporating the additional cues directly into the facial action code classification and provides more robustness in difficult environments as will be shown in the next section.

III.G Driver Awareness and Attention Analysis

There is often a large discrepancy in performance between systems tested in laboratory settings and systems tested in real-world environments. This is due to a wide array of difficulties found in complex real-world environments. Specific to the driving environment, complexities such as drastic lighting changes, hard shadows, camera movement, reduced constraints on the subject, vibrations, and occlusions can all reduce system performance. It is therefore important to test systems in real-world situations to get a sense for their performance under conditions not easily simulated in the laboratory.

Furthermore, it is important to look at specific facial actions that are important to understanding the driver's attention and awareness. Important cues for inferring driver fatigue include eye blinks and yawns. Important cues for inferring driver attention and intent include head movement and head pose.

III.G.1 Results for Driver Awareness and Attention Analysis

Examples of the system operating under a large variety of the conditions described above can be seen in figure III.9.

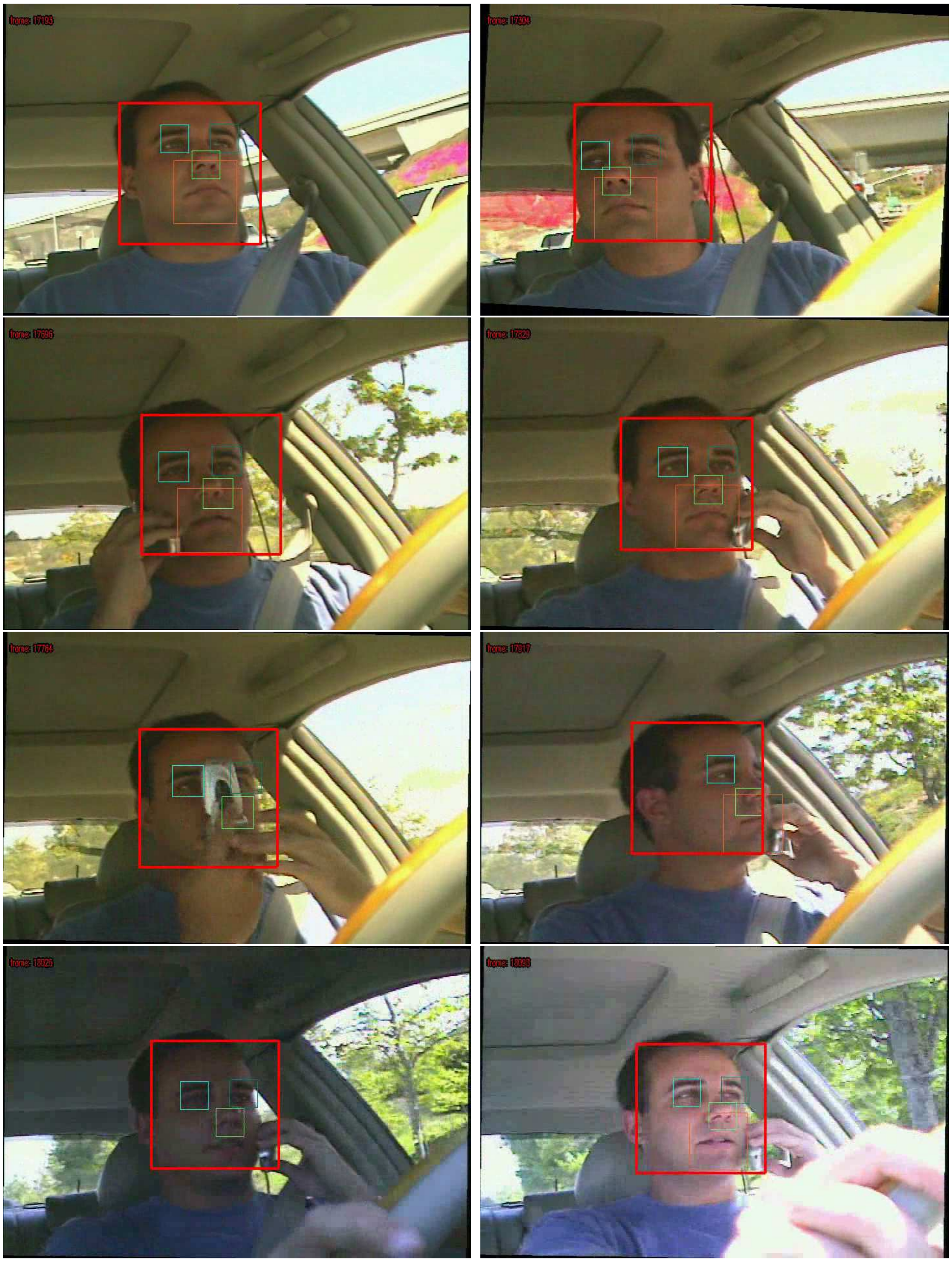


Figure III.9: Scenes showing facial feature tracking in a real-world environment.

III.H Conclusions

In this chapter we have shown a novel framework for detecting and tracking faces and facial features. Using a hierarchy of filters, we can create robustness to noise and occlusion. Facial landmark tracking is improved by using the prior information of more coarse levels of details, such as the location of the face or specific facial regions. Finally we demonstrated the value of this framework by applying it to facial action code recognition using a standard database for facial expressions. We further tested the system in a real-world environment exploring facial actions relevant to the specific task of understanding driver awareness and attention. This type of framework has far reaching interest in articulated body tracking and human-computer interface applications.

The text of Chapter III, in part, is a reprint of the material as it appears in: J. M^{c}Call and M. M. Trivedi, “Facial Action Coding Using Multiple Visual Cues and a Hierarchy of Particle Filters,” in *Proceedings of IEEE Workshop on Vision for Human Computer Interaction in Conjunction with CVPR 2006*, New York, New York, June 2006, and J. M^{c}Call and M. M. Trivedi, “Pose invariant affect analysis using thin-plate splines,” in *Proceedings of International Conference on Pattern Recognition*, August 2004, pp. 958–964. I was the primary researcher of the cited materials and the co-author listed in these publication directed and supervised the research which forms a basis for this chapter.

Chapter IV

Driver Attention and Intent Analysis

IV.A Introduction

In order to develop effective counter measures for enhancing safe and smooth operation of an automobile in traffic, it is helpful to examine the full context in which driving occurs. There are three main components of the overall driving context:

- **Environment**: including roadway infrastructure and the dynamic climatic situations,
- **Vehicle**: including ever increasing telematic devices and infotainment gadgetry, and

- **Driver**: an essential part of the human-vehicle system which needs to be maneuvered safely in the environment.

The complex dynamics of various events and interaction of various entities in the above tripartite "EVD" system components affect the overall safety of a vehicle as well as the condition of the traffic flow. For instance, properly designed roads, traffic signs, traffic regulations and policies have all been recognized as important factors in making traffic safer on the US Interstate Highways. Improved design of vehicles and safety systems, such as seat belts, brakes, and airbags are key factors in reducing injuries. The vehicle based safety systems are typically viewed as one of the two kinds. The first one is termed as "Passive". The purpose here is to minimize the severity of injuries sustained in case of accidents. Examples of these are seat belts, airbags, collapsible steering columns, and shatter resistant windshields. The second kind is "Active," which are supposed to prevent vehicular accidents. Good examples of these are anti-locking brakes. Obviously, it is more desirable to prevent an accident rather than reduce the severity of injuries. However, active safety systems pose lot more difficult and challenging problems. One of the key requirements in the design of an active safety system is the ability to accurately, reliably, and very quickly identify the conditions which would lead to an accident and to force corrective actions so that the accident can be prevented.

An active safety system has three parts. The front end of an active safety system is a sensing subsystem, which needs to provide an accurate description of the dynamic state of the EVD system. The second important subsystem is an analysis subsystem which needs to analyze the EVD dynamic state using a model based approach to compute some sort of a measure of safety underlying that particular EVD state. If this measure falls under a predefined threshold of margin of safety, then the analysis module needs to direct the active safety control unit to initiate a corrective course of action so that the vehicle can always operate within the margins of accident-free safety zone. There are some very challenging problems involved in each of the above three subsystems of an active safety system.

IV.A.1 Recent Research in Driver Behavior Analysis

Due to the importance of driver behavior to vehicle safety, many researchers have attempted to model drivers behaviors. Probabilistic models have been used to analyze driver behavior and recognize driver actions. Kuge et al. [9] used Hidden Markov Models (HMMs) to analyze driver behavior. Broadhurst et al. [79] present a probabilistic framework for determining the motion of objects in a scene and from that detecting the probability of a collision. They use Monte Carlo sampling to estimate the probability distributions. Other systems have also been developed using cognitive architectures and mental models. Goodrich and Boer [80] use mental models of drivers based to create human-centered automated systems with augmented dynamics, control, and perception based on driver's perceptions and behaviors.

Along with recognizing specific driver behaviors, others have focused on recognizing driver intent to generate advanced predictions of driver behavior. Probabilistic methods such as HMMs [81], dynamic Bayesian networks (DBNs) [82], and sparse Bayesian learning [3] have proven to be quite useful for such situations.

Oliver and Pentland [81] used Coupled HMMs to recognize specific driver behaviors and intents. Their data set included video, face and gaze detection, car and lane detection, and internal vehicle sensor data from 70 drivers. Various HMMs were trained on subsets of the different feature vectors and performance of these models was subsequently compared. Accuracy of the system for behavioral recognition as well as prediction were shown.

Sparse Bayesian learning [83] has been show to be effective in creating robust classifiers for driver intent analysis [3]. Lane change intent analysis was performed using information from a camera viewing the driver, the internal vehicle sensors, and the lane position and trajectory. It was shown that including information about the driver's head movements increased classifier performance such that equivalent prediction accuracy was achieved a half of a second sooner than only using the lane and vehicle information.

Salvucci [84] has employed knowledge-based cognitive architectures to model strategic "behavioral trajectories," which allow intelligent systems to behave similarly to human drivers or predict the future actions and cognitive load of human drivers. This type of model has also proven useful for studying driver behavior under secondary cognitive loads [85, 86] and for detecting driving maneuvers such as lane changes in real time [87].

Other systems have been developed to monitor the state of the driver. Systems such as those proposed by Ji et al. [88], solve the problem of determining driver fatigue using DBNs. Cheng, Park and Trivedi [89] present a driver activity hierarchy to break down interactions into actions, gestures, and poses. A vision system is used to generate operation triplets of $\langle$agent-motion-target$\rangle$. These Operation triplets describe driver actions and are incorporated into a DBN to probabilistically identify driver-vehicle interactions such as adjusting the radio, turning, and shifting gears.

Bergasa et al. [90] recently proposed a real-time system for monitoring driver vigilance using fuzzy systems. Smith et al. [91] developed a system for capturing driver visual attention using Finite State Automata.

Table IV.1 contains a summary of the key concepts of a variety of the above mentioned work.

Table IV.1: Comparison of various research studies in Detecting and Predicting Driver Behavior

	Objective	Inputs	Data‡	Method	Results	Comments
Kuge et al. ('98) [9]	Behavior Recognition and Prediction	Simulator Data, Lane Position	S	Hidden Markov Models (HMMs)	Recognition rates of lane changes at various times after the start of the maneuver	
Oliver et al. ('00) [81]	Behavior Recognition and Prediction	Discrete Head Pose, Vehicle Sensors, Lane Position	R	HMMs and Coupled HMMs	True positive detection rates for various maneuvers	Detection of a variety of maneuvers. Manually annotated data collection effectively removes sensor noise.
Salvucci ('06) [87]	Behavior Modeling, Synthesis, and Prediction	Simulator Data, Vehicle Sensors, Lane Position	R,S	Knowledge-based cognitive architecture	Receiver-operator-characteristic comparing simulator to real-world data	Comparison of simulator to real-world data emphasizes importance of sensor noise. Lack of driver information.
McCall et al. ('05) [3]	Intent Prediction	Head Movement, Vehicle Sensors, Lane Position	R	Computer vision feature extraction with sparse Bayesian learning classification	Comparative results of lane change intent prediction with and without head movement information	Defined lane change event as the point of crossing the lane and evaluating how soon before reliable predictions can be made.

‡S=Simulator Data, R=Real-world Data, L=Laboratory Data

Table IV.1 continued. . .

	Objective	Inputs	Data‡	Method	Results	Comments
Ji et al. ('06) [88]	Fatigue Detection	Video of Person	L	dynamic Bayesian network	Correlation to psychomotor test for fatigue	Only driver fatigue recognition, no surround or vehicle context
Cheng et al. ('05) [89]	Activity Recognition	Multi-perspective video of driver	R	image segmentation and dynamic Bayesian networks	Driver activity tracked and actions recognized	Only driver activity recognition, no surround or vehicle context
Goodrich and Boer [80] ('03)	Mental modeling for human-centered automation	Vehicle Sensors, Lead Vehicle Information	S,R	Knowledge-based mental models	Analysis of driver behavior and comparison to hypothesized model	Includes study on how users interact with knowledge-based automation systems.
Broadhurst et al. ('05) [79]	Dangerous Situation Detection	Position and trajectories of vehicles and obstacles	S	probabilistic framework for determining expected vehicle trajectories	Output of the system for a single scenario	Predicts expected paths of all objects and vehicles in the scene.

‡S=Simulator Data, R=Real-world Data, L=Laboratory Data

IV.B Attention and Intent Analysis for Lane Departure Warning

At its core, driver intent inference presents a challenging classification problem; namely, given a diverse array of multi-modal features, how can we infer or classify driver intentions. While certainly we may pose a large number of candidate intentions, as already mentioned, we will focuss on two: lane changing (either right or left) and lane keeping. This dichotomous problem is well-known to be of far-reaching significance in the realm of intelligent vehicle support systems [9].

In designing our DIIS classifier, we have at our disposal the following types of variables: *Vehicle State variables*, including gas pedal position, brake pedal depression, longitudinal acceleration, vehicle speed, steering angle, yaw rate, and lateral acceleration; *Environment Variables*, including road curvature metric, heading, lateral lane position, lateral lane position 10m ahead, and lateral lane position 20m ahead; and *Driver State Variables*, including side-to-side head movement and up/down head movement.

Given that each of these variables is a time series, the set of possible candidate features is considerably large. As such, we would like to have a method for judiciously selecting a small subset of features that are useful in classifying driver intents. Moreover, we would like our model to output class-membership probabilities rather than simply class labels. An extremely effective paradigm for this task is sparse Bayesian learning as described next.

IV.B.1 Sparse Bayesian Learning

Sparse Bayesian learning (SBL) is a powerful approach recently introduced into the machine learning literature for solving regression and classification problems [83]. The methodology relies on a parameterized prior that encourages models with few nonzero weights. As such, SBL is especially adept at pruning features, even when the number of candidates is extremely large. Moreover, the sound probabilistic underpinnings of SBL allow us to estimate class-membership probabilities as desired.

The basic form of the actual SBL discriminant functions we considered is given by

$$y(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) \tag{IV.1}$$

where $\boldsymbol{x}$ is an input feature vector (described below), the w_i's are learned model weights, and the $\phi_i(\cdot)$'s are flexible basis functions. $y(\boldsymbol{x})$ is then applied to a sigmoidal link function and a Bernoulli distribution is assumed for the probability of class C, given $\boldsymbol{x}$, i.e., $P(C|\boldsymbol{x})$. If we choose $\phi_i(\cdot) = K(\cdot, \boldsymbol{x}_i)$, where $K(\cdot,\cdot)$ is a kernel function (or feature space mapping) and $\boldsymbol{x}_i$ is a training example, we obtain the relevance vector machine (RVM), a Bayesian competitor to the popular support vector machine (SVM). However, the SBL framework is much more general in that we can consider overcomplete representations, i.e., the case where M is greater than the number of training examples. This allows us to employ multiple (complete) kernels and bases simultaneously while still controlling for overfitting. The sparsity of w is enforced by the use of prior knowledge of the parameters being inferred. Specifically, SBL models the weights as Gaussian random variables and enforces a prior distribution of the parameterization of the weights. The “most probable” weights are obtained using maximum likelihood. A more comprehensive description of SBL can be found in [83]. For our purposes, we only need to think of SBL as a principled way of learning a robust mapping from large candidate feature sets to class-membership probabilities.

At any given time t, it seems reasonable that effective driver intent inference must be based on current and previous values of the observable variables. To this end, the actual SBL algorithm is presented with temporal blocks from each of the different variables (e.g., steering angle, speed, etc.). In other words, at time t, the effective feature vector $\boldsymbol{x}(t)$ becomes

$$\begin{aligned} \boldsymbol{x}(t) \;=\; & [\ \text{LateralPos}(t), \ldots, \text{LateralPos}(t-N+1); \\ & \ \text{Heading}(t), \ldots, \text{Heading}(t-N+1); \\ & \text{etc. }], \end{aligned} \tag{IV.2}$$

where N represents the number of past values of each variable that have been stored internally. For our purposes, we selected N such that the feature vector represented a one second long sliding window of data. Thus, each feature of our feature vector $\boldsymbol{x}(t)$ represents a specific feature of our input modalities sampled at a specific time within our temporal block. The SBL algorithm then computes a sparse representation using these features to estimate the probability of an imminent lane change. This is followed by a quantile filter to smooth the result. Embedded in this formulation is the fact that temporal variations in maneuver execution are handled implicitly by SBL. This is due to the fact that the SBL approach creates a maximally sparse weighting vector that only emphasizes the features at specific times before the event that are important for the classification problem. SBL effectively discovers the temporal ordering in adjusting the weights for specific features at specific times.

SBL is particularly well suited for computer vision applications for a number of reasons. First, the SBL methodology naturally facilitates the assimilation of multiple modalities of sensor information. By sifting through numerous, possibly overcomplete, candidate inputs, SBL prunes irrelevant or redundant features to produce highly sparse representations. From a practical standpoint, this frugal representation facilitates robust, real-time, frame-by-frame driver intent classification using limited on-board hardware. Moreover, these sparse expansions permit greater interpretability, which is important as we investigate which sensor modalities are essential and which are expendable.

IV.B.2 Evaluation Metrics

Appropriate evaluation metrics are an important component of any DIIS system. Previous systems have relied heavily on classification error or similar such measures. In principle, we might like to simply report the classification accuracy over a large sample of continuous driving. Unfortunately, there are many problems with such an approach. First, there is the problem of deciding when a “true” lane change event occurs, i.e., when does it begin, end etc. While we may logically choose to define the specific lane change instant as the time when the vehicle center crosses the lane boundary, it is unclear how far in advance of this time we should consider an acceptable horizon to label as a true lane change. Additionally, this procedure ignores significant information present in the probabilistic outputs afforded by our SBL-based system. This information allows us to weigh the relative importance of maximizing the detection probability with the desire to avoid false alarms.

In addressing these issues, we developed the following performance metric. First, we created a large data set where no attempt was made to change lanes, i.e., a strict lane keeping data set. Next, we collected a second data set containing numerous lane changes maneuvers. Now because our DIIS outputs a bounded number between zero and one at every time instant t, i.e., $P\left(C|\boldsymbol{x}\left(t\right)\right)$ where C represents the class "lane change", we may always pick some threshold $\mathcal{T}$ and then decide:

$$\begin{array}{rl} \text{IF} \quad P(C|\boldsymbol{x}(t)) > \mathcal{T} & \rightarrow \text{ lane change is occurring} \\ \text{ELSE} \qquad\qquad\qquad & \rightarrow \text{ lane keeping} \end{array}$$

By varying $\mathcal{T}$ from zero to one, we may create plots of the following:

X - Probability of a false alarm at any given sample in the lane keeping data set.

Y - Probability of detection n seconds before LC in the lane change data set.

These modified receiver-operator-characteristic (ROC) curves provide substantially more information than current metrics presented in the literature. A system designer, using the information from this metric, can then decide the specific point on the ROC curve the system should operate. This metric provides information necessary to evaluate the trade-offs between higher false alarm rates and increased detection accuracy. Moreover, it naturally solves the problems raised above and, as discussed next, it addresses specific DIIS ideological concerns.

IV.B.3 Ideological Issues

The goal of our driver intent inference system is to predict when a driver knowingly or intentionally is about to change lanes. We would like to distinguish this from cases where a driver unknowingly or capriciously drifts over or near lane boundaries.

While at a high level we are distinguishing between two classes, lane keeping and lane changing, there are actually four implicit classes to consider:

i Intentional decision to change lanes followed by an actual lane change execution (*common*).

ii Alert lane keeping (*common*).

iii Intentional decision to change lanes, but the decision is modulated by traffic patterns or other concerns and the actual maneuver execution is delayed or abandoned (*less common*).

iv Capricious lane keeping where a driver unintentionally drifts near or across a lane boundary (*less common*).

With this taxonomy in place, several questions immediately come to mind with regard to existing algorithms/evaluation procedures. First, most previous works have assumed that all intended lane changes are axiomatically followed by immediate crossing of the lane boundary. But what about case (iii)? In actual driving environments, these cases will likely be labeled as false alarms even though they really are not. Our evaluation metric outlined above circumvents this problem by using a known, pure lane keeping file (i.e., no case (i) or case (iii) examples) and a separate file with numerous lane changes, either type (i) or (iii). By focussing only on the lane changes in the latter, we need not worry about falsely categorizing the type (iii) cases.

Secondly, suppose now that no examples of case (iii) exist, i.e., all lane change decisions are promptly followed by an actual lane change maneuver. Thus, we only need consider (i), (ii), and (iv). A robust DIIS should separate (i) from (ii) and (iv), which are both lane keeping events; however, a trajectory-forecasting-based approach will often separate (i) and (iv) from (ii). Moreover, the algorithms will incur a small penalty for this mistake since case (iv) is a relatively rare occurrence.

While type (iv) events may be rare in practice, they are of paramount concern in vehicle support systems.[1] Fortunately, we have found that including driver state information (e.g., head position data), facilitates bridging the gap between trajectory forecasting and driver intent inference.

[1] Of course the severity of this problem is determined by how the DIIS will ultimately be used.

IV.B.4 DIIS Results

To test our full DIIS system and compute the evaluation statistics described above, we collected significant lane keeping and lane changing datasets per the requirements set forth above. These data were collected from three drivers over large stretches of significantly curved highways. Significant curvature helps to create more type (iv)-like cases, allowing us to better see the distinction between trajectory forecasting and intent inference. Results are shown below in Figures IV.1 and IV.2 which reflect prediction accuracy with respect to various times before lane change occurrence. In both cases, *Area* refers to the area under the ROC curve while *DP* (for discrimination power) represents the point along the curve at which $1-X = Y$. We note that as the prediction horizon becomes larger, prediction fidelity decreases.

In contrast, when we exclude driver state information, results are significantly worse as expected. This is displayed in Figures IV.3 and IV.4. This is most likely because the curved nature of the highway made ideal lane keeping difficult, rendering trajectory forecasting alone insufficient for predicting driver intentions.

From these ROC curves, we can see that the classifier performance when including head movement at 3.0 seconds before the lane change is about equivalent to the classifier performance when we do not include the lane change at 2.5 seconds. We, therefore, can provide an accurate estimate earlier when head movement data is included in the feature vector. This is further illustrated by looking at a time series of lane change maneuvers. Figure IV.5 shows the lane position vs. frame number in the top graph and the lane change probability vs. frame number in

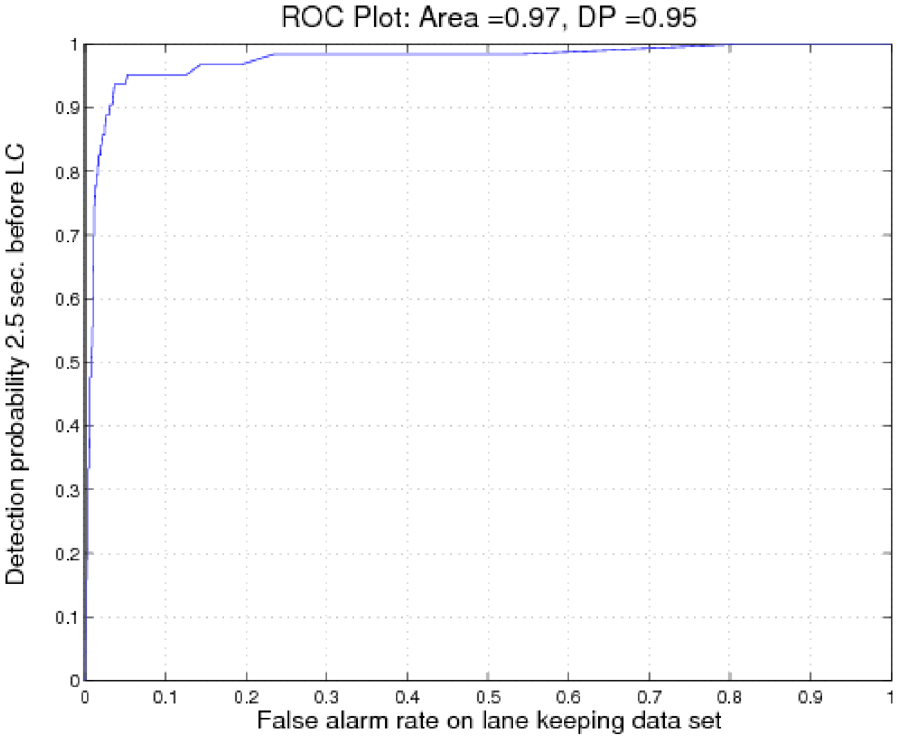


Figure IV.1: ROC curve obtained from 2.5 seconds before a lane change.

the bottom graph. The solid blue line represents the lane change probability when head movement is included and the dashed red line represents the lane change probability when head movement is not included. The graph shows the performance gain acquired when using head movement. However, in some situations, the head movement does not provide added information as is the case of the lane changed performed in combination with a previous lane change (figure IV.5, frame 12,300). In this situation, the classifier performs only slightly better than the classifier that ignores head movement. Figure IV.6 shows some frames taken from this video. Notice the significant increase in the estimated probability of a lane change using the head data apparent in frame 12132 (figure IV.6d).

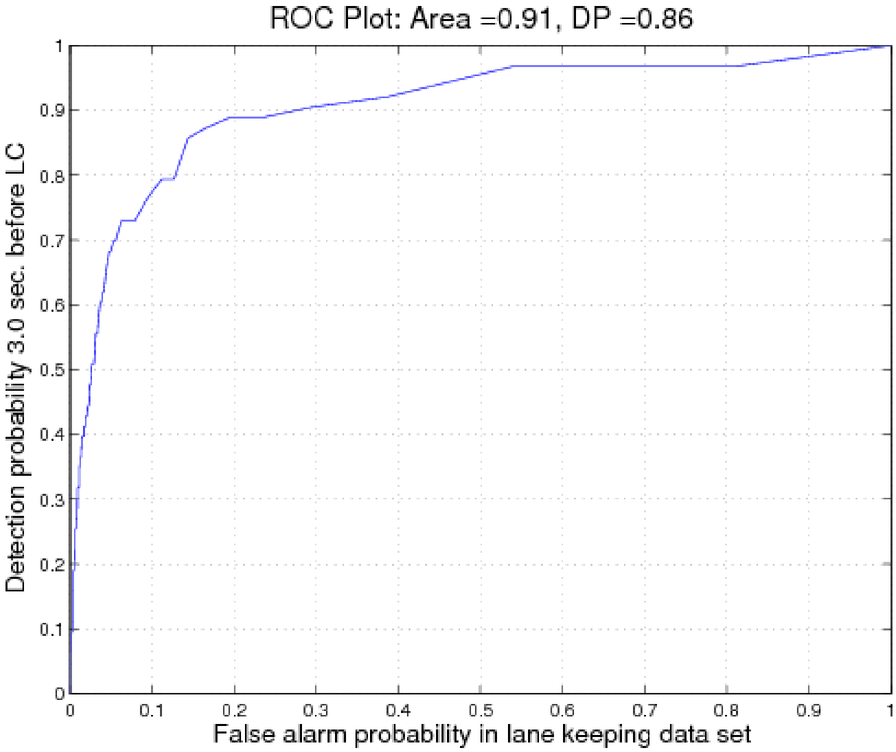


Figure IV.2: ROC curve obtained from 3.0 seconds before a lane change.

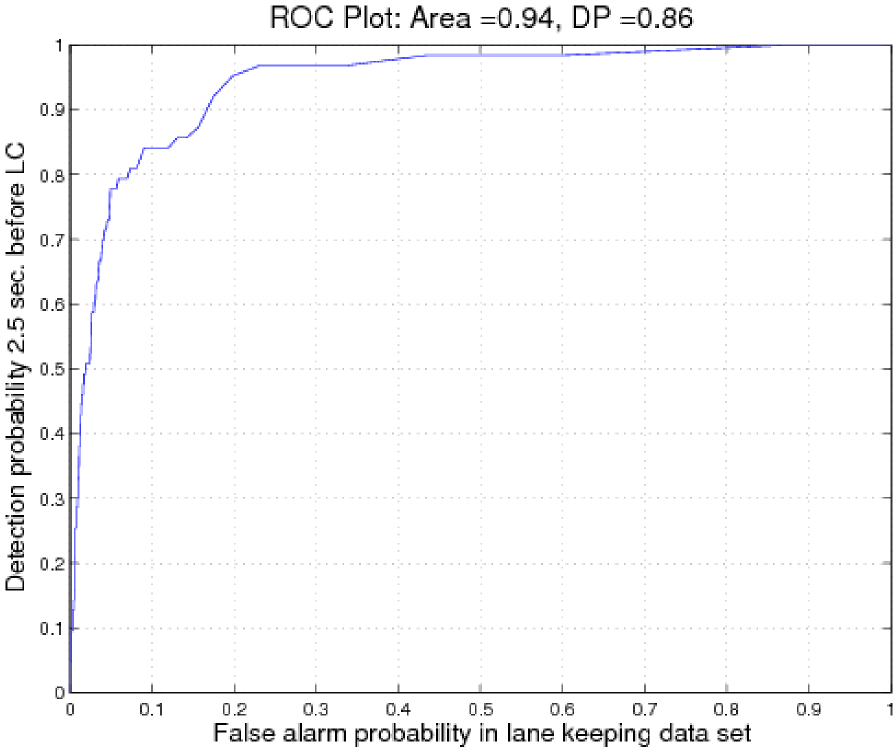


Figure IV.3: Using no driver state information (i.e., pure trajectory forecasting), ROC curve obtained from 2.5 seconds before a lane change.

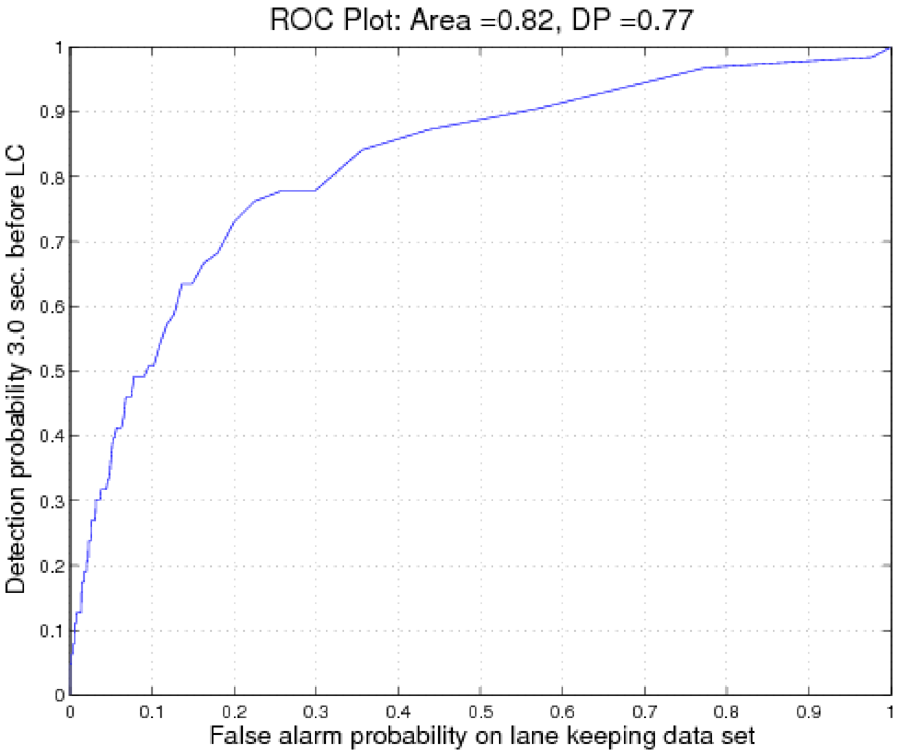


Figure IV.4: Using no driver state information (i.e., pure trajectory forecasting), ROC curve obtained from 3.0 seconds before a lane change.

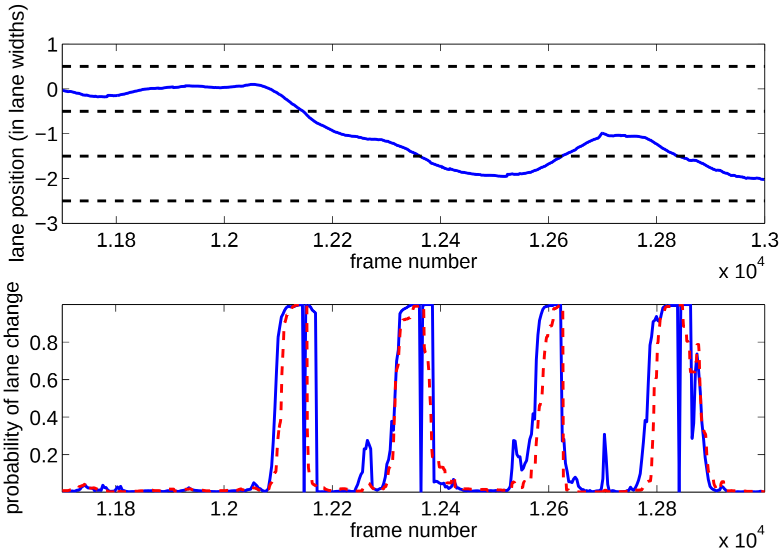
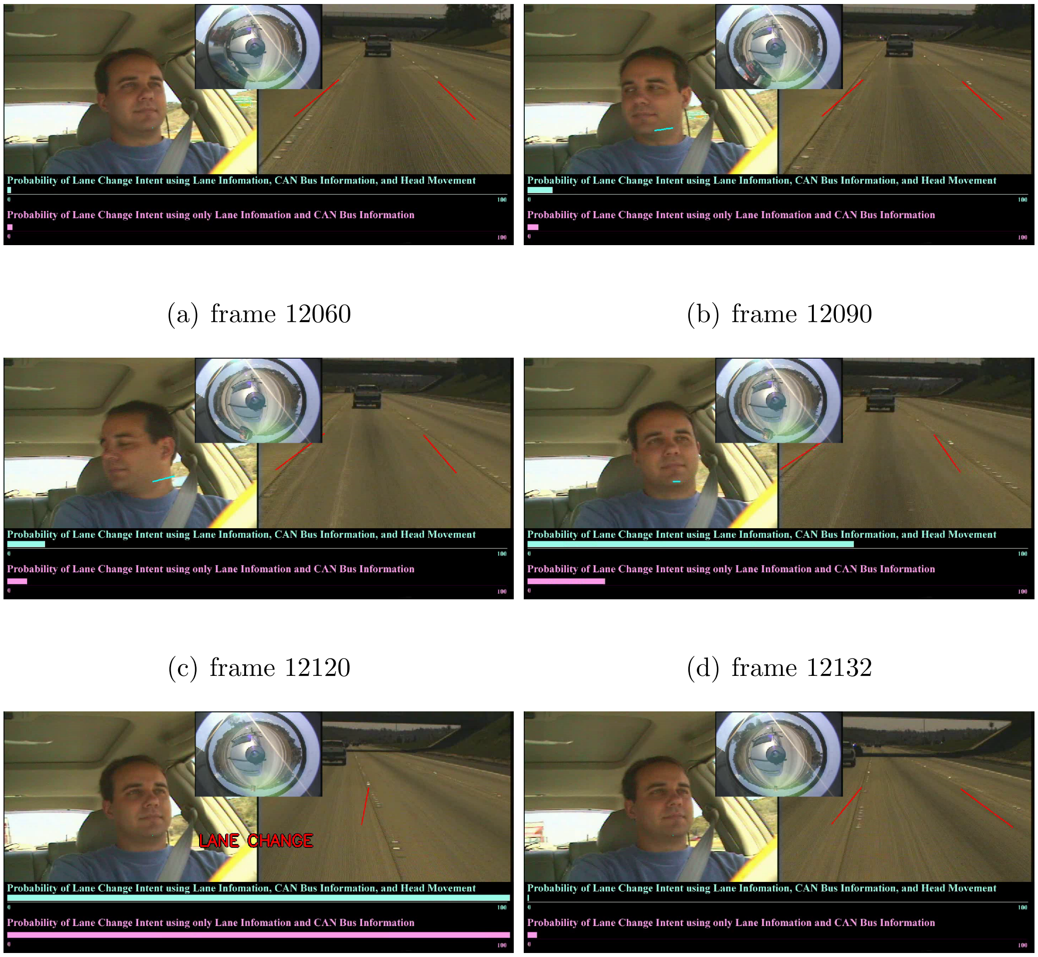


Figure IV.5: Lane position vs. frame number (top) and the probability of a lane change vs. frame number (bottom) for a driving sequence containing lane changes. In the bottom graph, the solid blue line is the probability of a lane change using head movement data and the dashed red line is the probability of a lane change without using head movement data.


Probability of Lane Change Intent using Lane Infomation, CAN Bus Information, and Head Movement
0
100
Probability of Lane Change Intent using only Lane Infomation and CAN Bus Information
0
100
LANE CHANGE

(a) frame 12060

(b) frame 12090

(c) frame 12120

(d) frame 12132

(e) frame 12180

(f) frame 12210

Figure IV.6: Frames from the video analyzed in figure IV.5. (a) shows a normal lane keeping intent. As the driver looks to the next lane in (b) and (c), the probability of a lane change intent is increased. (d) shows the lane change probability using head movement is significantly higher than classifying the driver's intent without head movement. In (e), the lane change has occurred and the probabilities have peaked at 100%. (f) shows a completed lane change and the probabilities have returned to near zero.

IV.C Driver Behavior Analysis Using Naturalistic Driving Data

In the preceding section we classified driver lane change intent prediction using a data set designed to help distinguish between intended and unintended departures. This was done by including lane keeping data that included purposeful drift to the edge of the lane to try to mimic unintended lane departures. While this can be thought of as a good approximation to unintended lane departure, there are certain side effects that are undesirable. These undesirable effects include conditioning on the data because the driver is knowledgeable of the task being monitored.

Creating a system which is trained using solely natural driving data also poses some difficulties. The main difficulty being that unintended lane departures occur infrequently, requiring an enormous amount of data to capture. Aside from collecting such a database, sifting through this data to find such instances would require an enormous undertaking. To solve this problem, we must then divide the problem into more tractable problems. We do so using a Bayesian framework.

IV.C.1 A Bayesian Network for Assessing the Criticality of Driving Situations

In order to asses the criticality of the situation, we condition the probability of a critical situation on the need for action to be performed and the intended action of the driver. In our context action can be either applying the brakes or changing lanes. The equations describing this probabilistic network are

$$P\left(C|B_s,\overline{B_d},O\right) = \\ \frac{P\left(B_s,\overline{B_d}|C,O\right)P\left(C,O\right)}{\sum_{c\in C}P\left(B_s,\overline{B_d}|C=c,O\right)P\left(C=c,O\right)} \tag{IV.3}$$

where C represents the criticality of the system, O represent our input observations, B_s represents the need for action based on the vehicle and surround sensors, and $\overline{B_d}$ represents the probability that the driver does not intend to begin an intended action. Furthermore, assuming conditional independence, the relationship between B_s and $\overline{B_d}$ can be described by (IV.4). While this "naive" Bayes assumption does not necessarily hold, in practice, this assumption can greatly simplify the system and provide good results [92]. We can simply the equations further by assuming that the driver's observable responses are conditioned solely on the driver's intended actions.

$$P\left(B_s,\overline{B_d}|C,O\right) = P\left(B_s|C,O\right)P\left(\overline{B_d}|O\right) \tag{IV.4}$$

Using this framework, we can train these probabilities using naturalistic driving data because we have simplified the problem. Driver intended actions are therefore decoupled from the surround. Assuming that the data set contains a wide range of driver behavior, including some inattention, we can create a better estimate of the driver's true intended actions. This effectively eliminates the need to specifically look for sequences which contain critical situations because of this decoupling of driver intent from surround.

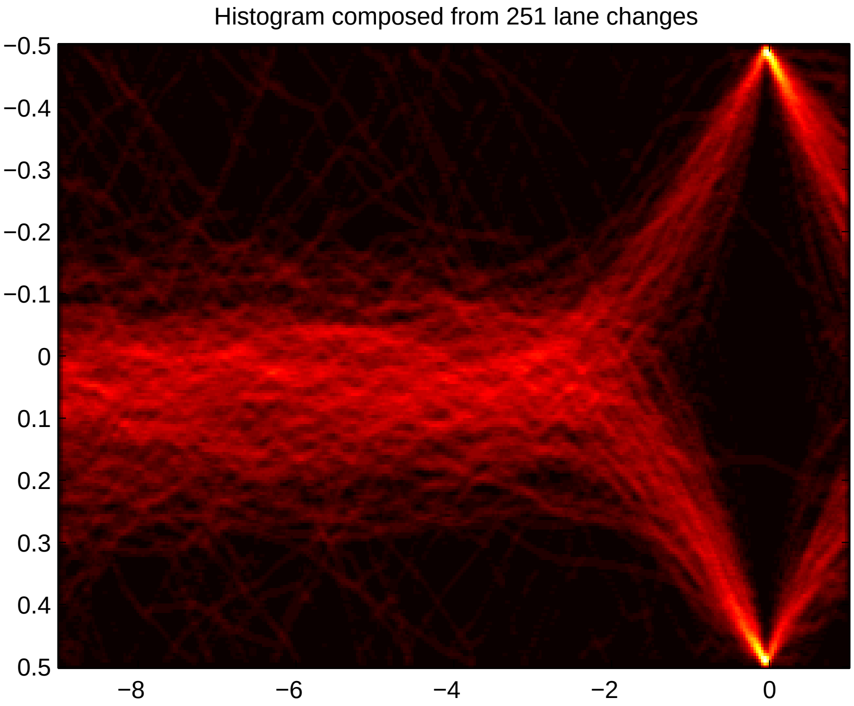


Figure IV.7: A histogram image in which each pixel's brightness represents the frequency of the observed lane position (y-axis) according to the relative time to the lane change event (x-axis.

IV.C.2 Lane Change Intent Inference From Observed Driver Behavior

Using the same sparse Bayesian learning methodology as described in the previous section, we can then train the probability of driver's intent to change lane using a significantly larger database of natural driving behavior. Before we do so, it is helpful to examine the typical patterns inherent in certain actions. Figure IV.7 shows a histogram image in which each pixel's brightness represents the frequency of the observed lane position (y-axis) according to the relative time to the lane change event (x-axis.) The lane positions are all normalized to lane width. From

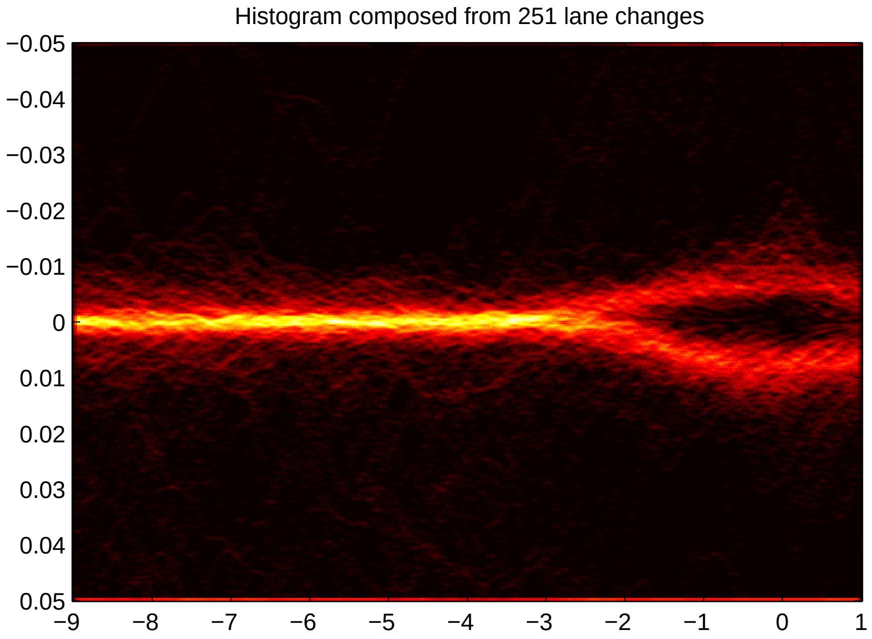


Figure IV.8: A histogram image in which each pixel's brightness represents the frequency of the observed lateral velocity (y-axis) according to the relative time to the lane change event (x-axis.

this figure and a similar histogram for lateral velocity (figure IV.8 we can see that typical lane changes begin somewhere around two seconds before the crossing of the lane boundary, but can be initiated as late as one and a half seconds before the lane crossing. This is in contrast to the relatively limited data of the preconditioned intent analysis. One might then expect to observable driver behavior indicative of a lane change anywhere from about 5-6 seconds before the event up to the event itself.

We trained the probability of lane change intent using a data set of over 250 lane change instances and over 1000 lane keeping or not intending a lane change events occurring in natural driving situations. The data was separated into training

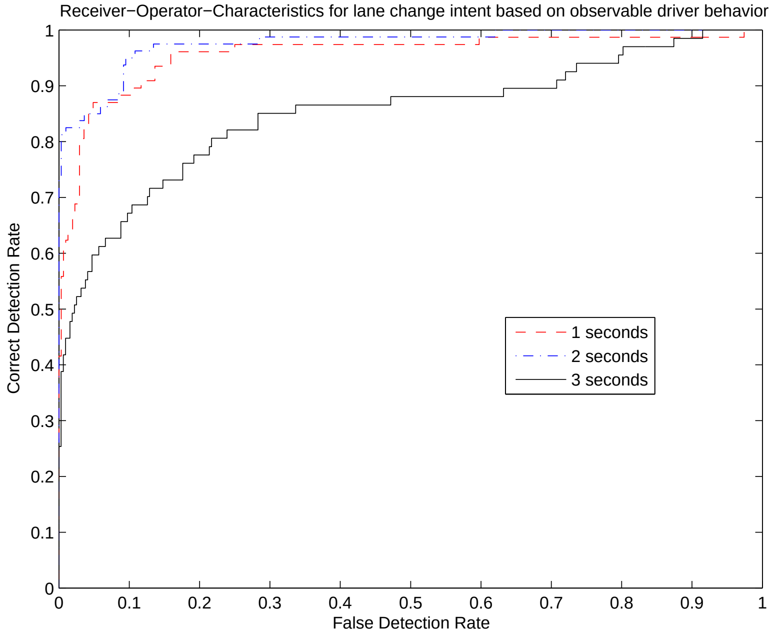


Figure IV.9: Receiver-operator-characteristic curve for lane change intent classification at 1.0, 2.0, and 3.0 seconds before the lane crossing

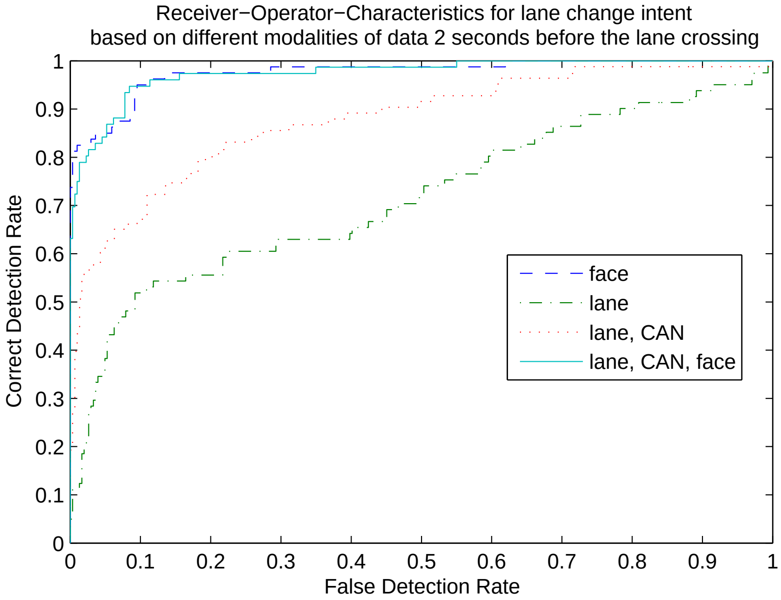


Figure IV.10: Receiver-operator-characteristic curve for lane change intent classification at 2.0 seconds before the lane crossing comparing different combinations of sensor modalities

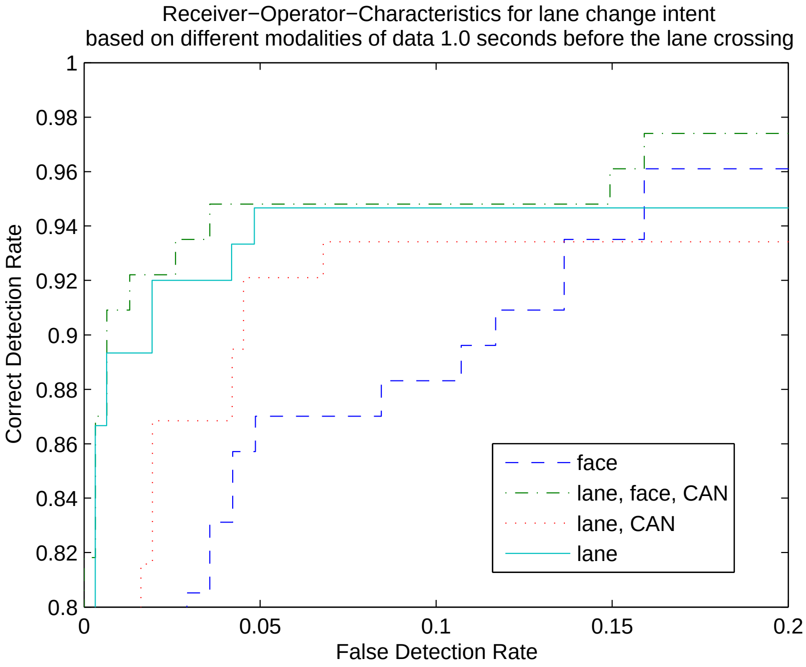


Figure IV.11: Receiver-operator-characteristic curve for lane change intent classification at 1 second before the lane crossing comparing different combinations of sensor modalities

and testing set with no overlap. Results for lane change prediction using driver head movement alone can be seen in figure IV.9. The results were obtained using a one second sliding window. We can see a degradation of the performance at one second before the lane crossing event. This is due to the fact that the observable intent is generally present before the lane change event and not always during the event. Results comparing lane change prediction using combinations of different modalities can be seen in figure IV.11.

IV.D Attention and Intent Analysis for Rear-end Collision Avoidance

In this section we will focus on the development of a specific active safety system, that of brake assistance. This will utilize sensor subsystem to extract informational cues about the vehicle, vehicle surround, as well as driver state. A novel analysis module will consider these inputs to access the need for braking and situational criticality, and will provide signals which can trigger appropriate alarms or can even be used to initiate automatic braking.

IV.D.1 The Importance of Driver Behavior to Collision Avoidance

Data from accident reports in the United States show that most vehicle accidents are at least partially caused by driver inattention. It therefore makes sense to focus advanced safety systems on mitigating the root cause of these accidents: driver behavior. Human factors studies have shown that reaction times are influenced by secondary tasks such as cell phone usage [85] or in-vehicle navigation system usage [93]. National crash surveys have also shown that driver drowsiness is also a major factor in vehicle collisions [94].

It is also important to examine how information is given to the driver in the case of warnings. Psychologists have shown the driver's respond to certain modalities of stimulus (e.g. aural, visual, or haptic) differently depending on their current cognitive load and sensory stimulation. For example, the central bottleneck theory proposes that certain brain functions and responses can be performed in parallel while certain functions such as reasoning is performed serially [95]. It is therefore important to be able to selectively alert the driver based on the driver's state and the surrounding situation. Examples of how a system might adjust its warning and corrective actions based on various situations are shown in Figures IV.12 through IV.14. In each of the figures, an intelligent vehicle is approaching a slow moving truck with different levels of awareness of the driver. The red (darker) area behind the truck represents a critical region where braking is required. The yellow (lighter) area represents the region in which the driver should be warned if they are not aware of the situation. Figure IV.12 demonstrates a situation in which the driver is unresponsive. In this case the system first warns the driver, but then applies brakes automatically when the driver does not respond. Figure IV.13 demonstrates a situation in which the driver is inattentive or distracted. The system first warns the driver. When the driver responds by initiating a braking action or or other corrective maneuver, the system recognizes the drivers actions and allows the driver to control the vehicle. Figure IV.14 demonstrates a situation in which the driver is aware. When the vehicle enters a potentially dangerous situation, the vehicle recognizes the drivers intent to brake and does not distract the driver any further.

Another important factor when considering vehicle safety systems is user acceptance of the system. A recent National Highway and Traffic Safety Administra-

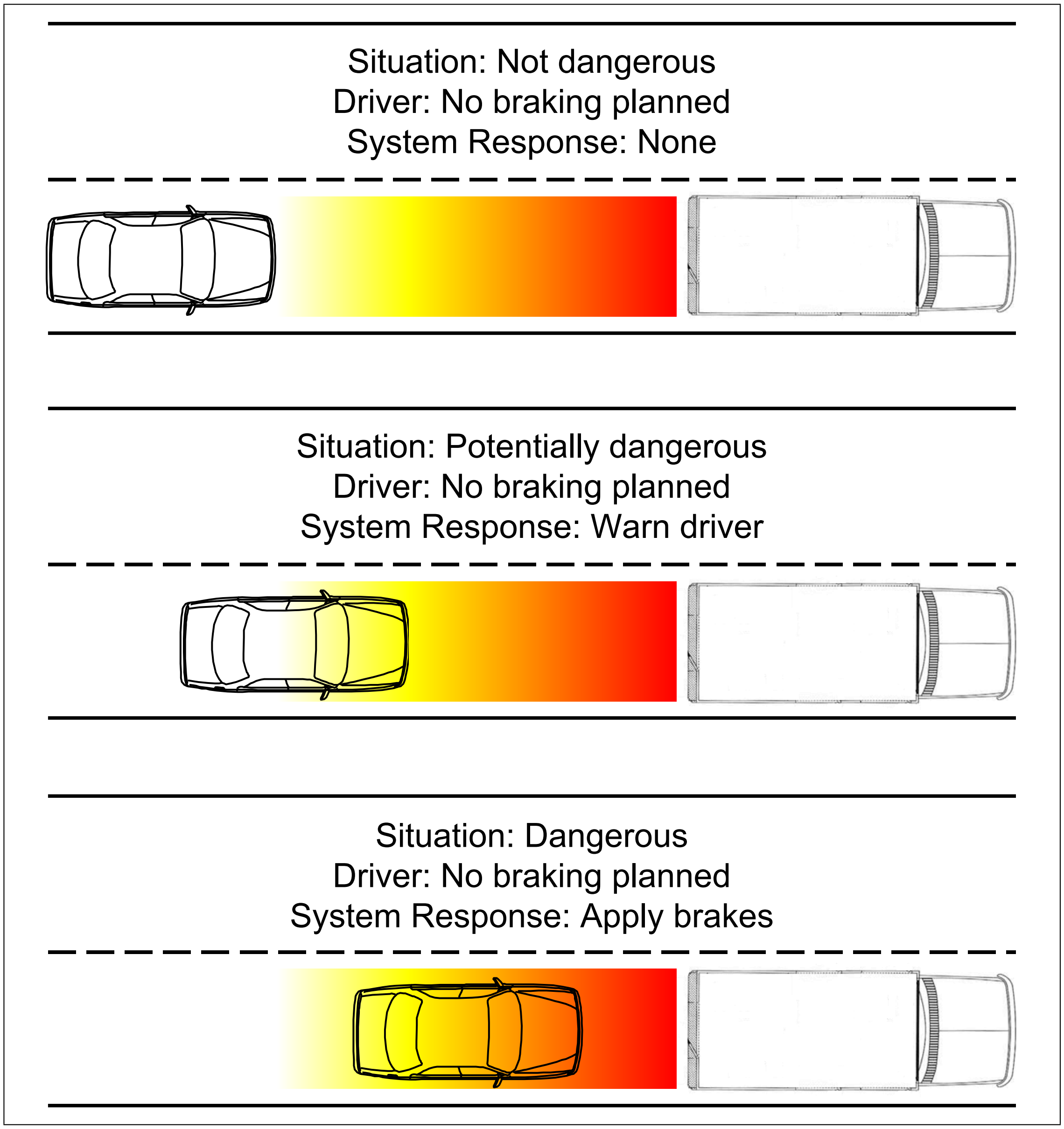

Situation: Not dangerous
Driver: No braking planned
System Response: None
Situation: Potentially dangerous
Driver: No braking planned
System Response: Warn driver
Situation: Dangerous
Driver: No braking planned
System Response: Apply brakes

Figure IV.12: System response to an unresponsive driver. The red (darker) area behind the truck represents a critical region where braking is required. The yellow (lighter) area represents the region in which the driver should be warned if he/she is not aware of the situation.

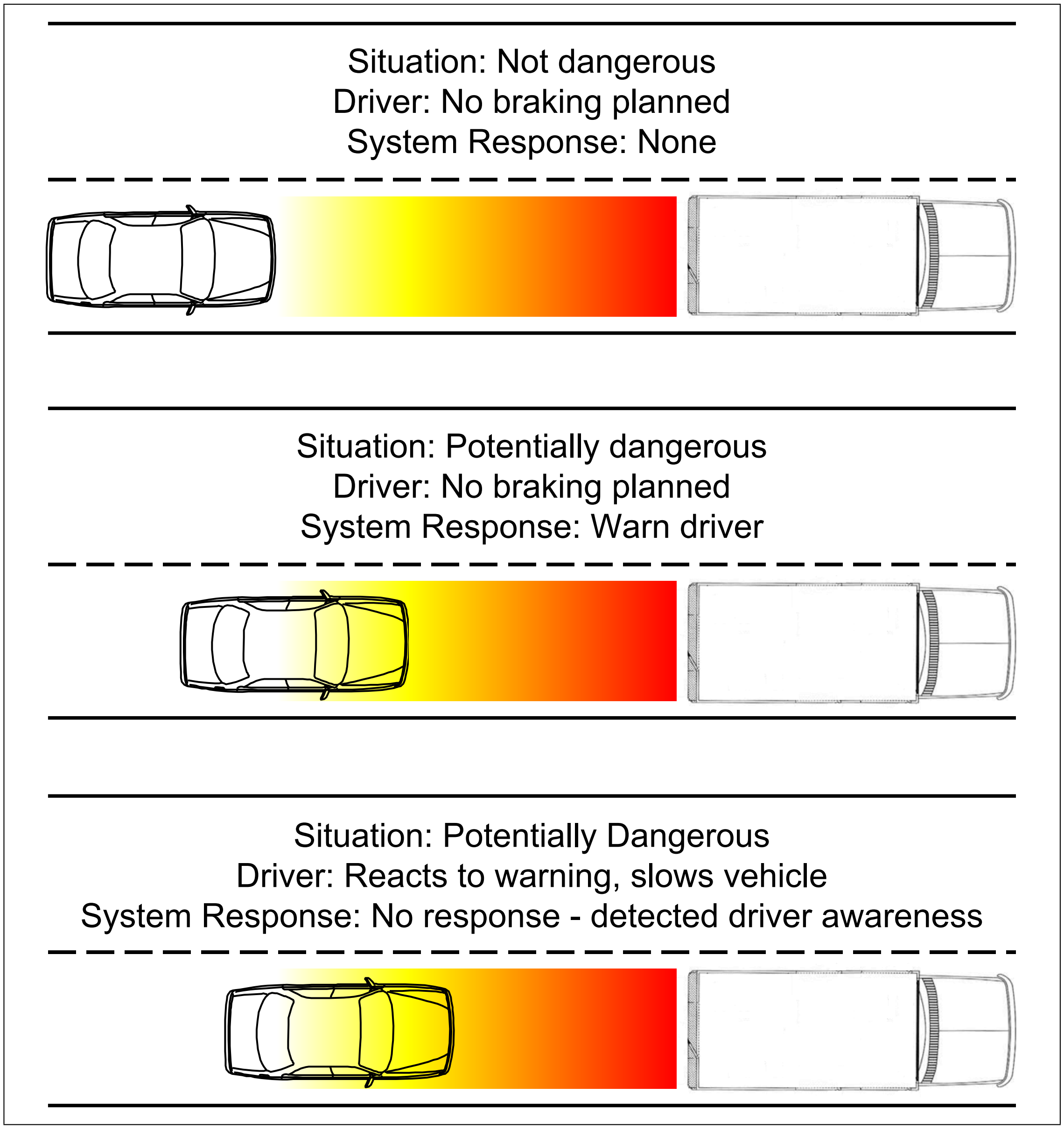

Situation: Not dangerous
Driver: No braking planned
System Response: None
Situation: Potentially dangerous
Driver: No braking planned
System Response: Warn driver
Situation: Potentially Dangerous
Driver: Reacts to warning, slows vehicle
System Response: No response - detected driver awareness

Figure IV.13: System response to an initially inattentive driver that becomes alert after being warned by the system.

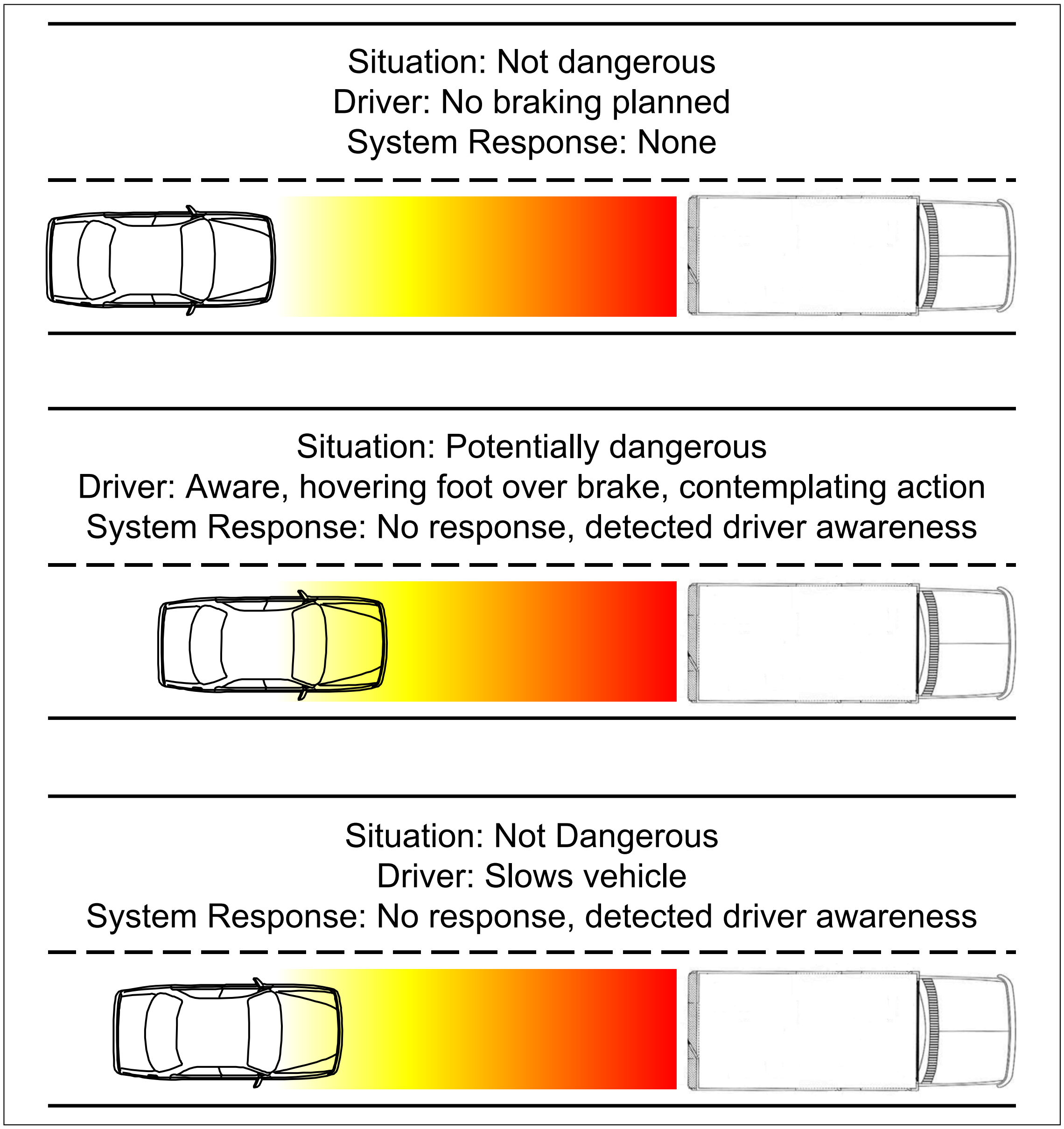

Situation: Not dangerous
Driver: No braking planned
System Response: None
Situation: Potentially dangerous
Driver: Aware, hovering foot over brake, contemplating action
System Response: No response, detected driver awareness
Situation: Not Dangerous
Driver: Slows vehicle
System Response: No response, detected driver awareness

Figure IV.14: System response to an fully aware and responsive driver.

tion (NHTSA) report [96] determined that alerts the driver believed were unnecessary contributed significantly to negative perceptions of a collision avoidance system. Similarly, they found that many of the imminent alerts were actually false alarms and also contributed to a decrease in driver acceptance. It is therefore important to take into account what the driver's planned actions are (to prevent unnecessary alerts) as well as how driver's typically react in similar situations (to determine the importance of the warning.)

IV.D.2 Related Research

Recent Research in Braking Assistance Systems

An estimated 309,000 rear end collisions occurred in 2000 according to the General Estimates System (GES) crash database [1]. Of these, 65% were listed with inattention as the leading contributing factor. Additionally, 40% of these crashes occurred under adverse environmental conditions such as wet road surfaces or poor lighting conditions. It follows that systems which recognize dangerous situations or inattentive and work to warn the driver could help mitigate a large portion of these accidents.

Longitudinal vehicle control and braking assistance systems have been studied extensively in recent years. Vahidi et al. [97] provide an extensive summary of such efforts. Many of these systems are designed for either autonomous vehicle control or Adaptive Cruise Control (ACC) and based on metrics such as the distance to the lead vehicle. Sun et al. [98] proposed a fuzzy decision making algorithm and vision perception to ascertain the "degree of exceeding safe distance." Time-to-Collision has also been used to assess danger in intelligent vehicles. Labayrade et al. [99] fused sensor data from stereo cameras and a LASER scanner in building a collision mitigation system. The system generates warnings and brakes according to preset thresholds.

Others have based their systems on learned driver behavior. Hillenbrand et. al [100] demonstrate a system that takes into account the driver's braking action by classifying braking into either normal braking or emergency braking. No predictive or direct information beyond that obtained from the vehicle is used. Biral et al. [101] examined user preferences and safety margins to generate cost-function for an optimal controller.

Systems have been proposed which attempt to address the issue of user acceptance by only intervening when absolutely necessary (e.g. when the system parameters are outside of certain satisfactory levels [102].) Further gains can be made using systems the recognize drivers intent to perform certain actions. Detecting situations in which the driver is alert and indicating through actions or body language that he intends to take corrective action can prevent the system from issuing unnecessary alerts or corrections. The system we will describe in the next section is based on this principle.

IV.D.3 Human-Behavioral Based Predictive Braking Assistance

In this section we present a situation-aware predictive braking assistance system that identifies not only the need for braking action, but also whether or not a braking action is being planned by the driver. Our system uses a Bayesian framework to determine the criticality of the situation by assessing (1) the probability that braking should be performed given observations of the vehicle and surround and (2) the probability that the driver intends to perform a braking action. To do so, we will use the framework described in the previous section. As we will describe in Section IV.D.3, these density functions are learned from the sensory inputs described in Section IV.D.3. This model is depicted graphically in fig. IV.15.

Sparse Bayesian Learning for Probability Estimation

Tipping [83] has shown that sparse Bayesian learning (SBL) is an effective technique for classification and regression on a variety of data sets. The discriminant function used in our system is shown in (IV.5). By enforcing sparsity in the learning of the feature weights (w_i) of the basis functions (K_i), both robustness to over-fitting and pruning of spurious features is obtained. Because we are examining data from a large variety of sources over a time window, we get a rather large feature vector. This makes the enforcing sparsity in our learning algorithm a very desirable trait. Furthermore, its underlying Bayesian framework allows for probabilistic outputs,

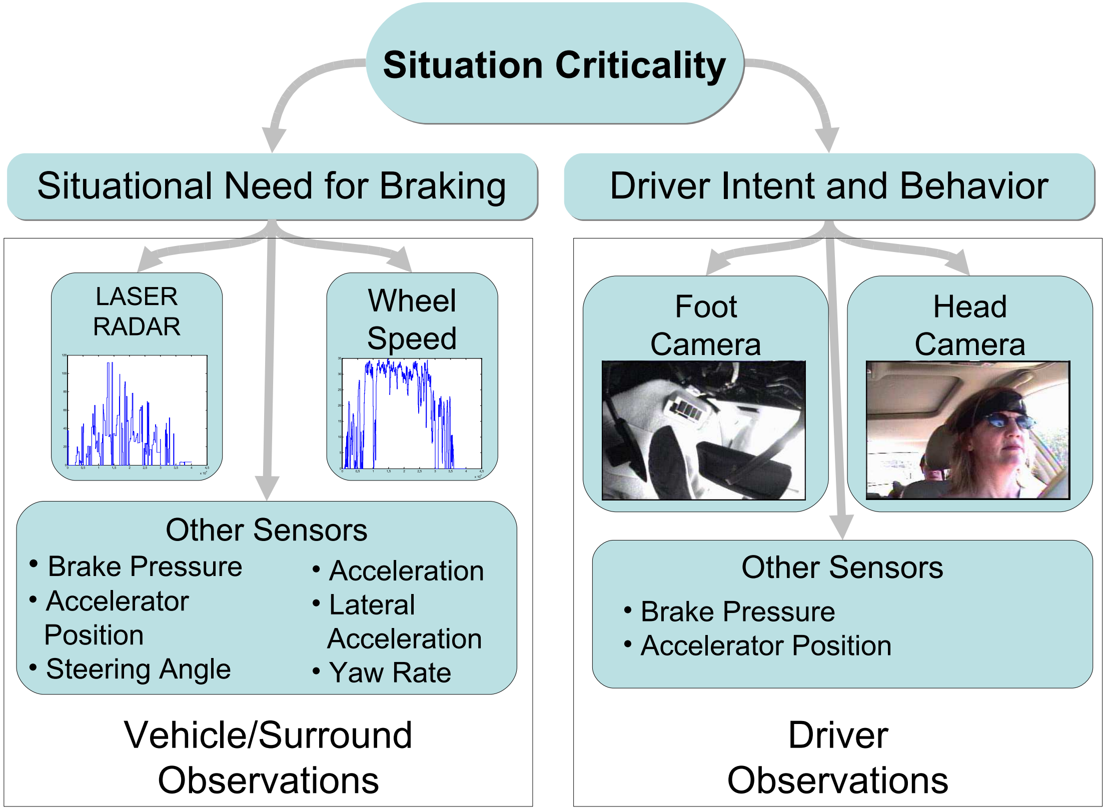


Figure IV.15: Bayesian framework to determine the criticality of the situation by assessing (1) the probability that braking should be performed given observations of the vehicle and surround and (2) the probability that the driver intends to perform a braking action.

fitting well into our Bayesian network.

$$P\left(B|O=x\right)\approx\sigma\left(y\left(x\right)\right)=\sigma\left(\sum_{i=1}^{M}w_iK_i(x)\right) \tag{IV.5}$$

where

$$\sigma\left(y\right)=\frac{1}{\left(1-e^{-y}\right)} \tag{IV.6}$$

and B represents the random variable for which we are estimating the density function (i.e. either B_s or $\overline{B_d}$). In our implementation we choose to use a radial basis function for K_i.

Sensory Inputs

The framework we have just described presents us with natural classes of sensory inputs: sensors that convey information about the vehicle state and surround and sensors that convey information about the driver's intended actions. Note that certain sensors such as steering wheel positions and pedal actions can provide information about both the vehicle and the driver and therefore belong to both classes of sensory inputs. All data was collected from real-world driving using a intelligent vehicle test bed outfitted with a variety of on-board sensors, color cameras, near-infrared (NIR) cameras, and LASER RADAR.

Vehicle and Surround Sensors

Onboard vehicle sensors obtained from the CAN data bus include:

- steering angle
- wheel speed
- longitudinal acceleration
- lateral acceleration
- yaw rate
- brake pedal pressure
- accelerator pedal position

In addition to these sensors, a LASER RADAR range finder is also installed in the vehicle giving information on the distance and relative velocity of the lead vehicle. Information provided by LASER RADAR "cut-in" sensors provide information about the relative distance to vehicles on the periphery of the current lane. For illustration, a time series of selected signals is shown in fig. IV.16.

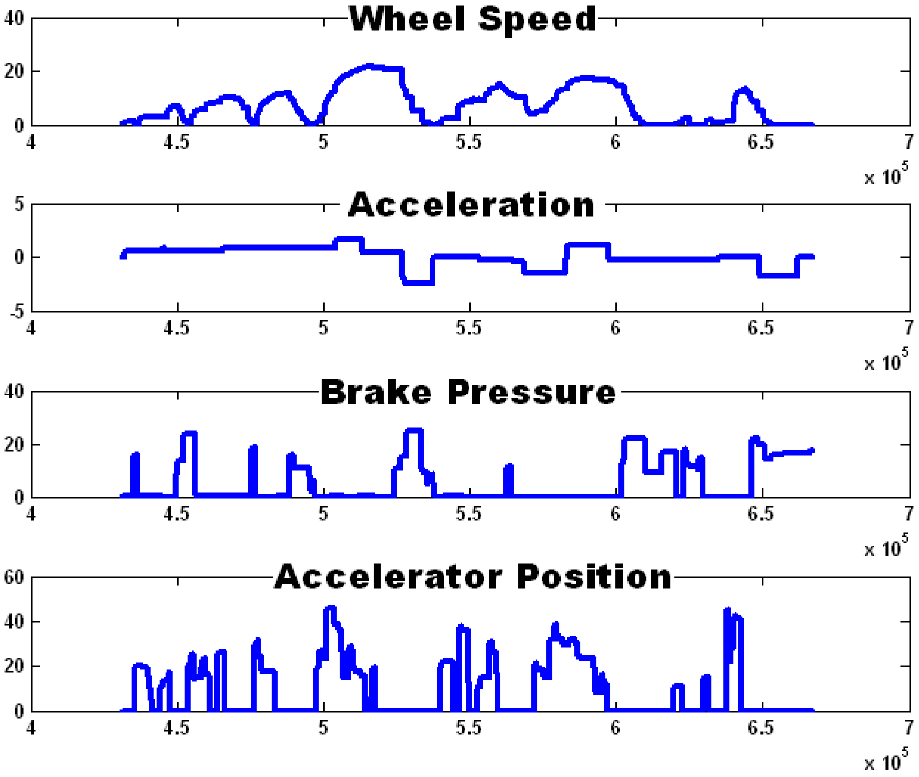


Figure IV.16: Time series of data collected from the vehicles CAN bus. From top the graphs depict speed, acceleration, brake pressure, and accelerator pedal position.

Driver Behavioral Sensors

In order to capture information about the driver's actions, we have installed a color camera observing the driver's head and a NIR camera observing the driver's feet. Example images from these sensors can be seen in fig. IV.17.

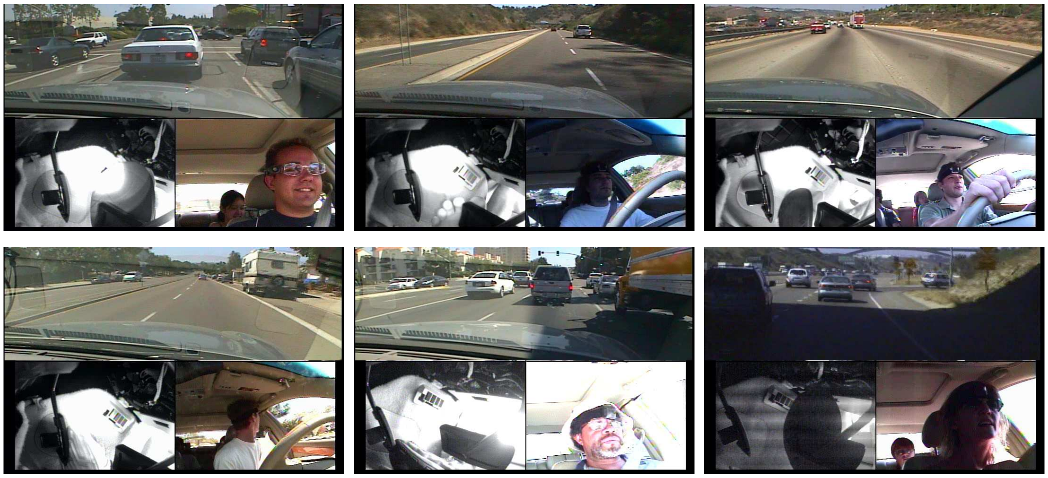


Figure IV.17: Images from the data set displaying a wide variety of driver behavior and environmental conditions. A total of 28 different subjects comprising of over 22 hours of data were used in this study.

Driver head and facial movement are then estimated using optical flow around the area of the driver's head. Also, a face detector provides information on whether or not the driver is looking forward. In this way we can capture information about the driver's head movements, lip and facial feature movements, and whether their attention is on the road ahead of them. Driver foot movements are captured using a combination of the pedal positions and pressures as well as tracking the "hovering" of the foot in anticipation of a braking action. We do so by building an overcomplete basis representation of the image motion using Haar-wavelets. This basis set is then used to generate observation vectors for the density estimation. The SBL classifier then creates a sparse weighting vector which eliminates those features which are not important to classification.

Model Training

In training our model, we first need to make an assumption about the probability distributions used. For our experiments, we have chosen to model the criticality of the situation as binomial (i.e. either critical, requiring system intervention, or non-critical). We also assume the prior, $P\left(C\right)$, is uniform, thereby simplifying (IV.3) to

$$P\left(C|B_s,\overline{B_d},O\right)=kP\left(B_s,\overline{B_d}|C,O\right) \tag{IV.7}$$

where k is a scale factor derived from the denominator in (IV.3) and $P\left(C\right)$. Combining this with (IV.4), we can see that we need to learn the probabilities $P\left(B_s|C,O\right)$ and $P\left(\overline{B_d}|C,O\right)$. However, our system further simplifies if we assume that $P\left(\overline{B_d}|C,O\right)$, or the driver's inattentiveness, is situationally independent, yielding

$$P\left(\overline{B_d}|C,O\right)=P\left(\overline{B_d}|O\right)=\left(1-P\left(B_{ca}|O\right)\right) \tag{IV.8}$$

where $P\left(B_{ca}|O\right)$ represents the probability that the driver is planning corrective action, given the current observations. This assumption is necessary in order to ensure that that the prediction of the driver's intended actions is based solely on the driver's attentiveness and not biased by the actions of attentive drivers in various situations.

To learn the density function $P\left(B_s|C,O\right)$, we look at the braking profiles for situations that are deemed critical. For our system, we labeled all situations encountered in driving which required heavy braking while the time-to-collision (TTC) metric is smaller then threshold T_{TTC}. We then separated our vehicle and vehicle surround training data into two classes (critical and non-critical) and used them to train the expected braking profiles (thereby representing the need of the situation for braking). In the density function $P\left(B_s|C,O\right)$, $O = O_{vs}$ where O_{vs} represents the observations taken from the vehicle and surround. Driving behavior can also vary greatly between drivers, making predicting comfortable safety margins between drivers difficult. To help relieve this problem, it is also possible to specify the density functions based on the safety margins computed from the TTC and vehicle dynamics. This would allow a more rigid definition of a critical event, but would remove the learned driving behavior from the estimation. Combining these two types of classifier yields a classifier in which a learned behavior classifier with a low false positive rate is used until a specific safety margin limit is reached.

We can apply a similar technique for learning $P\left(B_{ca}|O_{db}\right)$ using the input observations from the driver behavioral sensors, represented by O_{db}. Again we split the data into classes of "planning a braking action" and "normal driving without braking". Observational data is taken from a time window preceding the actual braking action.

IV.D.4 Data Collection and Results

Because our system is focused mainly on collecting natural driving behaviors of a common driving action (braking), we chose to collect real-world data for our system. This data collection was performed using the Laboratory for Intelligent and Safe Automobiles Infinti Q45 (LISA-Q) test bed [103]. The test bed collects data from up to 8 video sources, the vehicle CAN bus, a GPS sensor, the turn signals, and the LASER RADAR system. The signal sources captured and used for our system are described in section IV.D.3. Signals from each of the data sources are collected and synchronized on a main capture computer in the trunk of the vehicle. All cameras in the cockpit were designed to be as unobtrusive as possible to create a more natural driving environment.

Data was collected from 28 different drivers on varying routes of about 40 minutes to an hour each. In total, over 22 hours of driving data was used in the analysis. All The drives were performed on city and highway roads; badly congested traffic was avoided. The drivers were not told that they were being monitored for braking behavior in order to create as natural of a driving environment as possible.

Results for Predicting the Need for Braking in Real Driving Scenarios

As discussed above, estimating and accurately classifying driver behavior from the vehicle dynamics and surround is difficult because of the large variations in driver behavior. Certain drivers are more comfortable with shorter time-to-collision before initiating a braking maneuver. Fig. IV.18 shows a receiver operating characteristic (ROC) curve for predicting braking from vehicle and surround observations in real driving scenarios. This plot helps us understand the discriminative power of the classifier and the amount of variation in the safety margins of different drivers. By predicting situations in which driver's would normally start braking, we can issue earlier warnings should the driver be distracted or inattentive.

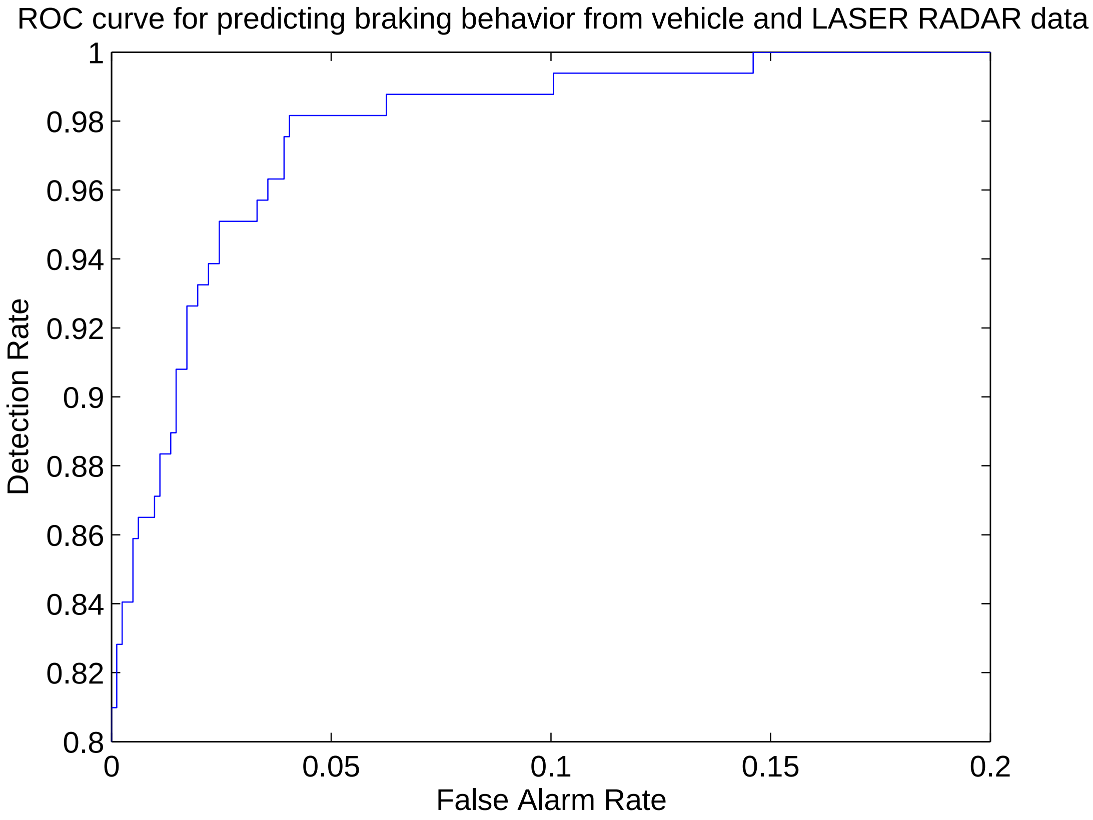


Figure IV.18: ROC curve for predicting the need for braking from vehicle and LASER RADAR data.

Results for Advanced Prediction of Driver Intent

One of the most important parts of our system is the estimation of the $P\left(B_{ca}|O_{db}\right)$ density function. An accurate prediction of braking behavior before any braking action has occurred can inform the system that the driver is aware and in control of the situation. We tested the accuracy of this part of our system using the data set as described in Section IV.D.3 by looking at the driver's behavior before any braking action has occurred. This allows our system not only to determine when the driver is braking by examining the pedal positions, but also predict in advance the intention of the driver to initial a braking maneuver. Because of our method of data collection, certain false alarms can be caused by drivers preparing a braking action but aborting before applying any pedal pressure. An extensive hand labeling of the data would be required to eliminate these types of false alarms.

Comparing the importance of information from different modalities

It is also important to examine the importance of each modality of information. Comparing performance based on specific modalities can help system designers choose which modality is the most cost effective given their specific system constraints. Figure IV.20 shows a comparison of the classifier performance using the following: pedal and steering information only; pedal, steering, and foot movement information; and pedal, steering, foot movement, and head movement information. The prediction is performed 1.0 seconds before the braking event. We can see that the system performance increases significantly at low false alarm rates when foot

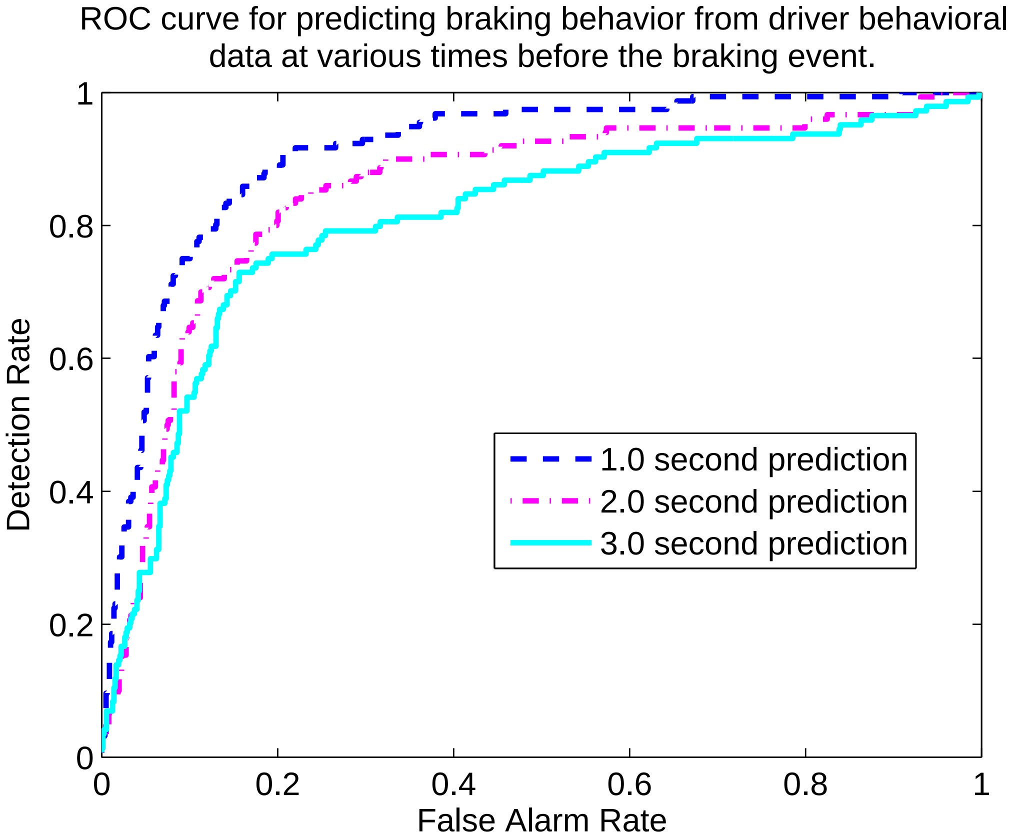


Figure IV.19: ROC curve for predicting driver braking behavior from driver behavioral sensors. The classifier was trained to predict braking action at one half second, one second, and one and a half seconds before the braking occurs.

movement information is added. However, the gain from additionally adding head movement information is relatively small. Another measure of classifier performance is the area under the ROC curve.

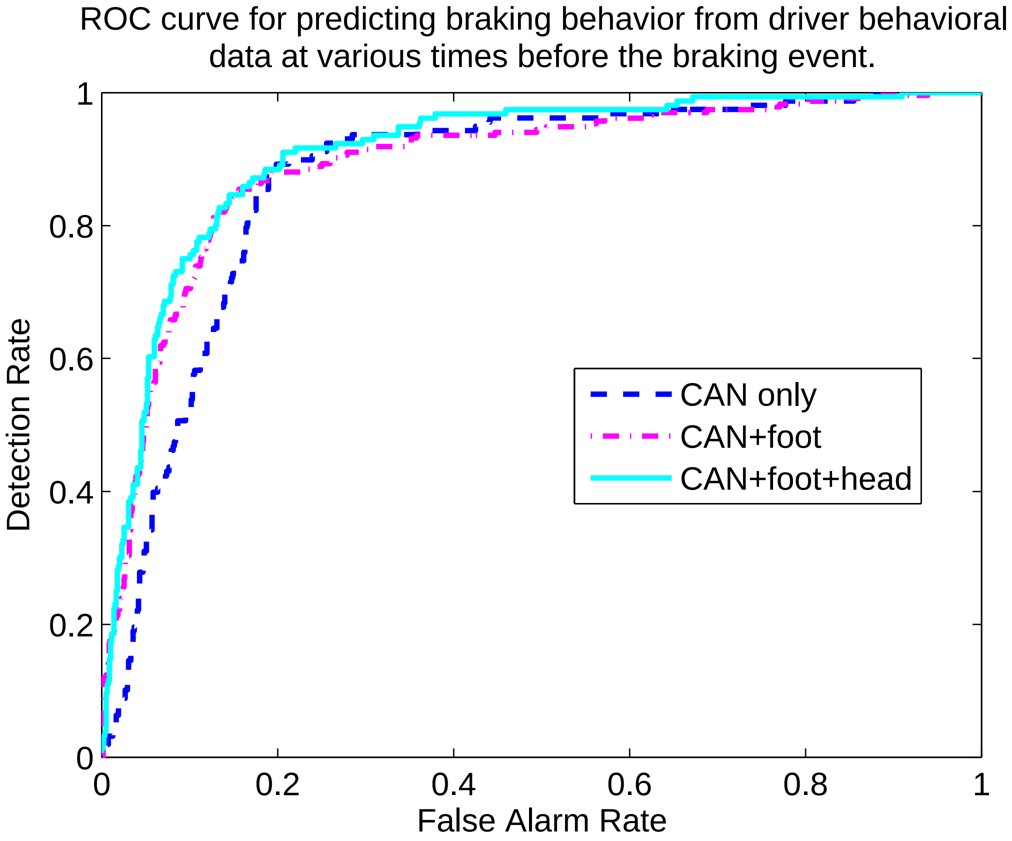


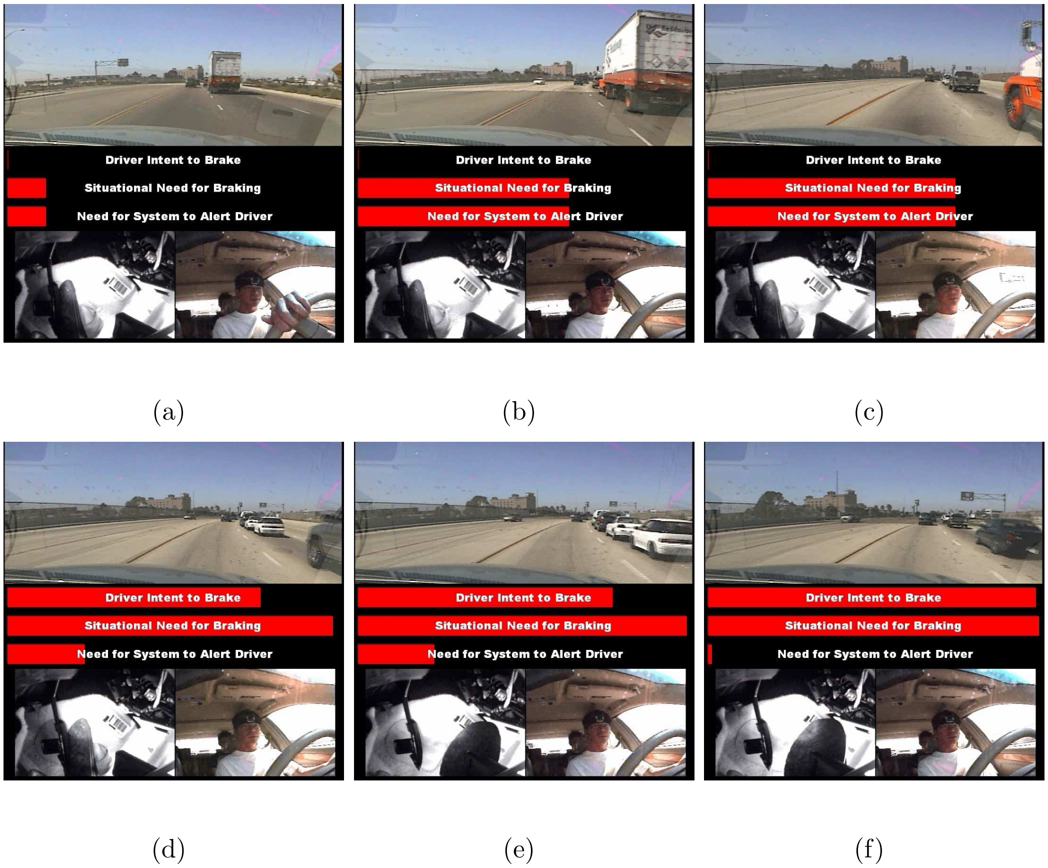
Figure IV.20: ROC curve for predicting driver braking behavior from driver behavioral sensors. The classifier was trained to predict braking action using pedal and steering information (CAN only), pedal, steering, and foot movement information (CAN+foot), and pedal, steering, foot and head movement information (CAN+foot+head).

Results of Case Studies

To help demonstrate the system, we will show the results of generated during a specific braking maneuver and analyze the system performance. For this study we have trained the classifier to detect driver intent to brake one second before braking. The probabilities of the driver's intent to brake and the need for braking based on the vehicle data are shown in fig. IV.22 and fig. IV.23, respectively. The combined probability of a critical event requiring system intervention is shown in fig. IV.24. The system first identifies the need for a braking action about 5.5 seconds before the braking takes place. When the driver's intentions to brake are observed by the system, at about 2.8 seconds before the braking action, the criticality of the situation is reduced. Cut-scenes from this sequence are shown in figure IV.21. Figures IV.21a-c show the driver accelerating around a truck towards a slow moving vehicle. The need for braking increases as the vehicle moves around the truck. Once the driver remove his foot from the accelerator and moves towards the brake (Figures IV.21d-f), the system recognizes his actions and reduces the criticality of the situation. Figure IV.26 depicts a driver approaching a car after exiting a freeway. As the foot is lifted from the accelerator and moved to the brake, the probability of the driver's intent to brake increases.

IV.E Interpretation of driver intent ROC curves

Interpretation of the ROC curves is extremely important for the design of a driver assistance system that interacts with the driver. One of the most important goals for such system is to reduce the occurrence of incorrect warnings as much as possible. This means that the amount of situations in which the driver intents an



(a) (b) (c)

(d) (e) (f)

Figure IV.21: Selected video frames from the foot camera, head camera, and the forward viewing camera as well as situation, driver, and alert probabilities during a braking action. Subfigures a-c show the driver accelerating around a truck towards a slow moving vehicle. The need for braking increases as the vehicle moves around the truck. Once the driver remove his foot from the accelerator and moves towards the brake (Subfigures d-f), the system recognizes his actions and reduces the criticality of the situation.

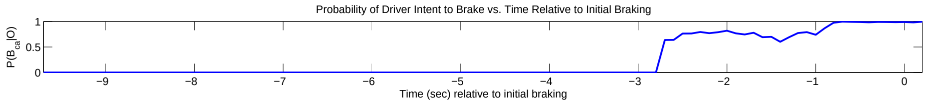


Figure IV.22: Probability of Driver Intent to Brake vs. Time Relative to Initial Braking

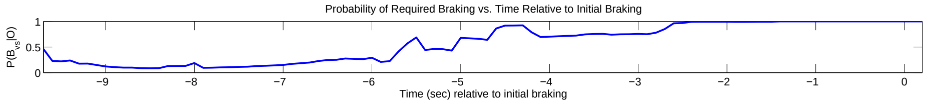


Figure IV.23: Probability of Required Braking vs. Time Relative to Initial Braking

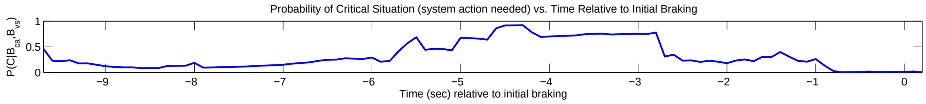


Figure IV.24: Probability of Critical Situation (i.e. system action needed) vs. Time Relative to Initial Braking

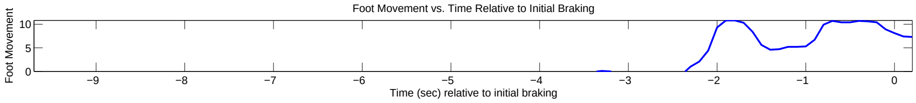
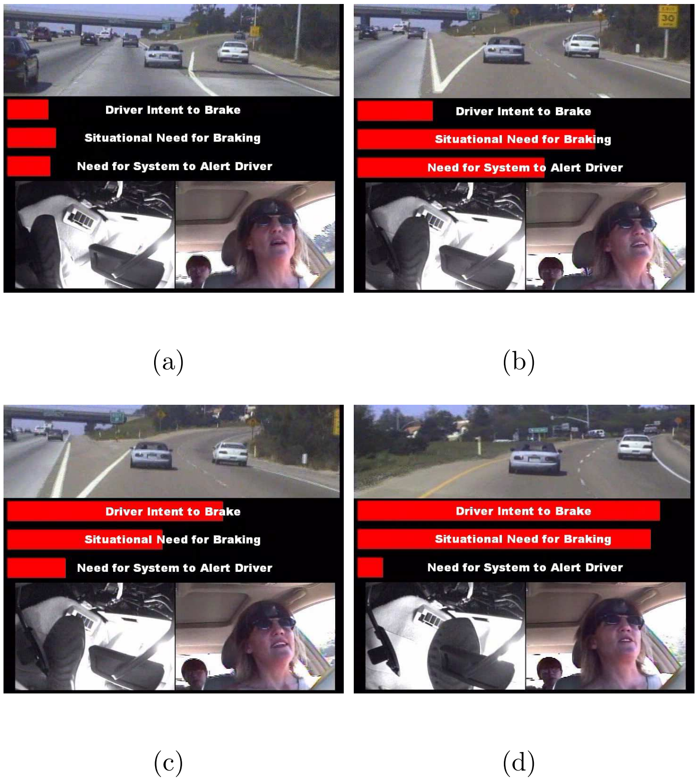


Figure IV.25: Foot Movement vs. Time Relative to Initial Braking



(a) (b)

(c) (d)

Figure IV.26: A scene showing a driver approaching a car after exiting a freeway. As the foot is lifted from the accelerator and moved to the brake (subfigures b-d), the probability of the driver's intent to brake increases.

action but this action is not recognized by the system must be kept to a minimum. It is therefore important to minimize the missed detections or equivalently maximize the detection rate of the system. According to the ROC curves, this can occur at the expense of a higher false detection rate. A false positive detection also has severe consequences in a driver assistance system. While for a large majority of the time these false positives will not effect the situation, a false positive during a critical situation would cause the system to augment it behavior because it incorrectly recognized the driver's intent (e.g. reducing the priority of a warning because the driver was classified as making an evasive maneuver.) To help optimize this interaction, a driver assistance systems could be designed to use varying levels of alerts, each using the a point on the ROC curve corresponding to an empirically determined user acceptance of false alerts.

IV.F conclusions

In this chapter we have presented a complete system for lane departure and rear-end collision avoidance based on driver attentiveness. First we introduced a method of detecting lane departures and assessed the performance of the system using information about the driver and vehicle. Following this we introduced a brake assistance system that similarly compares the drivers actions to the vehicle surround. By analyzing the driver's intended action based solely on his observable actions, we can find discrepancies between the predicted driver's intent and the predicted vehicle trajectory. This allows us to using real-world training data, containing only a few critical events, to attempt to assess the criticality of the situation.

Another key aspect of any driver assistance system is its interface with the driver itself. While we have shown the observational portion of the system, the types of warnings purveyed to the driver are no less important. One benefit of the probabilistic framework on which our system is based is that the interaction with the driver can be manipulated based on fuzzy indications of how critical the situation is. The modality and intensity of the alert can then be augmented based on the driver and his/her surroundings. An example of this might be that at higher speeds, where collisions could prove more severe, the system might warn the driver at lower probabilities of critical situations. Another useful possible application of this continuous valued criticality parameter would be in tailoring the system to specific users. If a user is known to respond poorly to certain modalities (by entering critical situations despite being warned) or become annoyed by false alerts, the system could adjust the modality or thresholds for warnings accordingly.

However, determining the optimal warning levels, the modality used in warning the driver (aural, visual, haptic), and the proper times in which to warn the driver require much more controlled studies. While data collection and behavioral observations can be made using a real vehicle in real driving scenarios, simulator studies would allow the system to be evaluated in more critical situations.

The text of Chapter IV, in part, is a reprint of the material as it appears in: J. M^{c}Call and M. M. Trivedi, “Driver Behavior and Situation Aware Brake Assistance for Intelligent Vehicles," in *Proceedings of the IEEE*, submitted for review, and J. M^{c}Call, D. Wipf, M. M. Trivedi, and B. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," *IEEE Transactions on Intelligent Transportation Systems*, in press. I was the primary researcher of the cited materials.

Chapter V

Conclusions

In this dissertation we have introduced a novel method for fusing predicted driver behavioral information with vehicle and surround information. We have provided a framework for creating a driver support system using visual cues from the sensor level to the attention and intent interpretation level. The framework allows for the assessment of the criticality of the current situation and the need for intervention by an intelligent vehicle safety system. Data for training and testing of the system was compiled from real-world driving scenarios, thereby tuning the system to common braking behaviors. By using sensors that capture the driver's intended actions as well as the vehicle and surround information, we can create systems that are more complementary to the driver's actions and less prone to annoy the driver. Individual components of the system were evaluated and a demonstration of the system as a whole was shown.

The major contributions of this research include: a framework for training and analyzing driver intent based on real-world data; a framework for analyzing situation criticality based on the surround situation and driver intent (this framework allows for the use of training data that may or may not contain any actual critical events;) algorithms for robust extraction and tracking of visual cues for lane markings; algorithms for fusion and tracking of facial features in noisy environments; and a system design and methodology for real-world data collection in an intelligent vehicle. These combined systems therefore form the basis of a complete human-centered driver assistance system with the exclusion of the feedback interface to the driver. By looking at each of these elements combined into a real-world system, we can get a better understanding of the performance, challenges, and strengths of such systems.

V.A Future Research Directions

Future extensions of this work include the possibility of creating a more complex dynamic Bayesian network for intent and attention analysis. By further conditioning the data in such a model, smaller amounts of training data could be needed to accurately classify certain intents. A more complex model of the interaction between intent and attention could be used, providing more insight into the relevance of specific input modalities. For example, a hidden Markov model tracking the focus of the drivers attention might provide better insight into how head pose factors into the braking task. Similarly, we could determine the probability that the

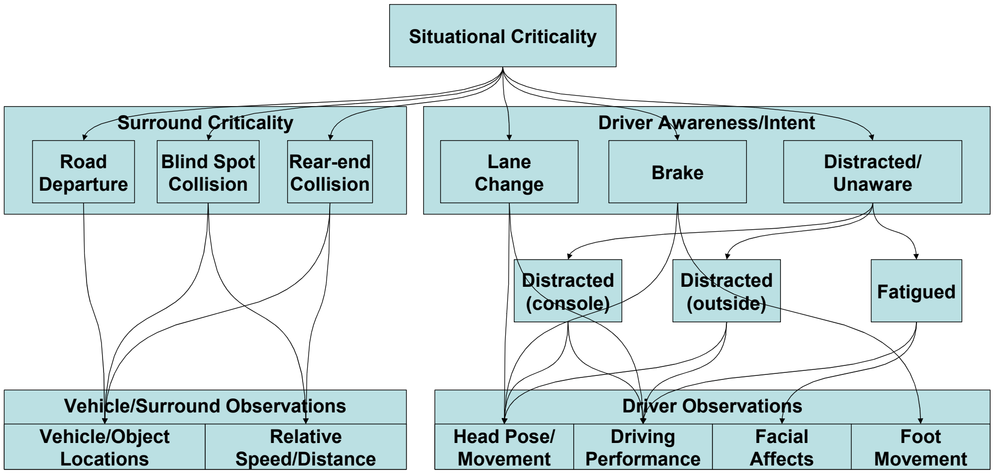


Figure V.1: A dynamic Bayesian network in which driver intent and attention are assumed conditionally independent of the situation criticality.

driver is focused on the radio or center console and use this as an observation of the drivers intent that is conditionally independent of the other observed cues. Figure V.1 depicts a possible network based on these ideas.

Other extensions would be to include the driver feedback into the system, thereby closing the loop of the human-machine interface. This would help understand better how drivers react to specific types of alerts and what the best feedback modalities are based on the wide variety of situations one encounters while driving.

Appendix A

The Lisa-Q Intelligent Vehicle Test Bed

A.A Introduction

Many researchers are currently working on the problem of making our highways safer for driving. One method of providing this safety is to examine the link between the driver and the automobile to see what modifications can be made to the automobile to assist the driver. This includes advanced warning systems and improved user interfaces. In order to warn the driver of potentially dangerous situations such as vehicle cut-ins, driver distraction, driver drowsiness, problems with the vehicle, and unintended lane departures, a complete vehicle context is required. This complete vehicle context includes the vehicle surroundings, vehicle interior, and vehicle state. At the backbone of this research is the creation of a human-centered intelligent vehicle that captures this complete context and interacts with its occupants to inform or warn the driver of potentially dangerous situations. In order for this research to progress, an intelligent vehicle test bed must be created to accomplish the tasks of 1) collecting data on driver behavior in order to best respond to various situations and better understand the drivers intent, 2) testing algorithms for sensing the vehicle context, including its interior, and feeding that back into the vehicle's user interface, and 3) collecting complete surround data (both interior and exterior) in order to create an annotated ground truth data set for training of intelligent systems.

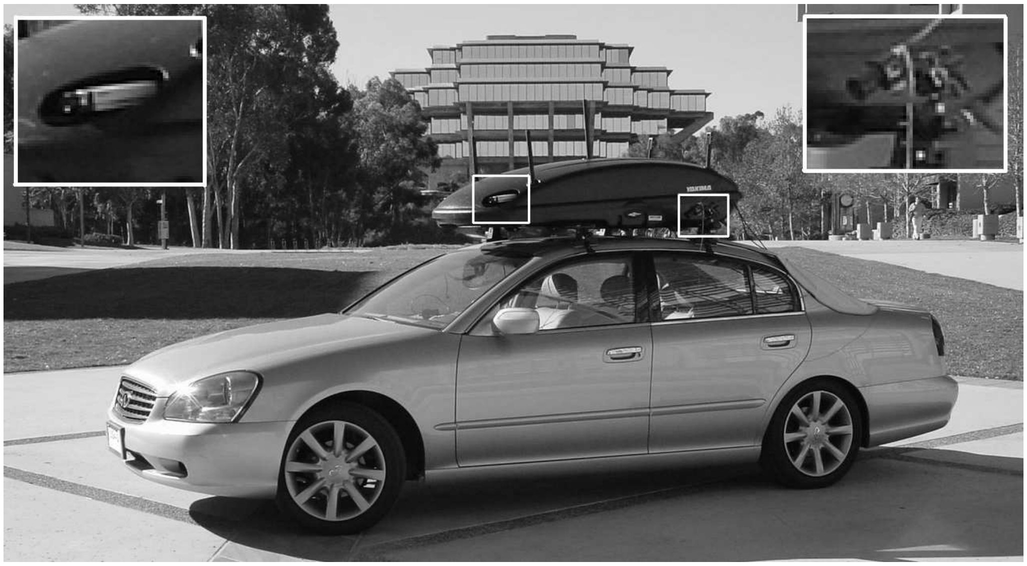


Figure VI.1: The LISA-Q intelligent vehicle test bed. Inset are close up views of the front camera (left inset) used for detection and tracking and side camera (right inset) used for generating ground truth.

The system described in this chapter, the Laboratory for Intelligent and Safe Automobiles - Q45 (LISA-Q) test bed, based on an Infiniti Q45 car model (Figure VI.1), is the first intelligent vehicle system that fully accomplishes these tasks. The LISA-Q test bed focuses on creating a system capable of collecting large amounts of data from a variety of modular sensing systems and processing that data in order to be fed back to the human occupant. Sensor systems include rectilinear cameras, wide field-of-view camera systems, GPS and navigation systems, internal automobile vehicle state sensors, as well as other sensor systems useful for study in intelligent vehicles. The system contains an array of computers that serve for data collection as well as real-time processing of information.

A.A.1 Related Research and Test Bed Vehicles

There have been many other vehicle test-beds that have been built for research into intelligent vehicles. These vehicles are mostly built for autonomous driving and or data collection with one specific purpose or algorithm in mind, or only capture a portion of the vehicles context, both interior and exterior. In our approach we intend to fill in the gaps between various data modalities and algorithms by creating a test bed that has complete sensor coverage of the surround, the interior, and the vehicle itself. Active research has been performed in using various sensors to create automated vehicles. One popular way to detect objects and navigate is through rangefinder sensors and GPS. Carnegie Mellon's NAVLAB [104] offers solutions for curb and people detection by using SIIC sensors and video for driver feedback. At the University of Minnesota, GPS is being used exclusively to guide busses on shoulder lanes of freeways. Other researchers have used vision-based systems to help develop research in this area. Nissan Motor Co. has developed a test bed [105] that uses both vision and embedded magnetic lane markers for lane keeping and collision avoidance (both from obstacles and lane departure). University of Parma's ARGO [106] used its stereo camera pair to do lane detection and localization of obstacles. The Australian National University Intelligent Vehicle Project used camera based lane tracking and obstacle detection. At the Institut fuer Systemdynamik und Flugmechanik, MarVEye, a camera array unit, is tested in VaMoR [21], a truck. It was used to autonomously drive the vehicle on rural roads. The University of Michigan Transportation Research Institute equipped a Honda accord with lane tracking and driver monitoring cameras in order to perform human factors research [107]. Other such systems and algorithms can be found in a recent survey by V. Kastrinaki et al. [19].

A.B The Laboratory for Intelligent Vehicles Infiniti Q45 Test Bed

The LISA-Q attempts to go beyond these systems by providing a system for complete synchronized context capture. This allows the LISA-Q test-bed to be used not only for surround analysis such as lane detection and obstacle detection, but also for monitoring driver behavior and state [7] as well as the vehicle state. By using the complete context of the vehicle surround, vehicle interior, and vehicle state, we can develop driver assistance systems that are more human centered.

A.B.1 Vehicle and Surround Information Capture

The LISA-Q information capture system is designed to obtain complete coverage of the vehicle surround, the vehicle interior, and the state of the vehicle for extended periods of time. This is achieved by a variety of sensor systems including rectilinear cameras, omnidirectional cameras, laser radar, microphones, and internal vehicle sensors. Figure VI.2 shows the LISA-Q information capture system

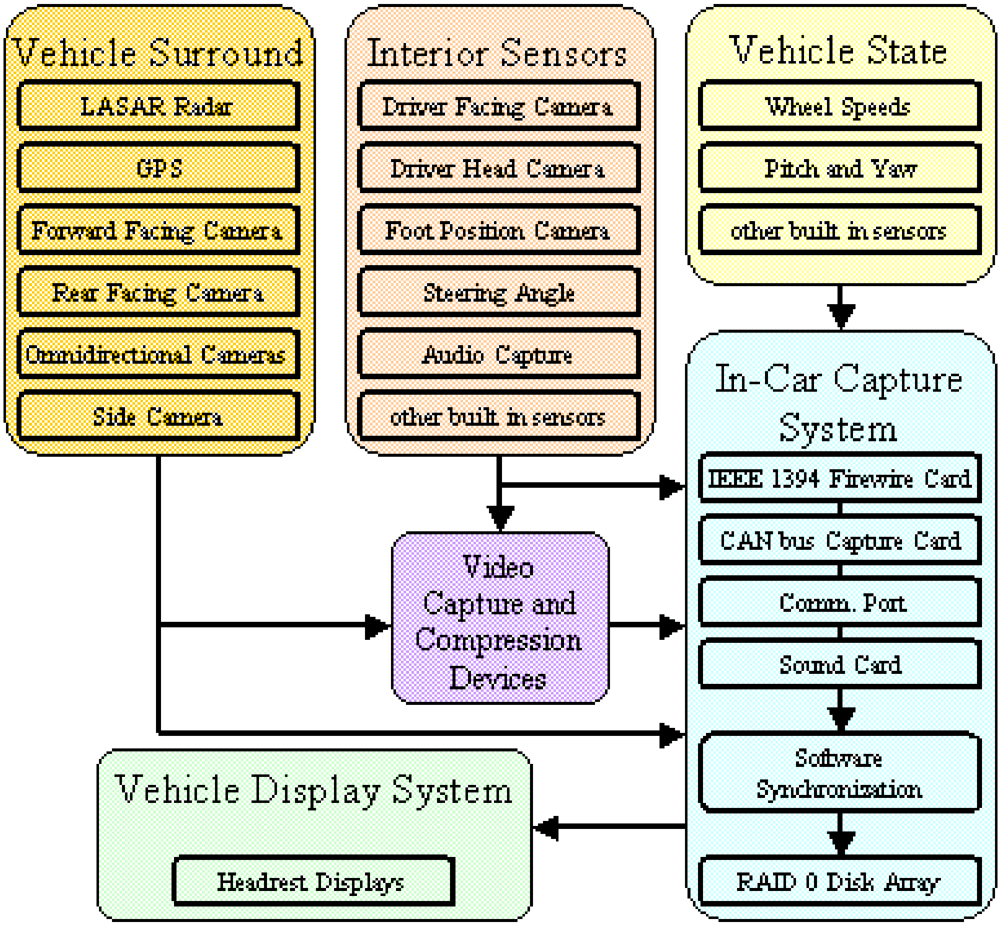


Figure VI.2: The LISA-Q Information Capture System

Vehicle Surround Capture

One of the requirements of an intelligent vehicle is to have information of the vehicles surround. For our test bed we have divided the vehicle surround into six sections, in front of the vehicle, to the rear of the vehicle, and both a front side and rear side for both the driver and passenger side of the vehicle. We can then choose sensors to get full coverage of the sections and assign importance to sections in building a surround map of the vehicle. For instance, the front section should include an area at least 80 meters in front of the vehicle, while the side sections need only extend 15-20 meters in order to capture lanes adjacent to the vehicle. Rear surround might contain less important data to the intelligent vehicle there by reducing the necessary resolution or coverage area.

To attain the external surround coverage, many types of sensors are used. The wide field of view camera (omnidirectional vision sensor) covers the short range (approximately 15-20 meters) in every segment. Front is further covered by a stereo pair of rectilinear cameras and a five beam laser range finder. Two rectilinear cameras cover the rear side left, rear, and rear side right. Figure VI.3 shows the layout and field of views of these sensors.

Vehicle Interior Capture

In order to provide driver/occupant analysis, the interior of the vehicle must also me captured (Figure VI.4). This is important for human behavior study in order to get information on the driver's state and decision making processes. The

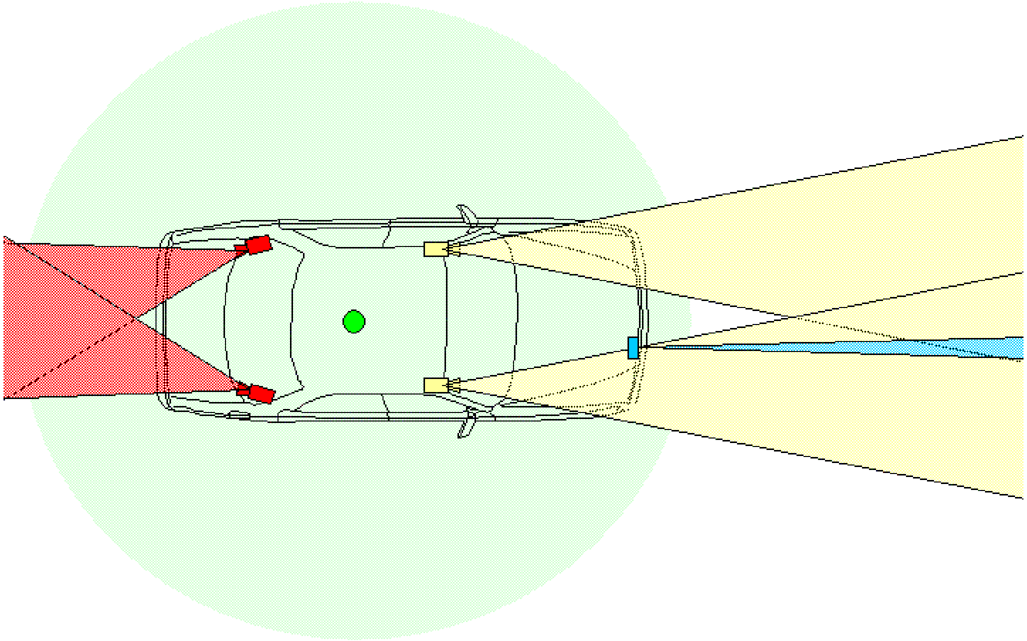


Figure VI.3: External Surround Sensors (Laser Radar, Front Rectilinear cameras, Omnidirectional Vision Sensors, Rear Rectilinear cameras)

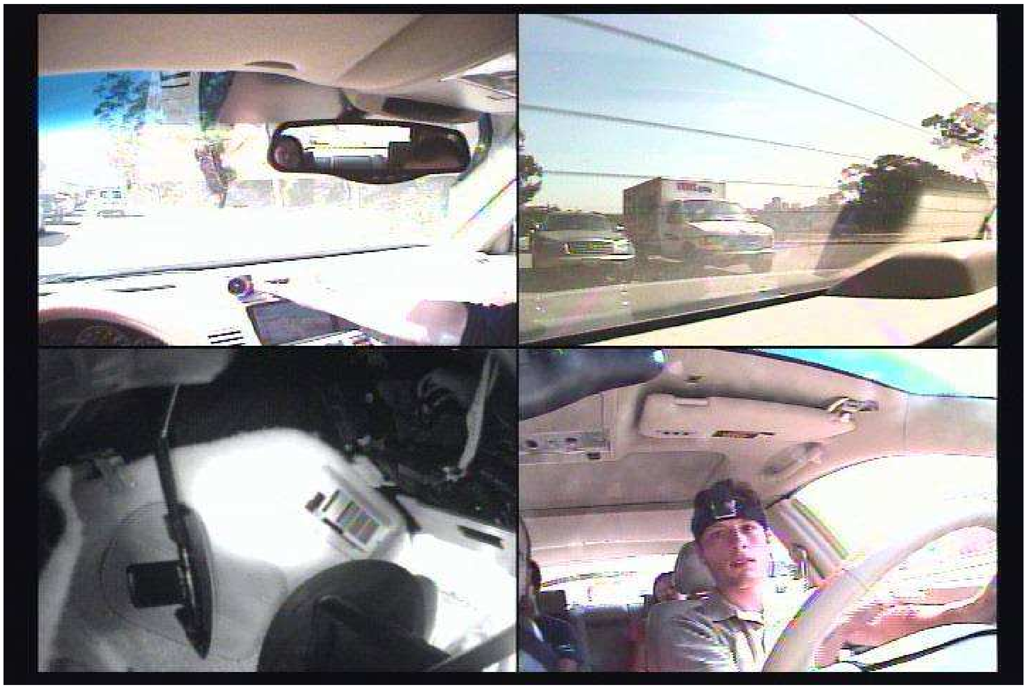


Figure VI.4: Top left: head cam capture; Top right: rear cam capture; Bottom left: Foot cam (note floating foot over brake pedal); Bottom right: Face Cam

interior context also plays an important part in many intelligent vehicle systems [48, 108]. In order for the interior capture system to be useful in these applications it must be unobtrusive so that it does not affect the behavior of the driver. Therefore cameras visible to the driver must be small or partial hidden from view. To attain internal coverage discretely, several video sensor types are used. To estimate current sight attention, a rectilinear headband camera (dubbed the subcam) is used. For capturing foot movement and hovering over each pedal, a near infrared sensitive black and white rectilinear camera with infrared illuminators is used. These video feeds are important for behavioral analysis studies because they provide information on the driver's actions. These combined with the vehicle surround and vehicle state help build a complete context for understanding driver behavior.

2.3 Vehicle State Capture The vehicle state can contain information valuable to determining human behavior and driver intent. Furthermore, vehicle state variable such as vehicle speed, steering angle, and so on can be useful for tracking ego-motion [109] and other algorithms for surround capture. Positional data obtained from GPS is also useful for determining the surround context based on maps. This is important for both the behavioral studies to get the drivers context as well as surround analysis algorithms to provide more information on the vehicles context (i.e. Highway, city streets, intersections, etc.) This information is available on the Infiniti Q45 CAN data bus. CAN capture is accomplished using LabView and a National Instruments NI-CAN capture card. The CAN bus data provides vehicle information including but not limited to speed, acceleration, braking, yaw rate, and distance to lead vehicle information from the Laser Radar system. The output of this system is time-stamped with the system time in milliseconds for synchronization with other sensor streams. GPS is captured using a Garmin GPS system connected to the data capture computer's serial port. The serial port is accessed and the data is parsed using software written explicitly for the LISA-Q data capture.

A.B.2 LISA-Q High-Bandwidth Sustained Data Capture System

In order to collect data for use in behavioral studies, the LISA-Q test-bed must be capable of collecting multiple video and data streams for periods of about 1 hour. The LISA-Q system addresses the problem of sustained high-bandwidth synchronized data capture by using real-time hardware compression and an integrated computer system for data capture (See Figure VI.2). In order to capture the complete vehicle context we require at least 4 full frame NTSC video streams as well as CAN bus data, GPS data, and audio data. Some of the video streams which are used for behavioral studies analysis but not algorithmic development can be combined into a quad stream.Using one quad stream allows us to obtain 7 video streams in the same bandwidth as 4 full frame video streams. Four full frame video streams in uncompressed RGB format take roughly 120 Megabytes/sec of bandwidth. Collecting data for a typical 1 hour run would then require 428 Gigabytes of data storage. This is too much bandwidth and capacity to make an uncompressed system feasible. Because of this, in the LISA-Q, we pipe the video streams through high quality DV converters to compress the data in real time while still preserving image quality. The result is a 25 megabit/sec bandwidth per video stream that will allow us to expand the capture system to 8 full frame video streams and beyond. This allows multiple data streams to be collected for long periods of time, thereby allowing extended off-line testing of algorithms and behavioral analysis studies.

A.C Real-World Versus Simulator Data Collection

Collecting data sets for driver behavior recognition in real-world environments is problematic for a variety of reasons:

- Interesting and often important events only occur occasionally, requiring large data sets.
- Events such as collisions are even less frequent and hazardous to the test subject and data collection system.
- The test subjects' knowledge of the test is prone to alter their behavior unless they are acclimated to the test vehicle over a large period of time.

However, certain events (commonly performed functions such as lane changing, braking, turning, etc.) can be extracted from these data sets and analyzed. A variety of intelligent vehicle test beds have been created to fulfill this data collection task [103, 110, 111]

Others have tried to avoid the above problems by using simulator studies. Simulator studies provide a convenient means of generating repeatable situations in which all vehicle and environmental parameters are accessible. Simulators require less time to capture data about very specific events, allowing for more subjects and tests to be conducted. While these studies solve many of the problems concerning rare events and dangerous situations by creating a tightly control environment, they also have some possibly undesirable aspects. Examples of these aspects are:

- Tightly controlled environment requires sensor noise and vehicle dynamics to be modeled and generated by the system. The model assumptions are not always similar to real-world conditions and systems.
- Driver behavior may vary due to the known artificiality and lack of consequences.
- Extremely complex simulators can be very difficult and expensive to build and maintain.
- Observing driver behavior using common techniques such as vision systems is quite different in a controlled environment than a real vehicle. Sensor and system design might not be optimal for real-world situations.

A large variety of simulator systems have been developed and used for testing; such as the National Advanced Driving Simulator [112] and the Virtual Test Track Experiment (VIRTTEX) at Ford Research Laboratory.

Bibliography

[1] B. N. Campbell, J. D. Smith, and W. G. Najm, "Examination of crash contributing factors using national crash databases," National Highway Traffic Safety Administration, Tech. Rep. DOT-VNTSC-NHTSA-02-07, October 2003.

[2] W. Najm, J. Koopmann, L. Boyle, and J. D. Smith, "Synthesis report: Examination of target vehicular crashes and potential its countermeasures," National Highway Traffic Safety Administration, Tech. Rep. DOT HS 808 263, June 1995.

[3] J. M^{c}Call, D. Wipf, M. M. Trivedi, and B. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," in *IEEE International Workshop on Machine Vision for Intelligent Vehicles*, June 2005.

[4] J. M^{c}Call and M. M. Trivedi, "Human behavior based predictive brake assistance," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Tokyo, Japan, June 2006.

[5] R. J. Apparies, T. C. Riniolo, and S. W. Porges, "A psychophysiological investigation of the effects of driving longer-combination vehicles," *Ergonomics*, vol. 41, no. 5, pp. 581–592, 1998.

[6] S. M. Belz, "An on-road inverstigation of self-rating of alertness and temporal seperation as indicators of driver fatigue in commercial motor vehicle operators," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2000.

[7] J. M^{c}Call and M. M. Trivedi, "Visual context capture and analysis for driver attention monitoring," in *Proceedings of IEEE Conference on Intelligent Transportation Systems*, October 2004, pp. 332–337.

[8] D. D. Salvucci, "Inferring driver intent: A case study in lane-change detection," in *Proceedings of the Human Factors Ergonomics Society 48th Annual Meeting*, 2004.

[9] N. Kuge, T. Yamamura, and O. Shimoyama, “A driver behavior recognition method based on a driver model framework,” Society of Automotive Engineers Publication, 1998.

[10] W. Kwon and S. Lee, “Performance evaluation of decision making strategies for an embedded lane departure warning system,” *Journal of Robotic Systems*, vol. 19, no. 10, pp. 499–509, 2002.

[11] F. Heimes and H.-H. Nagel, “Towards active machine-vision-based driver assistance for urban areas,” *International Journal of Computer Vision*, vol. 50, no. 1, pp. 5–34, 2002.

[12] W. Enkelmann, “Video-based driver assistance - from basic functions to applications,” *International Journal of Computer Vision*, vol. 45, no. 3, pp. 201–221, December 2001.

[13] The Johns Hopkins University Applied Physics Laboratory, “Survey of onboard technologies applicable to commercial vehicle operations,” US Department of Transportation, Tech. Rep., 1998.

[14] J. M^{c}Call, O. Achler, M. M. Trivedi, P. Fastrez, D. Forster, J. B. Haue, J. Hollan, and E. Boer, “A collaborative approach for human-centered driver assistance systems,” in *Proceedings of the 7th IEEE Conference on Intelligent Transportation Systems*, October 2004, pp. 663–667.

[15] H. Godthelp, P. Milgram, and G. J. Blaauw, “The development of a time-related measure to describe driving strategy,” *Human Factors*, vol. 26, pp. 257–268, 1984.

[16] K. Kluge, “Performance evaluation of vision-based lane sensing: some preliminary tools, metrics, and results,” in *Proceedings of IEEE Intelligent Transportation Systems Conference*, 1997, pp. 723–728.

[17] C. Taylor, J. Košecká, R. Blasi, and J. Malik, “A comparative study of vision-based lateral control strategies for autonomous highway driving,” *International Journal of Robotics Research*, vol. 18, no. 5, pp. 442–453, 1999.

[18] B. Ma, S. Lakshmanan, and A. O. Hero, “Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 5, pp. 135–147, September 2000.

[19] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, “A survey of video processing techniques for traffic applications,” *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[20] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial Vision in Road Vehicles," *Proceedings of the IEEE - Special issue on "Technology and Tools for Visual Perception"*, vol. 90, no. 7, pp. 1258–1271, July 2002.

[21] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-d road and relative egostate recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 199–213, February 1992.

[22] Y. Wang, E. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image and Vision Computing*, vol. 22, pp. 269–280, April 2004.

[23] J. Goldbeck, B. Huertgen, S. Ernst, and L. Kelch, "Lane following combining vision and DGPS," *Image and Vision Computing*, vol. 18, no. 5, pp. 425–433, April 2000.

[24] D. Pomerleau, "Neural network vision for robot driving," in *The Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed., 1995.

[25] 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, no. 1, pp. 62–81, Jan. 1998.

[26] K. Kluge and S. Lakshmanan, "A deformable template approach to lane detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 1995, pp. 54–59.

[27] S. Nedevschi, R..Schmidt, T. Graf, R. Danescu, D. Frentiu, T. Marita, F. Oniga, and C. Pocol, "3D lane detection system based on stereovision," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, October 2004, pp. 161–166.

[28] Y. Otsuka, S. Muramatsu, H. Takenaga, Y. Kobayashi, and T. Monj, "Multitype lane markers recognition using local edge direction," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, vol. 2, June 2002, pp. 604 – 609.

[29] C. Kreucher and S. Lakshmanan, "LANA: a lane extraction algorithm that uses frequency domain features," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 343–350, April 1999.

[30] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert: Special Issue on Intelligent System and their Applications*, vol. 11, no. 2, pp. 19–27, April 1996.

[31] Q. Li, N. Zheng, and H. Cheng, "Springrobot: a prototype autonomous vehicle and its algorithms for lane detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 300–308, December 2004.

[32] J. B. M^{c}Donald, “Detecting and tracking road markings using the hough transform,” in *Proceedings of the Irish Machine Vision and Image Processing Conference*, 2001.

[33] D.-J. Kang and M.-H. Jung, “Road lane segmentation using dynamic programming for active safety vehicles,” *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3177–3185, December 2003.

[34] N. Apostoloff and A. Zelinsky, “Robust vision based lane tracking using multiple cues and particle filtering,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, June 2003, pp. 558–563.

[35] B. Southall and C. J. Taylor, “Stochastic road shape estimation,” in *International Conference on Computer Vision*, 2001, pp. 205–212.

[36] S. Lee and W. Kwon, “Robust lane keeping from novel sensor fusion,” in *Proceedings IEEE International Conference on Robotics and Automation*, vol. 4, 2001, pp. 3704 – 3709.

[37] K. Astrom, R. Klein, and A. Lennartsson, “Bicycle dynamics and control: adapted bicycles for education and research,” *IEEE Control Systems Magazine*, vol. 25, no. 4, pp. 26–47, August 2005.

[38] F. Chausse, R. Aufrere, and R. Chapuis, “Recovering the 3d shape of a road by on-board monocular vision,” in *Proceedings of the 15th International Conference on Pattern Recognition*, 2000, pp. 325–328.

[39] K. Kluge and C. Thorpe, “The YARF system for vision-based road following,” *Mathematical and Computer Modelling*, vol. 22, no. 4-7, pp. 213–233, August 1995.

[40] C. Kreucher, S. Lakshmanan, and K. Kluge, “A driver warning system based on the LOIS lane detection algorithm,” in *Proceedings of the IEEE International Conference on Intelligent Vehicles*, October 1998, pp. 17–22.

[41] R. Risack, N. Mohler, and W. Enkelmann, “A video-based lane keeping assistant,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, October 2000, pp. 506–511.

[42] S. Baluja, “Evolution of an artificial neural network based autonomous land vehicle controller,” *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, no. 3, pp. 450 – 463, June 1996.

[43] J. W. Lee, “A machine vision system for lane-departure detection,” *Computer Vision and Image Understanding*, vol. 86, pp. 52–78, April 2002.

[44] J. W. Lee, C.-D. Kee, and U. K. Yi, "A new approach for lane departure identification," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2003, pp. 100–105.

[45] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[46] S. Szabo, K. Murphy, and M. Juberts, "The autonav / dot project: Baseline measurement system for evaluation of roadway departure warning systems," NISTIR 6300, National Institute of Standards and Technology, 1999.

[47] J.-Y. Bouguet. Camera calibration toolbox for matlab. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[48] K. Huang, M. M. Trivedi, and T. Gandhi, "Driver's view and vehicle surround estimation using omnidirectional video stream," in *Proceedings of IEEE Intelligent Vehicles Symposium*, Columbus, Ohio, September 2003, pp. 444–449.

[49] T. Gandhi and M. M. Trivedi, "Parametric ego-motion estimation for vehicle surround analysis using omni-directional camera," *Machine Vision and Applications*, vol. 16, no. 2, pp. 85–89, 2005.

[50] M. M. Trivedi, S. Y. Cheng, E. M. C. Childers, and S. J. Krotosky, "Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1698–1712, November 2004.

[51] Q. Ji and X. Yang, "Real time visual cues extraction for monitoring driver vigilance," *Lecture Notes in Computer Science: Computer Vision Systems*, vol. 2095, pp. 107–124, 2003.

[52] P. Ekman, "Expressions of emotion: An old controversy and new findings," *Philosophical Transactions of the Royal Society of London, Series B*, vol. 335(1273), pp. 63–69, 1992.

[53] P. Ekman and W. V. Freisen, "The facial action coding system: A technique for measurement of facial movement," San Francisco, CA, 1978.

[54] B. Fasel and J. Luettin, "Automatic Facial Expression Analysis: A Survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[55] M. Isard and A. Blake, "Condensation – conditional density propogation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[56] J. N. Bassili, "Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face." *Journal of Personality and Social Psychology*, vol. 37, pp. 2049–2059, 1979.

[57] I. A. Essa and A. Pentland, "Facial expression recognition using a dynamic model and motion energy," in *Proceedings of the International Conference on Computer Vision*, 1995, pp. 360–367.

[58] ——, "Coding, analysis, interpretation and recognition of facial expressions." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, July 1997.

[59] J.-J. Lien, T. Kanade, J. Cohn, and C. Li, "Detection, tracking, and classification of action units in facial expression," *Journal of Robotics and Autonomous Systems*, July 1999.

[60] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *International Journal of Computer Vision*, vol. 25, no. 1, pp. 23–48, 1997.

[61] H. Hong, H. Neven, and C. von der Malsburg, "Online facial expression recognition based on personal galleries," in *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 1998.

[62] C. Lisetti and D. Rumelhart, "Facial expression recognition using a neural network," in *Proceedings of the 11th International Flairs Conference*. AAAI Press, 1998.

[63] M. Dumas, "Emotional expression recognition using support vector machines," Machine Perception Lab, Univeristy of California, San Diego, Tech. Rep., 2001.

[64] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[65] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, 2002.

[66] I. Fasel, B. Fortenberry, and J. Movellan, "A generative framework for real time object detection and classification," *Computer Vision and Image Understanding*, vol. 98, pp. 182–210, 2005.

[67] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," *Lecture Notes in Computer Science: Proceedings of the 8th European Conference on Computer Vision*, vol. 3021, pp. 28–39, 2004.

[68] C. Su, Y. Zhuang, L. Huang, and F. Wu, “A two-step approach to mulitple facial feature tracking: Temporal particle filter and spatial belief propogation,” in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[69] Y. Rui and Y. Chen, “Better proposal distributions: Object tracking using unscented particle filter,” in *Proceedings of the IEEE Conference on Computer Vision and Patern Recognition*, 2001, pp. 786–793.

[70] R. Duda, P. Hart, and D. Stork, *Pattern Classification (Second ed.).* New York: John Wiley & Sons, Inc., 2000.

[71] P. Domingos and M. Pazzani, “Beyond independence: Conditions for the optimality of the simple bayesian classifier,” in *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp. 105–112.

[72] P. Phillips, H. Wechsler, J. Huang, and P. Rauss, “The FERET database and evaluation procedure for face-recognition algorithms,” *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[73] H. Schneiderman and T. Kanade, “Probabilistic modeling of local appearance and spatial relationships for object recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Patern Recognition*, 1998, pp. 45–51.

[74] F. L. Bookstein, “Principle warps: Thin-plate splines and the decomposition of deformations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, June 1989.

[75] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of online learning and an application to boosting,” in *Proceedings of the Second European Conference on Computational Learning Theory*, 1995.

[76] T. Kanade, J. F. Cohn, and Y. Tian, “Comprehensive database for facial expression analysis,” in *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, March 2000, pp. 46–53.

[77] Y.-L. Tian, T. Kanade, and J. F. Cohn, *Recognizing action units for facial expression analysis.* River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002, pp. 32–66.

[78] ——, “Recognizing action units for facial expression analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, February 2001.

[79] A. Broadhurst, S. Baker, and T. Kanade, “Monte carlo road safety reasoning,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, June 2005.

[80] M. Goodrich and E. Boer, “Designing human-centered automation: trade-offs in collision avoidance system design,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 33, no. 3, pp. 325–336, May 2003.

[81] N. Oliver and A. Pentland, “Driver behavior recognition and prediction in a smartcar,” in *Proceedings of SPIE Aerosense2000, ‘Enhanced and Synthetic Vision’*, 2000.

[82] I. Dagli, M. Brost, and G. Breuel, “Action recognition and prediction for driver assistance systems using dynamic belief networks,” *Lecture Notes in Artificial Intelligence, NODe Agent-Related Workshops*, vol. 2592, 2002.

[83] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[84] D. D. Salvucci, “Modeling driver behavior in a cognitive architecture,” *Human Factors*, In Press.

[85] ——, “Predicting the effects of cellular-phone dialing on driver performance,” *Cognitive Systems Research*, vol. 3, no. 1, pp. 95–102, 2002.

[86] ——, “A multitasking general executive for compound continuous tasks,” *Cognitive Science*, vol. 29, pp. 457–492, 2005.

[87] ——, “Lane-change detection using a computational driver model,” *Human Factors*, In Press.

[88] Q. Ji, P. Lan, and C. Looney, “A probabilistic framework for modeling and real-time monitoring human fatigue,” *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, (in press).

[89] S. Y. Cheng, S. Park, and M. M. Trivedi, “Multiperspective thermal IR and video arrays for 3D body tracking and driver activity analysis,” in *IEEE International Workshop on OTCBVS in Conjunction with IEEE International Conference on Computer Vision and Pattern Recognition*, June 2005.

[90] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, “Real-time system for monitoring driver vigilance,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, March 2006.

[91] P. Smith, M. Shah, and N. da Vitoria Lobo, “Determining driver visual attention with one camera,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 4, December 2003.

[92] C. Elkan, “Boosting and naive bayesian learning,” University of California, San Diego, Tech. Rep. CS97-557, September 1997.

[93] R. Srinivasan, “Effect of selected in-vehicle route guidance systems on driver reaction times,” *Human Factors*, vol. 39, no. 2, pp. 200–215, 1997.

[94] E. J. H. et al, “Feasibility study and conceptual design of a national advanced driving simulator,” The Gallup Organization, Tech. Rep. DOT-HS-809-566, April 2003.

[95] J. Levy, H. Pashler, and E. Boer, “Central interference in driving: Is there any stopping the psychological refractory period?” *Psychological Science*, vol. 17, no. 3, pp. 228–235, March 2006.

[96] “Automotive collision avoidance system field operational test report: Methodology and results,” University of Michigan Transportation Research Institute and General Motors, Tech. Rep. DOT-HS-809-900, August 2005.

[97] A. Vahidi and A. Eskandarian, “Research advances in intelligent collision avoidance and adaptive cruise control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, September 2003.

[98] T.-Y. Sun, S.-J. Tsai, J.-Y. Tseng, and Y.-C. Tseng, “The study on intelligent vehicle collision-avoidance system with vision perception and fuzzy decision making,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 112–117.

[99] R. Labayrade, C. Royere, and D. Aubert, “A collision mitigation system using laser scanner and stereovision fusiuon and its assessment,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, June 2005, pp. 441–446.

[100] J. Hillenbrand, K. Kroschel, and V. Schmid, “Situation assessment algorithm for a collision prevention assistant,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 459–465.

[101] F. Biral, M. D. Lio, and E. Bertolazzi, “Combining safety margins and user preferences into a driving criterion for optimal control-based computation of reference maneuvers for an ADAS of the next generation,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 36–41.

[102] M. Goodrich and E. Boer, “Designing human-centered automation: tradeoffs in collision avoidance system design,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 1, pp. 40–54, March 2000.

[103] J. M^{c}Call, O. Achler, and M. M. Trivedi, “Design of an instrumented vehicle testbed for developing human centered driver support system,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004, pp. 483 – 488.

[104] C. Thorpe, R. Aufrere, J. Carlson, D. Duggins, T. Fong, J. Gowdy, J. Kozar, R. MacLachlan, C. McCabe, C. Mertz, A. Suppe, C. Wang, and T. Yata, “Safe robot driving,” in *Proceedings of the International Conference on Machine Automation*, September 2002.

[105] S. Matsumoto, T. Yasuda, T. Kimura, T. Takahama, and H. Toyota, “Development of the nissan asv-2,” Nissan Motor Co., LTD, Tech. Rep. 382, 2001.

[106] M. Bertozzi, A. Broggi, and A. Fascioli, “Development and test of an intelligent vehicle prototype,” in *Proceedings of the 7th World Congress on Intelligent Transportation Systems*, November 2000.

[107] R. Sweet and P. Green, “Umtri’s instrumented car,” University of Michigan Trasportation Research Institute, Tech. Rep., 1993.

[108] S. Park and M. M. Trivedi, “Driver activity analysis for intelligent vehicles: Issues and developement framework,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, June 2005.

[109] T. Gandhi and M. M. Trivedi, “Dynamic panoramic surround map: Motivation and omni video based approach,” in *IEEE International Workshop on Machine Vision for Intelligent Vehicles in Conjunction with IEEE International Conference on Computer Vision and Pattern Recognition*, June 2005.

[110] S. Matsumoto, T. Yasuda, T. Kimura, T. Takahama, and H. Toyota, “Development of the nissan ASV-2,” Nissan Motor Co., LTD, Paper Number 382, 2001.

[111] V. L. Neale, T. A. Dingus, S. G. Klauer, J. Sudweeks, and M. Goodman, “An overview of the 100-car naturalistic driving study and findings,” in *Proceedings of the 19th International Technical Conference on Enhanced Safety of Vehicles*, Washington, DC, June 2005.

[112] D. Royal, “National survey of distracted and drowsy driving, volume 1: Findings report,” National Highway Traffic Safety Administration, Tech. Rep. DOT-HS-807-596, March 1990.