Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning

Joel C. McCall†, David Wipf‡, Mohan M. Trivedi†, Bhaskar Rao‡
†Computer Vision and Robotics Research Laboratory ‡Digital Signal Processing Laboratory
University of California, San Diego
{jmccall,dwipf,mtrivedi,brao}@ucsd.edu

Abstract

In this paper we demonstrate a driver intent inference system (DIIS) based on lane positional information, vehicle parameters, and driver head motion. We present robust computer vision methods for identifying and tracking freeway lanes and driver head motion. These algorithms are then applied and evaluated on real-world data collected in a modular intelligent vehicle test-bed. Analysis of the data for lane change intent is performed using a sparse Bayesian learning methodology. Finally, the system as a whole is evaluated using a novel metric and real-world data of vehicle parameters, lane position, and driver head motion.

1 Introduction

Intelligent vehicles and driver support systems have the potential to greatly enhance the safety of drivers and passengers by alerting the driver to dangerous situations. However, care must be taken to prevent the system from interfering with the driver in the middle of a corrective action, causing unnecessary distractions. Consequently, it is important for intelligent vehicles to not only recognize the situation, but also the driver's intended actions. In this paper, we will explore a vision system that estimates driver intentions in the specific area of lane changes, arguably one of the most important actions relevant to intelligent support systems.

The driver intent inference system (DIIS) we propose is composed of a few key components: the lane position tracking system, the driver head motion estimation module, the vehicle parameter collection system, and the lane change intent classifier. An overview of the system can be seen in figure 1. This paper provides results from the intermediate detection stages as well as overall classification results analyzed at various times preceding the lane change maneuver.

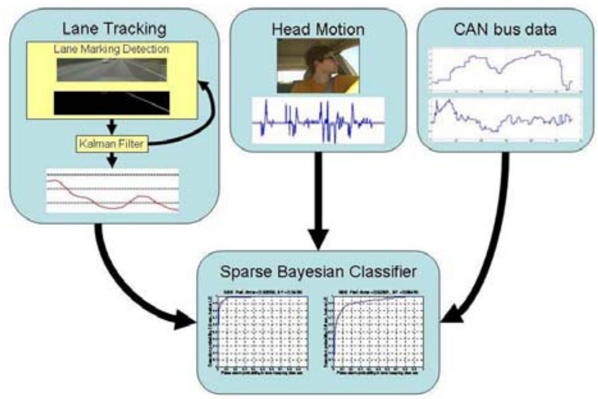

Figure 1. Lane change intent analysis system flow chart.

1.1 Relationship to Previous Work

Intelligent vehicle systems have been a topic of research for some time and envelope a wide area of research topics [3]. We will now address previous work related to driver intent inference and describe the improvements afforded by the proposed method. To begin, it is useful to make the following distinction:

Driver Intent Inference (Ideal):
Inferring if/when a driver is knowingly or intentionally about to execute a lane change.

Trajectory Forecasting (Practical):
Predicting if the vehicle trajectory is likely to cross the lane boundary in the near future (irrespective of driver awareness level).

Many current approaches, both data-driven [4] and model-based [5], essentially perform trajectory forecasting, using the results as a proxy for driver intent. More specifically, if the algorithm predicts that in the next few seconds

1063-6919/05 $20.00 © 2005 IEEE

the car will likely cross the lane boundary, then we must presume he/she intends a lane change. Kuge et. al. [4] developed Hidden Markov Models (HMMs) using observations of vehicle parameters and lane positions to model trajectories. In contrast, Salvucci et. al. [5] employed a template matching technique inspired by a cognitive architectures to model vehicle trajectories. While both techniques are certainly much easier than predicting driver intent, differences between trajectory forecasts and what a driver is actually intending can be extremely problematic as we will describe later.

To alleviate this problem, we have incorporated head motion into the modeling process to provide the DIIS with specific driver state information useful in addressing what is potentially known and what is not. This is extremely useful in differentiating deviant trajectories intended to change lanes and those attributable to capricious drift while lane keeping.

Next, we address the platform upon which data is collected and algorithms are developed. Many previous works, such as those mentioned above, are strictly simulator-based methodologies. Simulator studies, however, do not account for conditions encountered in real-world situations. Lighting changes, shadows, vibrations, and occlusions all contribute to added noise not present in a simulator. To address these issues, we have created an elaborate testbed using a fully functional vehicle.

Oliver et. al [6] have also created a system for studying driver behavior inside a real vehicle. Their work included head pose data as well as vehicle and lane parameters. Classification of various driving tasks (e.g., right turn, left turn, passing, etc.) was performed using graphical models including hidden Markov models (HMMs) and Coupled HMMs. The results shown in this research were interesting but failed to provide benchmark results for the null event, i.e., lane keeping. Rather, classification was only performed on small segments of data where an event was known to occur. Consequently, false alarm rates were not used to guide model development, nor included in the recorded results, making it impossible to evaluate the efficacy of this system. In general, false alarm rates are absolutely essential for human-machine interface modules because of the amount of annoyance generated by an incorrect analysis. Our system goes beyond the previous research by using receiver operator characteristic (ROC) curves as the cornerstone of a specially designed evaluation metric.

At a lower level of the system, lane tracking has also been an active research area in computer vision. Algorithms have been developed using techniques like Hough transforms [7], neural networks [8], and stochastic methods [9]. These techniques do not account for different types of lane markings (such as circular reflectors) or complex shadowing (such as tree shadows). Gehrig et. al. [10] propose a method for detecting multiple types of lane markings by combining different types of lane detectors. Our method of lane detection unifies the detection of multiple types of lane marking and provides robustness to shadowing.

Finally, given a large, diverse set of potential features available for constructing a DIIS, it would extremely useful to have a principled means of mapping candidate features into intention probabilities. The recently developed field of sparse Bayesian learning [12] provides a useful tool in this respect. Our method relies heavily on this approach in sifting through candidate features and discarding those that are irrelevant.

2 Vision Systems for Driver Intent

Data needed to infer driver intent comes from a variety of sources. In our system we have added sensors to a Infiniti Q45. A data collection computer located in the trunk of the car captures and synchronizes up to eight full frame video streams, the vehicle's CAN bus data, and GPS information. More information on the hardware setup of the intelligent vehicle test-bed used in this research is described in [1].

In order to create a feature vector for the classification stage, a time series of data describing the vehicle's surround, the driver, and the vehicle's internal state must be created. Lane position, heading, and curvature information is extracted from a forward looking rectilinear camera and is described in section 2.1. Head motion is extracted from a rectilinear camera viewing the driver as described in section 2.2. Vehicle parameters such as vehicle speed, steering angle, pedal positions, yaw rate, and lateral acceleration are obtained via the vehicles CAN bus.

2.1 Lane Tracking

2.1.1 Lane Detection Using Steerable Filters

The lane tracking method used in this system uses steerable filters [11] to enable the detection of multiple types of lane markings. Specifically, steerable filters are well suited to finding both lines and circular lane markings because filter responses of any orientation can be calculated using a small number of seperable filters. This paper extends the work shown in [2] by adding improved curvature detection and a more extensive evaluation.

The filterbank used in this system is composed of the three filters. These filters correspond to second derivatives of Gaussians and can be expressed by equations 1, 2, and 3.

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

$$G_{xy}(x,y) = \frac{\partial^2}{dxdy} e^{\frac{-(x^2+y^2)}{\sigma^2}} \quad (2)$$

$$G_{yy}(x,y) = \frac{\partial^2}{dy^2} e^{\frac{-(x^2+y^2)}{\sigma^2}} \quad (3)$$

It has been shown that the response of any rotation of the G_{xx} filter can be computed using equation 4 [11].

$$G2^{\theta}(x,y) = G_{xx}\cos^2\theta + G_{yy}\sin^2\theta + G_{xy}\cos\theta\sin\theta \quad (4)$$

Taking the derivative of (4), 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 5.

$$\theta_{min,max} = \arctan\left(\frac{G_{xx} - G_{yy} \pm A}{2G_{xy}}\right) \quad (5)$$

where,

$$A = \sqrt{G_{xx}^2 - 2G_{xx}G_{yy} + G_{yy}^2 + 4G_{xy}} \quad (6)$$

Using the formulas 4 and 5, we can find the values and angles of the minimum and maximum responses. This is useful for detecting circular reflectors because the minimum and maximum responses are relatively large. For detecting lines, the response of the filterbank in the direction of the lane should be near the maximum, and the minimum response should be low. Therefore, the same filterbank can be used to detect both circular reflectors and lines. Figure 2 shows a typical highway scene and the responses for both circular reflectors and lines.

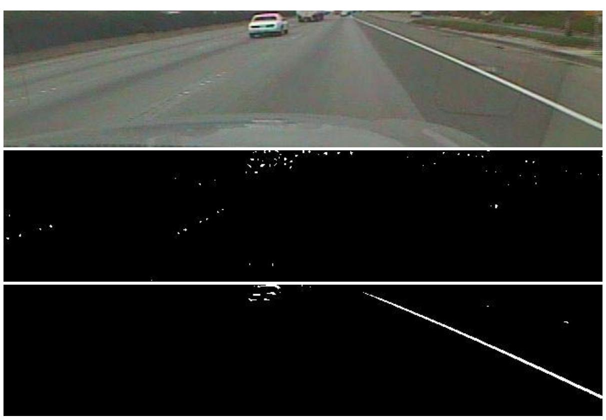

Figure 2. A typical highway scene in California (top), response for circular reflectors (middle), and response for solid line markings (bottom).

2.1.2 Road Curvature Detection

Circular reflectors become more difficult to detect when they are far away from the car. This makes road curvature detection based on circular reflectors difficult. Furthermore, line markings are not always present to allow for curvature detection based on lane markings alone. In order to compensate for this, we use and alternate method for detecting road curvature. Using an adaptive template of the lane, we find the least square error parabolic fit to the lane ahead. Template matching is performed up to 100 meters in front of the vehicle. Results of this detection are presented in figure 3.

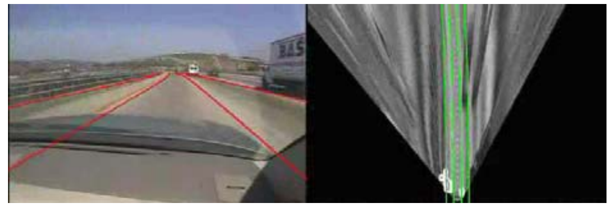

Figure 3. Lane tracking results overlayed onto video, aerial photograph of the road, and an inverse perspective warping of video.

2.1.3 Outlier Removal

In order to make the lane tracking more robust, outlier removal is performed based on a prior knowledge of the lanes. Specifically, detected markings are culled based on their proximity to the estimated lane position, the statistics of the detected markings, and the motion of the markings. In general, lanes markings are approximately straight near the vehicle. Therefore, the covariance of the detected lane marking positions should have one large eigenvalue and one small eigenvalue. Measurements for which the ratio of the eigenvectors fall below a threshold are discarded.

For our system, the videos frames being captured are interleaved. Because half of the frame is exposed one sixtieth of a second after the other half, we can detect the motion of detected lane markings by separating the even and odd lines. Estimating the motion within the frame creates robustness to dropped frames, which is important at freeway speeds. We then eliminate detected markings that are not moving with the road plane.

2.1.4 Lane Tracking

Lane tracking is performed using a Kalman filter. The Kalman state variables are updated using the measurements from the lane detection along with measurements of steering angle and wheel velocity provided by the vehicle's CAN bus. The state vector is composed of the lane position, lane heading, lane curvature, derivative of lane position, derivative of lane heading, lane width, vehicle speed, steering angle, and vehicle acceleration.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

2.1.5 Experimental results for lane position

The lane detection and tracking systems were evaluated on roads containing both circular markings and line markings. The mean absolute positional error was found to be 10.16 cm and the standard deviation of the error 13.17cm. Figure 4 shows the positional information from the lane tracker compared to the ground truth data used in testing. Ground truth was obtained by hand marking the lanes from a separate camera on the side of the vehicle.

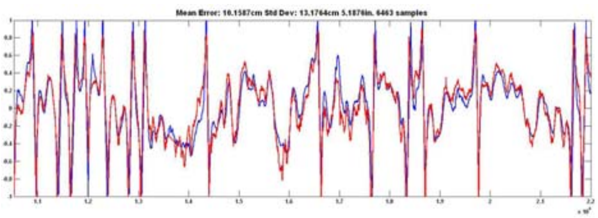

Figure 4. Lane tracking results. Red: Ground Truth. Blue: Detected Lane Position

2.2 Head Motion Estimation

Head pose is estimated using block matching. The drivers head in the previous frame is matched to the current frame, giving a disparity which can be considered inter-frame motion. Figure 5 shows the head motion of a driver plotted along with the lane position of the vehicle. It can be seen that the driver's head motion increases before and during lane change maneuvers.

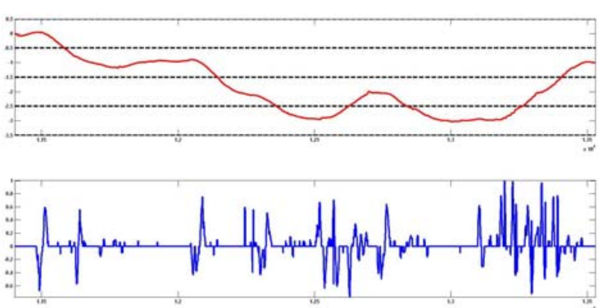

Figure 5. Lane position and head motion detection results. (*Top*) Black lines denote lane boundaries while the red line is the vehicle trajectory. (*Bottom*) Blue line represents side-to-side head motion.

3 Driver Intent Inference: System Description

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 [4].

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.

3.1 Sparse Bayesian Learning

Sparse Bayesian learning (SBL) is a powerful approach recently introduced into the machine learning literature for solving regression and classification problems [12]. 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

$$y(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) \tag{7}$$

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

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 IEEE

$\phi_i(\cdot) = K(\cdot, \boldsymbol{x}_i)$, 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. A more comprehensive description of SBL can be found in [12]. 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

$$\begin{aligned}\boldsymbol{x}(t) \quad = \quad & [\,\text{LateralPos}(t), \ldots, \text{LateralPos}(t-N+1); \\ & \text{Heading}(t), \ldots, \text{Heading}(t-N+1); \\ & \text{etc. }], \end{aligned} \tag{8}$$

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. 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. Where successful, this construction obviates the need for HMMs since we are essentially creating a large, fixed-length feature space and entrusting the SBL with sorting out relevant subspaces amenable to classification.

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.

3.2 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:

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

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. Moreover, it naturally solves the problems raised above and, as discussed next, it addresses specific DIIS ideological concerns.

3.3 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.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 IEEE

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.

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 6 and 7 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.

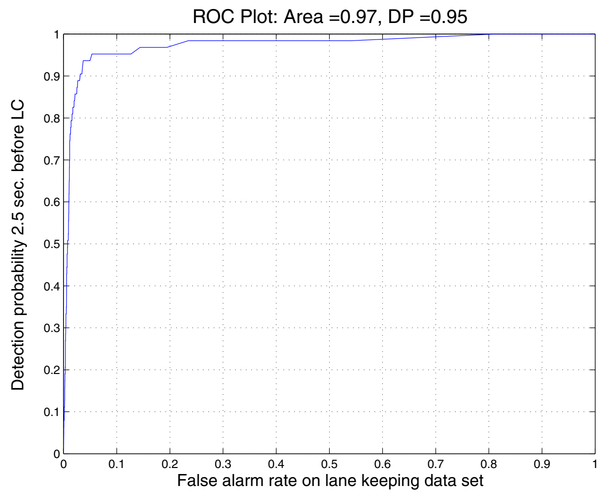

Figure 6. ROC curve obtained from 2.5 seconds before a lane change.

In contrast, when we exclude driver state information, results are significantly worse as expected. This is displayed in Figures 8 and 9. 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.

5 Summary and Conclusions

Accurately inferring driver intentions represents an important component of intelligent vehicle support systems. When errors do occur with such an inference, we have three potential culprits to contend with:

1. The lane tracker/ surround map failed,
2. The DIIS algorithm failed, or
3. The observable data was consistent with multiple driver intents.

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

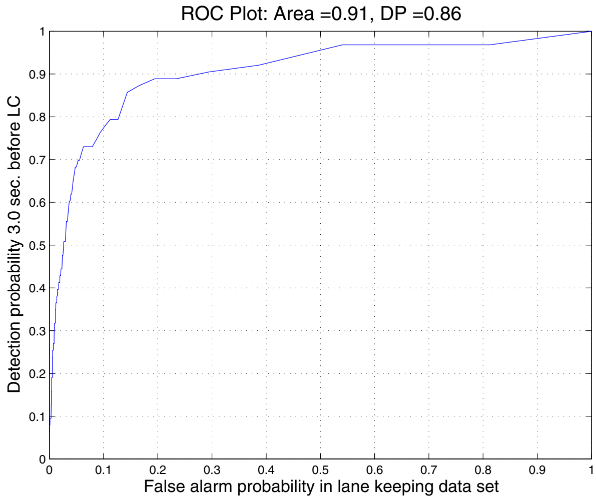

Figure 7. ROC curve obtained from 3.0 seconds before a lane change.

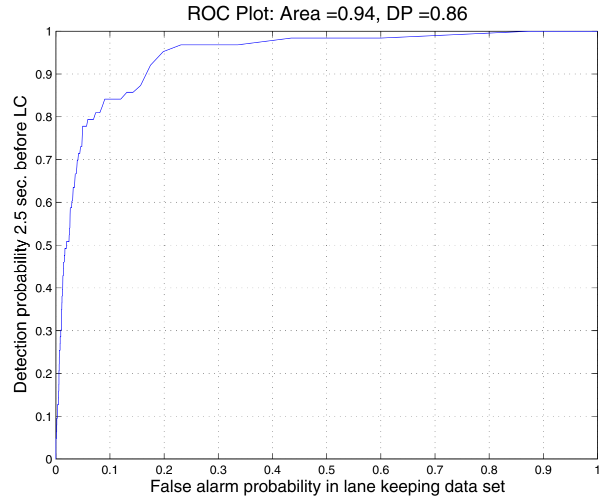

Figure 8. Using no driver state information (i.e., pure trajectory forecasting), ROC curve obtained from 2.5 seconds before a lane change.

In this paper, we have tried to address each of these issues. First, by developing a more general lane tracker capable of robustly handling various types of lane markings, we create a more accurate surround map. Secondly, by incorporating a state-of-the-art SBL classifier with well-motivated evaluation metrics, we reduce the likelihood of DIIS algorithmic failures. Thirdly, by incorporating driver state information and moving away from simple trajectory forecasting, we increase the likelihood that the observable data is consistent with one and only one driver intent. Finally, by utilizing an actual vehicle (as opposed to a simulator) for all data collection and model development, we move one step closer to a useable driver intent inference system.

Acknowledgments

The authors would like to thank UC Discovery Grants and the Digital Media Innovations Program, Nissan Motor Co. LTD., and their colleagues at the Computer Vision and Robotics Research Laboratory and Digital Signal Processing Laboratory.

References

[1] J. McCall, O. Achler, and M. M. Trivedi, "The LISA-Q human-centered intelligent vehicle test bed," in *Proc. IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004.

[2] J. McCall and M. M. Trivedi, "An integrated, robust approach to lane marking detection and lane tracking," in *Proc. IEEE Intelligent Vehicles Symposium*, Parma, Italy, June 2004.

[3] 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.

[4] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu "A driver behavior recognition method based on a driver model framework," *Proceedings of the Society of Automotive Engineers World Congress*, 2000.

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

[6] N. Oliver and A. Pentland, "Graphical models for driver behavior recognition in a smartcar," in *Proceedings of IEEE Intl. Conference on Intelligent Vehicles*, October 2000.

[7] J. B. McDonald, "Detecting and tracking road markings using the Hough transform," in *Proc. of the Irish Machine Vision and Image Processing Conference*, 2001.

[8] D. Pomerleau, "RALPH: Rapidly adapting lateral position handler," in *Proc. IEEE Intelligent Vehicles Symposium*, 1995, pp. 54–59.

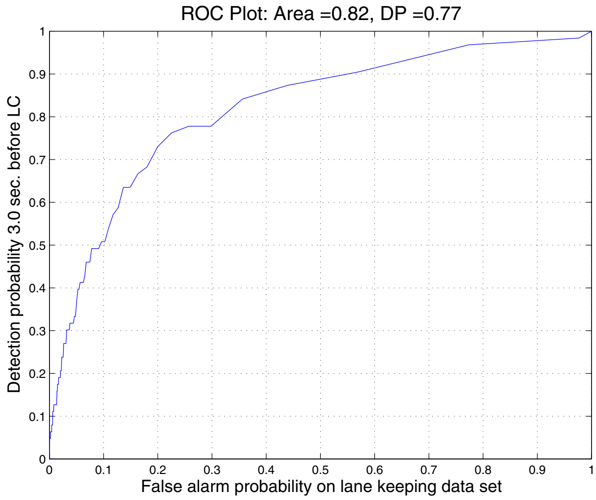

Figure 9. Using no driver state information (i.e., pure trajectory forecasting), ROC curve obtained from 3.0 seconds before a lane change.

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

[10] S. Gehrig, A. Gern, S. Heinrich, and B. Woltermann, "Lane recognition on poorly structured roads - the bots dot problem in California," *Proc. The IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, pp. 67–71.

[11] 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.

[12] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**