Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning

Joel C. McCall, David P. Wipf, Mohan M. Trivedi, and Bhaskar D. Rao, *Fellow, IEEE*

***Abstract*—In this paper, we demonstrate a driver intent inference system that is 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 that are 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.**

***Index Terms*—Computer vision, driver assistance systems, driver intent inference, intelligent vehicles, sparse Bayesian learning (SBL).**

I. 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's action, causing unnecessary distractions or even working against the driver. Consequently, it is important for intelligent vehicles to not only recognize the situation but the driver's intended actions as well. In this paper, we will explore a vision system that estimates driver intentions in the specific area of lane changes, which is 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 is shown in Fig. 1. This paper provides results from the intermediate detection stages as well as overall classification results that are analyzed at various times preceding the lane change maneuver.

A. Relationship to Previous Work

Intelligent vehicle systems have been a topic of research for some time and envelope a wide area of research topics [1].

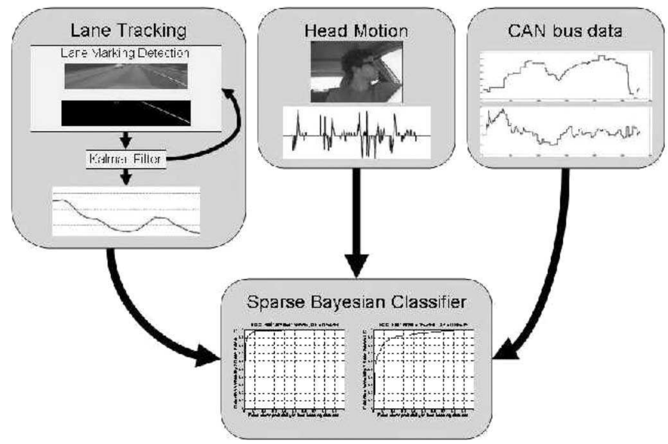


Fig. 1. Lane change intent analysis system flow chart.

We will now address previous work that is related to driver intent inference and describe the improvements afforded by the proposed method. As this paper focuses on driver lane change intent inference, we will focus our discussion of previous works to this application. To begin with, it is useful to make the following distinction between lane change intent inference and trajectory forecasting:

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 [2] and model-based [3], 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 the car will likely cross the lane boundary, then we must presume he/she intends a lane change. Kuge *et al.* [2] developed hidden Markov models (HMMs) using observations of vehicle parameters and lane positions to model trajectories. In contrast, Salvucci and Liu [3] employed cognitive architectures to model strategic "behavioral 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 that is useful in addressing what is potentially known and what is not. This is extremely

Manuscript received September 23, 2005; revised April 12, 2006. The Associate Editor for this paper was U. Nunes.

J. C. McCall is with the Mobile Devices Division, Microsoft Corporation, Redmond, WA 98052 USA (e-mail: jmccall@microsoft.com).

D. P. Wipf and B. D. Rao are with the Digital Signal Processing Laboratory, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: dwipf@ucsd.edu; brao@ucsd.edu).

M. M. Trivedi is with the Laboratory for Intelligent and Safe Automobiles, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: mtrivedi@ucsd.edu).

Digital Object Identifier 10.1109/TITS.2007.902640

TABLE I
RELATIONSHIPS BETWEEN THE PREVIOUS RESEARCH ACTIVITIES DESCRIBED IN THIS PAPER

Researchers	Inputs	Classifier	Context	Evaluation	Data Collection	Comments
Risack '00 [6], Lee '03 [7], Enkelmann '01 [8]	Lane and Vehicle Data	Trajectory Prediction	Lane Departure Warning	Lane Departure Detection Rate	Real-World Data	No Driver Analysis
Kuge '98 [2]	Lane Data Vehicle Data Surround Map	HMM	Behavior Recognition	Detection Rate	Simulator Data	Does not include sensor error Artificial driving scenario
Oliver '00 [4]	Lane Data Vehicle Data Head Pose Surround Map	HMM Coupled HMM	Behavior Recognition	Detection Rate	Real-World Data	Forced Classification No False Alarms
Salvucci '04 [9]	Lane Data Vehicle Data Surround Map	Cognitive Model	Behavior Recognition/mimicking	Model Performance vs. Human	Simulator	Focus on generating maneuvers Simulated Data
This Article	Lane Data Head Movement Vehicle Data	Sparse Bayesian Learning	Driver Intent Analysis	ROC curves with and without head movement data	Real-World Data	Intent rather than behavior classification

useful in differentiating deviant trajectories that are intended to change lanes and those attributable to capricious drift while lane keeping.

Next, we address the platform upon which data are 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 that are 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 test bed using a fully functional vehicle.

Oliver and Pentland [4] 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 HMMs and coupled HMMs. The results shown in this paper 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 that is 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. The differences and innovations from previous works and our system are accumulated in Table I. This paper also expands upon details and introduces some improvements to our previous work in this area [5].

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 [10], neural networks [11], and stochastic methods [12]. 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.* [13] 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 be extremely useful to have a principled means of mapping candidate features into intention probabilities. The recently developed field of sparse Bayesian learning (SBL) [14] 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.

II. VISION SYSTEMS FOR DRIVER INTENT

Data needed to infer driver intent come from a variety of sources. In our system, we have added sensors to an Infiniti Q45. A data-collection computer that is 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 paper is described in [15].

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 are extracted from a forward-looking rectilinear camera and are described in Section II-A. Head motion is extracted from a rectilinear camera viewing the driver, as described in Section II-B. Vehicle parameters such as vehicle speed, steering angle, pedal positions, yaw rate, and lateral acceleration are obtained via the vehicle's CAN bus.

A. Lane Tracking

1) Lane Detection Using Steerable Filters: The lane-tracking method used in this system uses steerable filters [16] 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 separable filters. A more detailed exploration of lane tracking using steerable filters, including a detailed evaluation, can be found in [17].

The filterbank used in this system is composed of three filters. These filters correspond to second derivatives of Gaussians. It has been shown that the response of any rotation of the second derivative Gaussian filters can be computed using (1), which is shown below [16]. In this equation, G_{xx}, G_{yy}, and G_{xy} represent the second derivatives of Gaussians in the x-direction, y-direction, and first derivatives in both the x- and y-directions, respectively.

$$G2^{\theta}(x,y) = G_{xx}\cos^2\theta + G_{yy}\sin^2\theta - 2G_{xy}\cos\theta\sin\theta. \quad (1)$$

Taking the derivative of (1), setting it equal to zero, and solving for θ, we can find the values which correspond to the minimum and maximum responses. These responses can be computed by the formulas given as

$$G2^{\theta_{\mathrm{min\,/\,max}}} = G_{yy} - \frac{2G_{xy}^2}{G_{xx} - G_{yy} \pm A} \quad (2)$$

where

$$A = \sqrt{G_{xx}^2 - 2G_{xx}G_{yy} + G_{yy}^2 + 4G_{xy}^2}. \quad (3)$$

By using (1) and (2), we can find the values of the minimum and maximum responses. This is useful in detecting small 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. Fig. 2(a) shows a typical highway scene and the responses for both circular reflectors and lines. The technique can be applied to images that are transformed into world coordinates using the inverse perspective equations. Example images of the inverse perspective warped image as well as results for circular and lane marking detection are shown in Figs. 2 and 3. The advantages to this technique include the ability to use a single filter size for a wider 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 further explore the utility of the inverse perspective image and transformation in the following sections.

2) Road Modeling and Curvature Detection: In our system, we are using a parabolic model for the road in which the center of each lane in the road is fit to a parabola where the lane markings are each a half lane width distance aware. This model is shown in Fig. 4. For straight roads, the curvature parameter C is zero.

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 that is based on lane markings alone. In order to compensate for

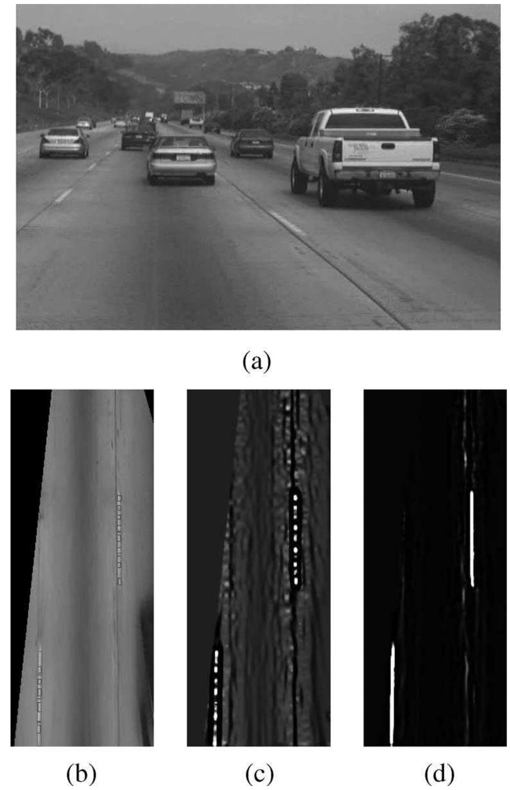


Fig. 2. (a) Image from a road scene containing solid line markings with embedded circular reflectors as well as (b) the image transformed using the inverse perspective transformation, which is subsequently filtered for (c) circular reflectors and (d) solid line markings.

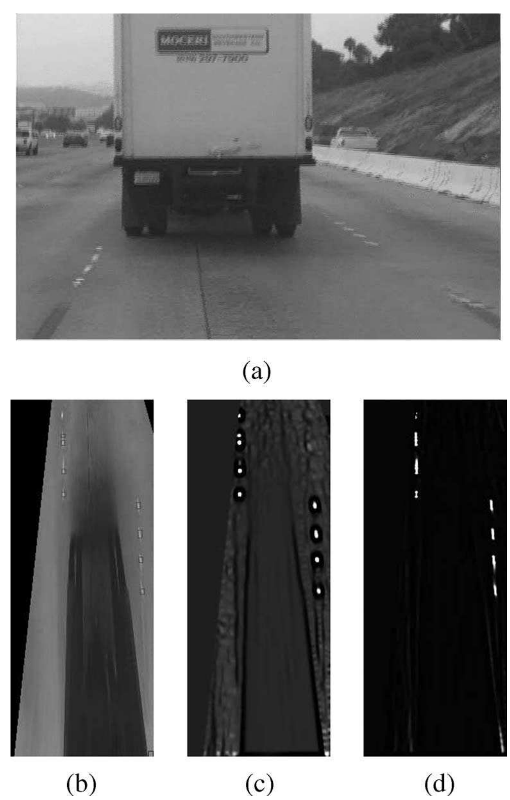


Fig. 3. (a) Image from a road scene containing circular reflector markings as well as (b) the image transformed using the inverse perspective transformation, which is subsequently filtered for (c) circular reflectors and (d) solid line markings.

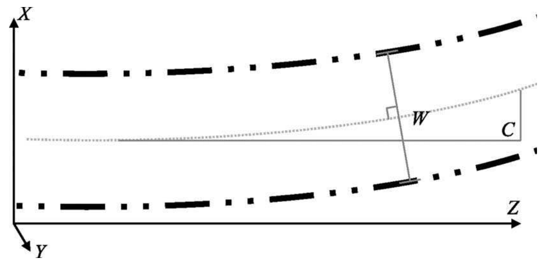


Fig. 4. Parabolic model for road curvature, where $C = d^2x/dz^2$ is the road curvature, and W is the road width.

Fig. 5. Lane-tracking results overlayed onto video. Clockwise from the upper left quadrant, the results are shown for daytime, dawn, afternoon, and nighttime driving scenes.

this, we use an alternate method for detecting road curvature. Using an adaptive template of the lane, we find the least square error fit to the lane ahead. Template matching is performed up to 100 m in front of the vehicle. Results of this detection are presented in Fig. 5.

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, lane 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 video 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 assume a planar homography and eliminate detected markings that are not moving with the road plane.

4) Vehicle Modeling and 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

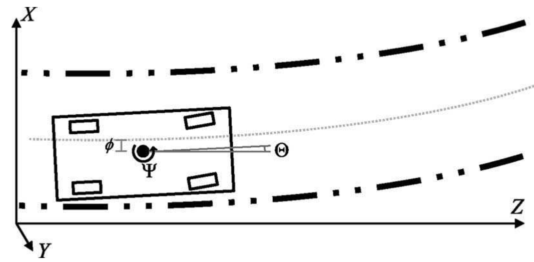


Fig. 6. Vehicle model consisting of position in the lane (ϕ), angle relative to the lane (Θ), and yaw rate (Ψ).

speed, steering angle, and vehicle acceleration (Fig. 6). The system and measurement equations are described as follows:

$$x_{k+1|k} = Ax_{k|k} \tag{4}$$

$$y_k = Mx_k \tag{5}$$

where

$$x = [\phi, \dot{\phi} \approx \Theta, \dot{\Theta}, \Phi, W]^{\mathrm{T}} \tag{6}$$

$$A = \begin{bmatrix} 1 & v\Delta t & \frac{(v\Delta t)^2}{2} & \frac{(v\Delta t)^3}{6l} & 0 \\ 0 & 1 & v\Delta t & \frac{(v\Delta t)^2}{2l} & 0 \\ 0 & 0 & 1 & \frac{v\Delta t}{l} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.5 \\ 1 & 0 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{8}$$

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 was 13.17 cm. Fig. 7 shows the positional information from the lane tracker compared to the ground truth data used in testing. Fig. 8 shows the absolute error in meters compared to the ground truth data. Ground truth was obtained by hand-marking the lanes from a separate camera on the side of the vehicle. Using a separate camera viewing the road straight down near the front of the vehicle provides a better view of the lane markings close to the car and is more robust to vehicle pitch. This camera was not used in automated lane marking detection because of the lack of look-ahead for curvature detection.

B. Head Motion Estimation

Head motion estimation is performed in two steps. First, the location of the driver's head in the image is determined. Second, the interframe motion is determined. In this way, we can construct a motion vector that allows us to identify head movements by the driver. These types of movements are useful in determining when the driver is inspecting mirrors or the surrounding when initiating a lane change.

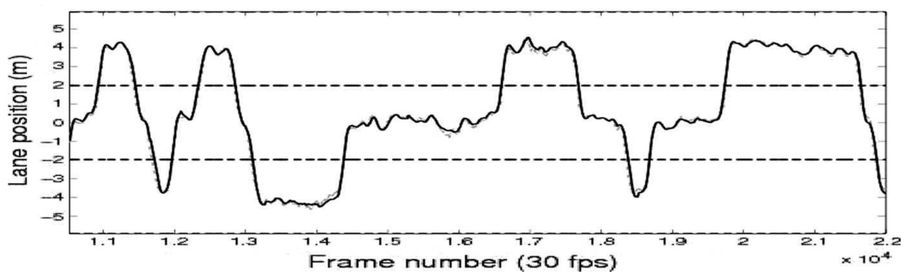


Fig. 7. Detected lane position in meters (solid black) superimposed on ground truth (dashed gray) plotted versus frame number with dashed lines marking the position of lane boundaries for an 11 000 frame (slightly over 6 min) sequence.

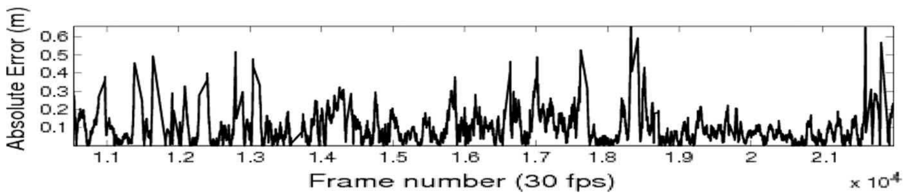


Fig. 8. Absolute error in lane position in meters for the sequence shown in Fig. 7.

Fig. 9. Head detection results for various lighting conditions.

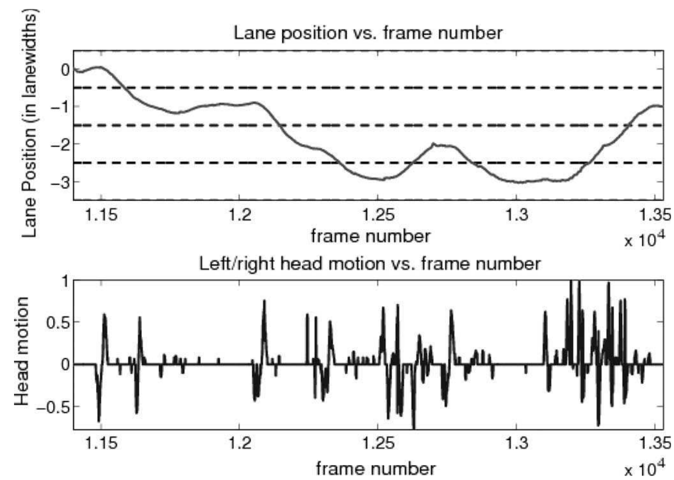


Fig. 10. Lane position and head motion detection results. The dashed lines represent lane boundaries, and the solid line represents (top) lane position and (bottom) side-to-side head movement.

Head detection in a vehicle environment required robustness to illumination changes, partial occlusions, and shadowing. In order to create a robust detector, we use a weighted combination of face area detectors similar to that presented by Schneiderman and Kanade [18]. More specifically, by combining the results of an eye region detector, a nose region detector, and a mouth region detector, we can more accurately detect faces under conditions such as nonuniform lighting and occlusion. Fig. 9 shows the results of the combined detector under various lighting conditions and occlusion.

Head pose is estimated using block matching, which is similar to that implored by Jain and Jain [19]. The driver's head in the previous frame is matched to the current frame, giving a disparity which can be considered interframe motion. Fig. 10 shows the head motion of a driver, which is 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.

III. Driver Intent Inference: System Description

At its core, driver intent inference presents a challenging classification problem, namely, given a diverse array of multimodal features, how can we infer or classify driver intentions? While we certainly may pose a large number of candidate intentions, as already mentioned, we will focus 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 [2].

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 a road curvature metric, heading, lateral lane position, lateral lane position 10 m ahead, and lateral lane position 20 m 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 SBL, as described next.

A. Sparse Bayesian Learning (SBL)

SBL is a powerful approach that is recently introduced into the machine learning literature for solving regression and classification problems [14]. The methodology relies on a parameterized prior that encourages models with few nonzero weights. As such, SBL is particularly 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

$$y(\boldsymbol{x}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{x}) \tag{9}$$

where $\boldsymbol{x}$ is an input feature vector (described below), w_i's are learned model weights, and $\phi_i(\cdot)$'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, which is a Bayesian competitor to the popular support vector machine. 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 simultaneously employ multiple (complete) kernels and bases, as well as any other derived features we think might improve classification performance. To avoid overfitting with so many candidate features, an independent Gaussian prior distribution is applied to each of the unknown weights, each with its own independent variance parameter. During a training process called evidence maximization, these M variance parameters are estimated from the data, with many of them being forced to zero if the associated feature is deemed irrelevant. As a result, the corresponding posterior weights are necessarily pushed to zero, and therefore, unnecessary features are automatically pruned. A more comprehensive description of SBL can be found in [14]. For our purposes, SBL acts 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

$$\begin{aligned}\boldsymbol{x}(t) = [&\text{LateralPos}(t), \ldots, \text{LateralPos}(t-N+1)\\ &\text{Heading}(t), \ldots, \text{Heading}(t-N+1); \text{etc.}]\end{aligned} \tag{10}$$

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 1-s-long sliding window of data. Derived features based on aggregate statistics of these temporal blocks (e.g., median values) were also included. 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, which are important for the classification problem. SBL effectively discovers the temporal ordering in adjusting the weights for specific features at specific times. 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. However, the method could easily be adapted to handle HMM-based features, if desired.

SBL is particularly well suited for computer vision applications for a number of reasons. For example, 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.

B. 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 want 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. In addition, 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 change maneuvers spaced throughout the recording. Now, because our DIIS outputs a number bounded between zero and one at every time instant t, i.e., $P(C|\boldsymbol{x}(t))$, where C represents the class "lane change," we may always pick some threshold $\mathcal{T}$ and then decide

$$\begin{array}{lll} \text{IF} & P\left(C|\boldsymbol{x}(t)\right) > \mathcal{T} & \rightarrow \text{lane change is occuring} \\ \text{ELSE} & & \rightarrow \text{lane keeping.} \end{array}$$

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 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 tradeoffs between higher false alarm rates and increased detection accuracy. Moreover, it naturally solves the problems that are raised above, and as discussed next, it addresses specific DIIS ideological concerns.

C. Ideological Issues

The goal of our DIIS 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.

1) intentional decision to change lanes followed by an actual lane change execution (common);
2) alert lane keeping (common);
3) 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);
4) 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 3)? 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 1) or case 3) examples] and a separate file with numerous lane changes, either type 1) or 3). By focusing only on the lane changes in the latter, we need not worry about falsely categorizing the type 3) cases.

Second, suppose now that no examples of case 3) exist, i.e., all lane change decisions are promptly followed by an actual lane change maneuver. Thus, we only need to consider 1), 2), and 4). A robust DIIS should separate 1) from 2) and 4), which are both lane keeping events; however, a trajectory-forecasting-based approach will often separate 1) and 4) from 2). Moreover, the algorithms will incur a small penalty for this mistake since case 4) is a relatively rare occurrence.

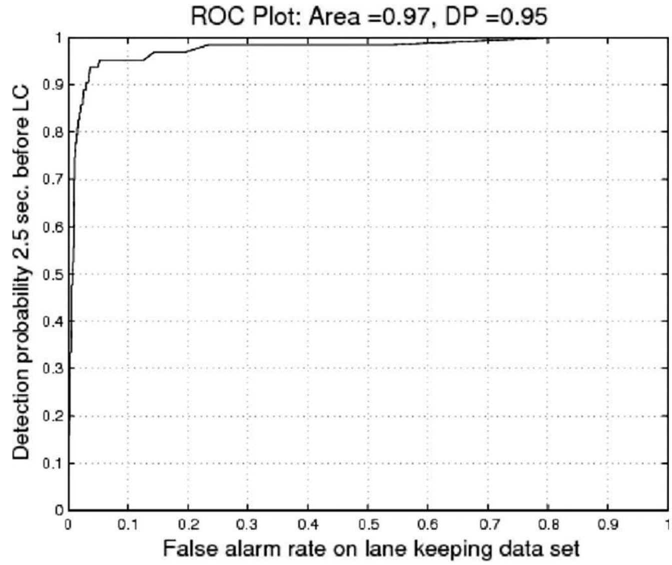


Fig. 11. ROC curve obtained from 2.5 s before a lane change.

While type 4) 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) seems to help bridge the gap between trajectory forecasting and driver intent inference.

IV. DIIS RESULTS

To test our full DIIS system and compute the evaluation statistics described above, we collected significant lane keeping and lane changing data sets 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 4)-like cases, allowing us to better see the distinction between trajectory forecasting and intent inference. Results are shown below in Figs. 11 and 12, 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 Figs. 13 and 14. 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 s before the lane change is about equivalent to the classifier performance when we do not include the lane change at 2.5 s. We therefore can

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

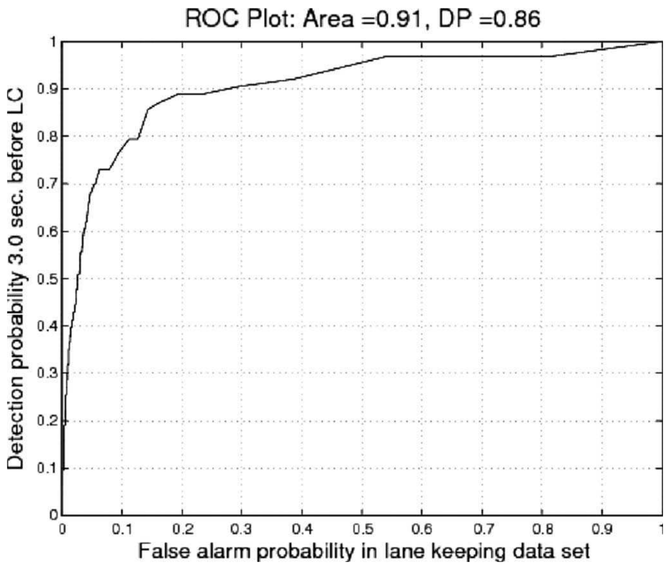


Fig. 12. ROC curve obtained from 3.0 s before a lane change.

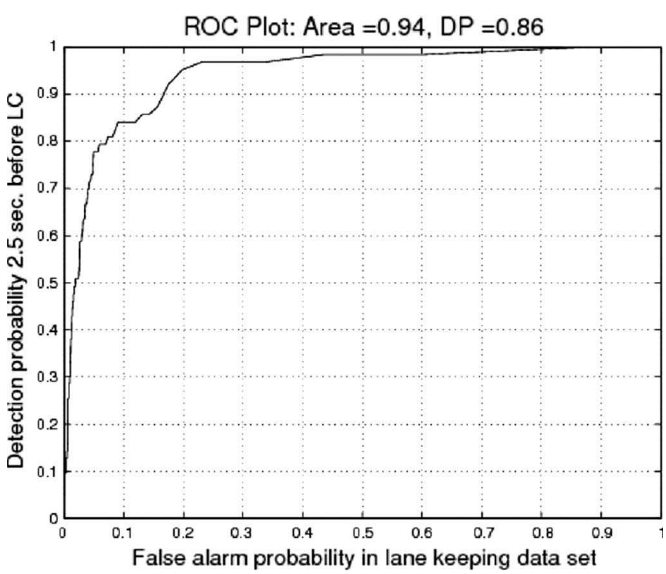


Fig. 13. Using no driver state information (i.e., pure trajectory forecasting), ROC curve obtained from 2.5 s before a lane change.

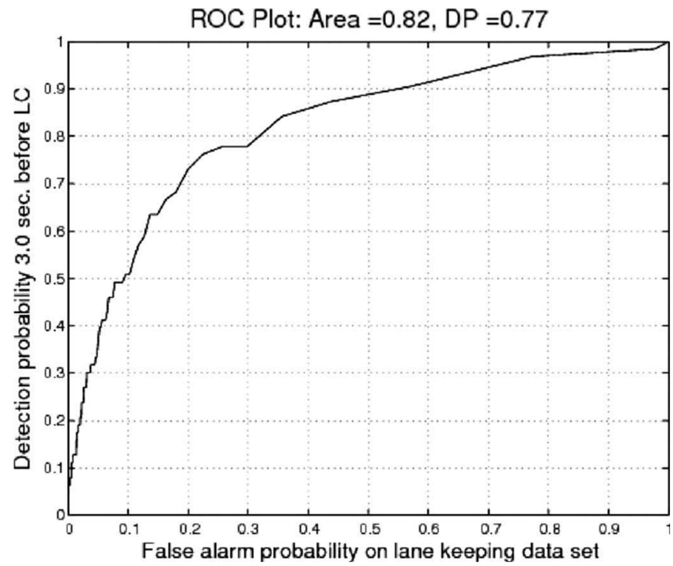


Fig. 14. Using no driver state information (i.e., pure trajectory forecasting), ROC curve obtained from 3.0 s before a lane change.

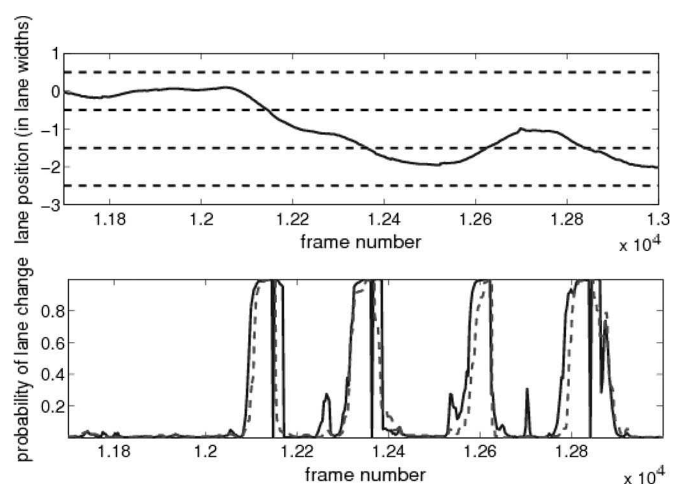


Fig. 15. Lane position versus frame number (top) and the probability of a lane change versus frame number (bottom) for a driving sequence containing lane changes. In the bottom graph, the solid line is the probability of a lane change using head movement data, and the dashed line is the probability of a lane change without using head movement data.

provide an accurate estimate earlier when head movement data are included in the feature vector. This is further illustrated by looking at a time series of lane change maneuvers. Fig. 15 shows the lane position versus frame number in the top graph and the lane change probability versus frame number in the bottom graph. The solid line represents the lane change probability when the head movement is included, and the dashed line represents the lane change probability when the 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 change performed in combination with a previous lane change (Fig. 15, frame 12 300). In this situation, the classifier performs only slightly better than the classifier that ignores the head movement. Fig. 16 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 12 132 [Fig. 16(d)].

V. Summary and Conclusion

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.
3) The observable data were consistent with multiple driver intents.

In this paper, we have tried to address each of these issues. First, by developing a more general lane tracker that is capable of robustly handling various types of lane markings, we create a more accurate surround map. However, lane-tracking failures

(a) frame 12060 (b) frame 12090

(c) frame 12120 (d) frame 12132

(e) frame 12180 (f) frame 12210

Fig. 16. Frames from the video analyzed in Fig. 15. (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.

can occur, particularly in situations where there are harsh lighting conditions, heavy traffic and occlusion of the lane markings, extremely poor road, or lane marking conditions. A more complete analysis of these types of errors can be found in [17]. Second, by incorporating a state-of-the-art SBL classifier with well-motivated evaluation metrics, we reduce the likelihood of DIIS algorithmic failures. However, our system is reliant on labeled training data and, therefore, is subject to errors when approaching situations that are outside of those presented to the system in training. Third, by incorporating driver state information and moving away from simple trajectory forecasting, we increase the likelihood that the observable data are consistent with one and only one driver intent. Our driver state information comes from the head motion of the driver. More detailed information about the drivers state, such as fatigue level or eye gaze vector, could provide more information and possibly increase classifier performance. Qualitatively, the errors occurring at larger prediction times tended to be caused by item 3): The observable data were consistent with multiple intents. This makes intuitive sense as well, since driving behavior that occurs at further intervals ahead of a lane change will likely resemble the driving behavior during normal lane keeping. In contrast, for shorter prediction times, the errors were often caused by item 1): errors in the lane tracker. 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 DIIS.

Acknowledgment

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] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image Vis. Comput.*, vol. 21, no. 4, pp. 359–381, Apr. 2003.

[2] N. Kuge, T. Yamamura, and O. Shimoyama, *A Driver Behavior Recognition Method Based on a Driver Model Framework*. Warrendale, PA: SAE, 1998.

[3] D. D. Salvucci and A. Liu, "The time course of a lane change: Driver control and eye-movement behavior," *Transp. Res. Part F*, vol. 5, no. 2, pp. 123–132, 2002.

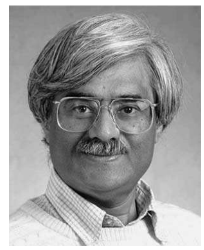
[4] N. Oliver and A. Pentland, “Graphical models for driver behavior recognition in a smartcar,” in *Proc. IEEE Int. Conf. Intell. Veh.*, Oct. 2000, pp. 7–12.
[5] J. McCall, D. Wipf, M. M. Trivedi, and B. Rao, “Lane change intent analysis using robust operators and sparse Bayesian learning,” in *Proc. IEEE Int. Workshop Mach. Vis. Intell. Veh.*, Jun. 2005, p. 59.
[6] R. Risack, N. Mohler, and W. Enkelmann, “A video-based lane keeping assistant,” in *Proc. IEEE Intell. Veh. Symp.*, Oct. 2000, pp. 506–511.
[7] J. W. Lee, C.-D. Kee, and U. K. Yi, “A new approach for lane departure identification,” in *Proc. IEEE Intell. Veh. Symp.*, 2003, pp. 100–105.
[8] W. Enkelmann, “Video-based driver assistance—From basic functions to applications,” *Int. J. Comput. Vis.*, vol. 45, no. 3, pp. 201–221, Dec. 2001.
[9] D. D. Salvucci, “Inferring driver intent: A case study in lane-change detection,” in *Proc. Human Factors Ergonom. Soc. 48th Annu. Meeting*, 2004, pp. 2228–2231.
[10] J. B. McDonald, “Detecting and tracking road markings using the hough transform,” in *Proc. Irish Mach. Vis. Image Process. Conf.*, 2001.
[11] D. Pomerleau, “Ralph: Rapidly adapting lateral position handler,” in *Proc. IEEE Intell. Veh. Symp.*, 1995, pp. 54–59.
[12] B. Southall and C. J. Taylor, “Stochastic road shape estimation,” in *Proc. Int. Conf. Comput. Vis.*, 2001, pp. 205–212.
[13] S. Gehrig, A. Gern, S. Heinrich, and B. Woltermann, “Lane recognition on poorly structured roads—The bots dot problem in California,” in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst.*, 2002, pp. 67–71.
[14] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.
[15] J. McCall, O. Achler, and M. M. Trivedi, “Design of an instrumented vehicle testbed for developing human centered driver support system,” in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun. 2004, pp. 483–488.
[16] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.
[17] J. McCall and M. M. Trivedi, “Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.
[18] H. Schneiderman and T. Kanade, “Probabilistic modeling of local appearance and spatial relationships for object recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1998, pp. 45–51.
[19] J. Jain and A. Jain, “Displacement measurement and its application in interframe image coding,” *IEEE Trans. Commun.*, vol. COM-29, no. 12, pp. 1799–1808, Dec. 1981.

Joel C. McCall received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, in 1999, the M.S. degree in electrical and computer engineering from the University of California, San Diego (UCSD), in 2003, and the Ph.D. degree in electrical and computer engineering specializing in intelligent systems, robotics, and control from the UCSD in 2006.

He is currently with the Mobile Devices Division, Microsoft Corporation, Redmond, WA. His research focuses on human-centered driver assistance and safety systems.

David P. Wipf received the B.S. degree in electrical engineering from the University of Virginia, Charlottesville, and the M.S. degree from the University of California, San Diego (UCSD), in 2003. He is currently working toward the Ph.D. degree in electrical and computer engineering with an emphasis on statistical signal processing and machine learning at UCSD.

His research involves the analysis of Bayesian algorithms for learning sparse representations.



Mohan M. Trivedi received the B.E. degree (with honors) from the Birla Institute for Technology and Science in India, in 1974 and the Ph.D. degree from Utah State University, Logan, in 1979.

He is a Professor of electrical and computer engineering and the founding Director of the Computer Vision and Robotics Research Laboratory, University of California at San Diego (UCSD), La Jolla, CA. He has a broad range of research interests in the intelligent systems, computer vision, intelligent (“smart”) environments, intelligent vehicles and transportation systems, and human-machine interfaces areas. In partnership with several automobile companies, he established the Laboratory for Intelligent and Safe Automobiles ("LISA") at UCSD to pursue a multidisciplinary research agenda. He has published nearly 300 technical articles and has edited over a dozen volumes, including books, special issues, video presentations, and conference proceedings.

Dr. Trivedi serves on the executive committee of the California Institute for Telecommunication and Information Technologies [Cal-IT2] as the Leader of the Intelligent Transportation and Telematics Layer at UCSD. He serves regularly as a Consultant to industry and government agencies in the USA and abroad. He was the Editor-in-Chief of the Machine Vision and Applications Journal. He served as the Chairman of the Robotics Technical Committee of the Computer Society of the IEEE and on the ADCOM of the IEEE SMC Society. He received the Distinguished Alumnus Award from the Utah State University, Pioneer Award (Technical Activities), and Meritorious Service Award from the IEEE Computer Society.

Bhaskar D. Rao (S’80–M’83–SM’91–F’00) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, India, in 1979 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1981 and 1983, respectively.

Since 1983, he has been with the University of California, San Diego, where he is currently a Professor with the Electrical and Computer Engineering Department. His research interests are in the areas of digital signal processing, estimation theory, and optimization theory, with applications to digital communications, speech signal processing, and human–computer interactions.

Dr. Rao has been a member of the Statistical Signal and Array Processing Technical Committee and the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society. He is currently a member of the Signal Processing for Communications Technical Committee and serves on the editorial board of the *EURASIP Signal Processing Journal*.