Know Before You Do: Anticipating Maneuvers via Learning Temporal Driving Models

Ashesh Jain, Hema S Koppula, Bharad Raghavan, Shane Soh, Ashutosh Saxena
{ashesh,hema,asaxena}@cs.cornell.edu {bharad,shanesoh}@stanford.edu
Cornell University and Stanford University

***Abstract*—Advanced Driver Assistance Systems (ADAS) have made driving safer over the last decade. They prepare vehicles for unsafe road conditions and alert drivers if they perform a dangerous maneuver. However, many accidents are unavoidable because by the time drivers are alerted, it is already too late. Anticipating maneuvers a few seconds beforehand can alert drivers before they perform the maneuver and also give ADAS more time to avoid or prepare for the danger. Anticipation requires modeling the driver's action space, events inside the vehicle such as their head movements, and also the outside environment. Performing this joint modeling makes anticipation a challenging problem.**

In this work we anticipate driving maneuvers a few seconds before they occur. For this purpose we equip a car with cameras and a computing device to capture the context from both inside and outside of the car. We represent the context with expressive features and propose an Autoregressive Input-Output HMM to model the contextual information. We evaluate our approach on a diverse data set with 1180 miles of natural freeway and city driving and show that we can anticipate maneuvers 3.5 seconds before they occur with over 80% F1-score. Our computation time during inference is under 3.6 milliseconds.

I. Introduction

Over the last decade cars have been equipped with various assistive technologies in order to provide a safe driving experience. Technologies such as lane keeping, blind spot check, pre-crash systems etc., are successful in alerting drivers whenever they commit a dangerous maneuver [23, 24]. Still in the US alone more than 33,000 people die in road accidents every year, the majority of which are due to inappropriate maneuvers [4]. We need mechanisms that can alert drivers *before* they perform a dangerous maneuver in order to avert many such accidents [36]. In this work we address this problem of anticipating maneuvers that a driver is likely to perform in the next few seconds (Figure 1).

Anticipating future human actions has recently been a topic of interest to both the robotics and learning communities [16, 17, 19, 50]. Figure 1 shows our system anticipating a left turn maneuver a few seconds before the car reaches the intersection. Our system also outputs probabilities over the maneuvers the driver can perform. With this prior knowledge of maneuvers, the driver assistance systems can alert drivers about possible dangers before they perform the maneuver, thereby giving them more time to react. Some previous works [12, 21, 30, 37] also predict a driver's future maneuver. However, as we show in the following sections, these methods use limited context and do not accurately model the anticipation problem.

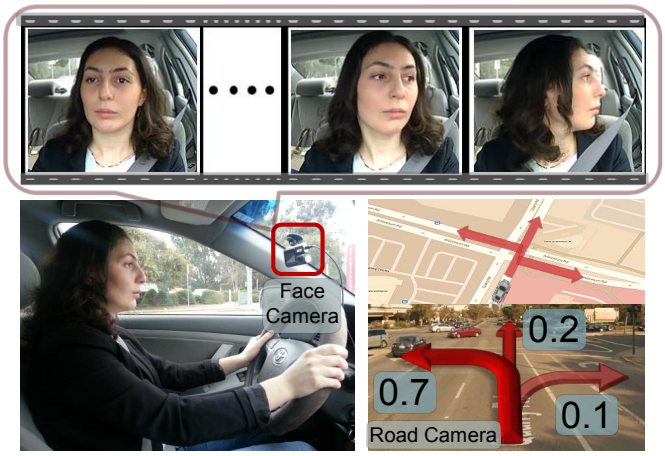


Fig. 1: **Anticipating maneuvers.** Our algorithm anticipates driving maneuvers performed a few seconds in the future. It uses information from multiple sources including videos, vehicle dynamics, GPS, and street maps to anticipate the probability of different future maneuvers.

In order to anticipate maneuvers, we reason with the contextual information from the surrounding events, which we refer to as the *driving context*. In our approach we obtain this driving context from multiple sources. We use videos of the driver inside the car and the road in front, the vehicle's dynamics, global position coordinates (GPS), and street maps; from this we extract a time series of multi-modal data from both inside and outside the vehicle. The challenge lies in modeling the temporal aspects of driving and in detecting the contextual cues that help in anticipating maneuvers.

Modeling maneuver anticipation also requires joint reasoning of the driving context and the driver's intention. The challenge here is the driver's intentions are not directly observable, and their interactions with the driving context are complex. For example, the driver is influenced by external events such as traffic conditions. The nature of these interactions is generative and they require a specially tailored modeling approach.

In this work we propose a model and a learning algorithm to capture the temporal aspects of the problem, along with the generative nature of the interactions. Our model is an Autoregressive Input-Output Hidden Markov Model (AIO-HMM) that jointly captures the context from both inside and outside the vehicle. AIO-HMM models how events from outside the vehicle affect the driver's intention, which then generates events inside the vehicle. We learn the AIO-HMM model

parameters from natural driving data and during inference output the probability of each maneuver.

We evaluate our approach on a driving data set with 1180 miles of natural freeway and city driving collected across two states – from 10 drivers and with different kinds of driving maneuvers. We demonstrate that our approach anticipates maneuvers 3.5 seconds before they occur with 80% precision and recall. We believe that our work creates scope for new ADAS features to make roads safer. In summary our key contributions are as follows:

- We propose an approach for anticipating driving maneuvers several seconds in advance.
- We model the driving context from inside and outside the car with an autoregressive input-output HMM.
- We release the first data set of natural driving with videos from both inside and outside the car, GPS, and speed information, with lane and driving maneuver annotations.

II. Related Work

Assistive features for vehicles. Recent years have seen many advances in driver assistance systems. Such systems specialize in lane departure warning, collision avoidance, traffic light detection, and other safety features [2]. These systems warn drivers when they perform a potentially dangerous maneuver [23, 24]. In contrast to these systems, our goal is to anticipate maneuvers several seconds before they occur. With anticipation, assistive systems can alert drivers before they make dangerous decisions.

Previous works have studied the driver's intent to make lane changes or turns by monitoring the vehicle's trajectory [7, 12, 21, 25, 37]. These works ignore the rich contextual information available from cameras, GPS, and street maps. The additional context from different sources also makes learning challenging, which previous works do not handle. Trivedi et al. [45] and Morris et al. [30] predict lane change intent using information from both inside and outside the vehicle. They both train a discriminative classifier which assumes that informative contextual cues always appear at a fixed time before the maneuver. We show that this assumption is not true, and in fact the temporal aspect of the problem should be carefully modeled. Our AIO-HMM takes a generative approach and handles the temporal aspect of this problem.

Anticipation and Modeling Humans. Our work is also related to previous works on modeling human motion. The modeling of human motion has given rise to many applications, anticipation being one of them. Wang et al. [50], Koppula et al. [17], and Mainprice et al. [27] demonstrate better human-robot collaboration by anticipating a human's future movements. Kitani et al. [16], Bennewitz et al. [6] and Kuderer et al. [19] model human navigation in order to anticipate the path they will follow. Dragan et al. [11] reasons for human intention for better assistive teleoperation. Similar to these works, we anticipate human actions, which are driving maneuvers in our case. However, the algorithms proposed in the previous works do not apply in our setting. In our case, anticipating maneuvers requires modeling the interaction between the driving context and the driver's intention. Furthermore, the informative cues for anticipation can appear at variable times before the maneuver. Such interactions and variability in the cues are absent in the previous works. We propose AIO-HMM to model these aspects of the problem.

Computer vision for analyzing the human face. The vision approaches related to our work are face detection and tracking [14, 40, 46, 53], building statistical models of the face [8] and pose estimation methods of the face [32, 52]. The Active Appearance Model (AAM) [8] and its variants [28, 51, 54] statistically model the shape and texture of the face. AAMs have also been used to estimate the 3D-pose of a face from a single image [52]. These vision algorithms have been used to design assistive features to monitor drivers for drowsiness and attentiveness [35, 41]. In our approach we detect and track the driver's face for anticipating maneuvers.

Learning temporal models. Temporal models are commonly used to model human activities [10, 18, 29, 48, 49]. These models have been used in both discriminative and generative fashions. The discriminative temporal models are mostly inspired by the Conditional Random Field (CRF) [22] which captures the temporal structure of the problem. Wang et al. [49] and Morency et al. [29] propose dynamic extensions of the CRF for image segmentation and gesture recognition respectively. The generative approaches for temporal modeling include various filtering methods, such as Kalman and particle filters [42], Hidden Markov Models [34], and many types of Dynamic Bayesian Networks [13, 31]. Some previous works [7, 20, 33] used HMMs to model different aspects of the driver's behaviour. Most of these generative approaches model how latent (hidden) states influence the observations. However, in our problem both the latent states and the observations influence each other. In particular, our AIO-HMM model is inspired by the Input-Output HMM [5]. In the following sections we will explain the advantages of AIO-HMM over HMMs for anticipating maneuvers and also compare its performance with variants of HMM in the experiments (Section VI).

III. Problem Overview

In this section we describe the maneuver anticipation problem and give an overview of our approach. Our goal is to anticipate driving maneuvers a few seconds before they occur. This includes anticipating a lane change before the wheels touch the lane markings or anticipating if the driver keeps straight or makes a turn when approaching an intersection.

Anticipating maneuvers is challenging for multiple reasons. First, it requires the modeling of context from different sources. Information from a single source, such as a camera capturing events outside the car, is not sufficiently rich. Additional visual information from within the car can also be used. For example, the driver's head movements are useful for anticipation – drivers typically check for the side traffic while changing lanes and scan the cross traffic at intersections.

Second, reasoning about maneuvers should take into account the driving context at both local and global levels. Local context requires modeling events in vehicle's vicinity such as the surrounding vision, GPS, and speed information. On the other hand, factors that influence the overall route contributes

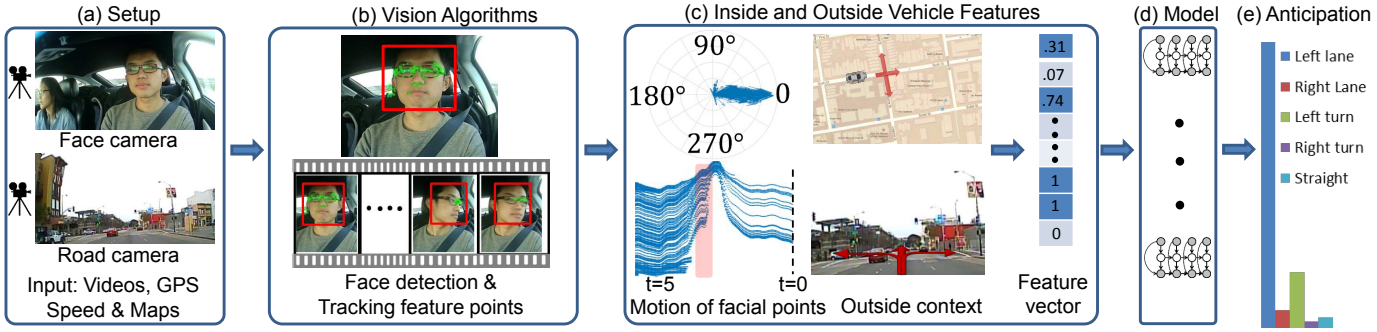


Fig. 2: **System Overview.** Our system anticipating a left lane change maneuver. (a) We process multi-modal data including GPS, speed, street maps, and events inside and outside of the vehicle using video cameras. (b) Vision pipeline extracts visual cues such as driver's head movements. (c) The inside and outside driving context is processed to extract expressive features. (d,e) Using our trained models we anticipate the probability of each maneuver.

to the global context, such as the driver's final destination. Third, the informative cues necessary for anticipation appear at variable times before the maneuver. In particular, the time interval between the driver's head movement and the occurrence of the maneuver depends on factors such as the speed, traffic conditions, the GPS location, etc.

We obtain the driving context from different sources as shown in Figure 2. Our anticipatory system includes: (1) a driver-facing camera inside the vehicle, (2) a road-facing camera outside the vehicle, (3) a speed logger, and (4) a global position coordinate (GPS) logger. The information from these sources constitute the driving context. We use the face camera for tracking the driver's head movements. The video feed from the road camera is used for extracting lane information. This information allows for additional reasoning on maneuvers that the driver is likely to perform. For example, when the vehicle is in the left-most lane, the only safe maneuvers are a right-lane change or keeping straight, unless the vehicle is approaching an intersection. Maneuvers also correlate with the vehicle's speed, e.g., turns usually happen at lower speeds than lane changes. Additionally, the GPS data augmented with the map information enables us to detect upcoming road artifacts such as intersections, highway exits, etc.

Our approach is to jointly model the driving context and the driver's intention before the maneuvers. We extract meaningful representations from the driving context (in Section V) and propose a model to handle the temporal aspects of the problem. We learn models for maneuvers and during inference we jointly anticipate the probability of each maneuver. In the next section, we describe our model and the learning algorithm.

IV. Our Approach

Driving maneuvers are influenced by multiple interactions involving the vehicle, its driver, outside traffic, and occasionally global factors like the driver's destination. These interactions influence the driver's intention, i.e. their state of mind before the maneuver, which is not directly observable. We represent the driver's intention with discrete states that are *latent* (or hidden). In order to anticipate maneuvers, we jointly model the driving context and the *latent* states in a tractable manner. We represent the driving context as a set of features, which we describe in Section V. We now present the motivation for our model and then describe the model, along with the learning and inference algorithms.

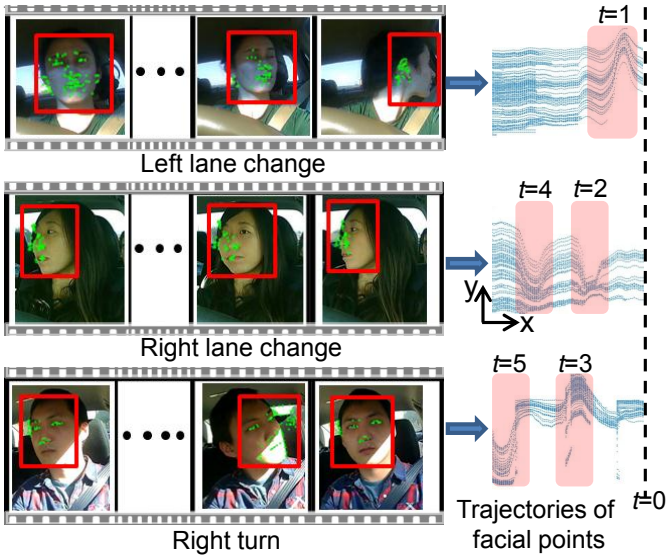


Fig. 3: **Variable time occurrence of events.** *Left*: The events inside the vehicle before the maneuvers. We track the driver's face along with many facial points. *Right*: The trajectories generated by the horizontal motion of facial points (pixels) 't' seconds before the maneuver. X-axis is the time and Y-axis is the pixels' horizontal coordinates. Informative cues appear during the shaded time interval. Such events occur at variable times before the maneuver. The order in which the cues appear is also important. For the right turn maneuver the driver first scans his right and then the traffic on his left.

A. Modeling driving maneuvers

Modeling maneuvers require temporal modeling of the driving context. The temporal aspect is critical because informative events, such as the driver's head movements, can occur at variable times before the maneuver, as illustrated in Figure 3. Discriminative methods, such as the Support Vector Machine [9] and the Relevance Vector Machine [43], which do not model the temporal aspect perform poorly (shown in Section VI-B). Therefore, a temporal model such as the Hidden Markov Model (HMM) [34] is better suited.

An HMM models how the driver's *latent* states generate both the inside driving context and the outside driving context. However, a more accurate model should capture how events *outside* the vehicle (i.e. the outside driving context) affect the driver's state of mind, which then generates the observations *inside* the vehicle (i.e. the inside driving context). More

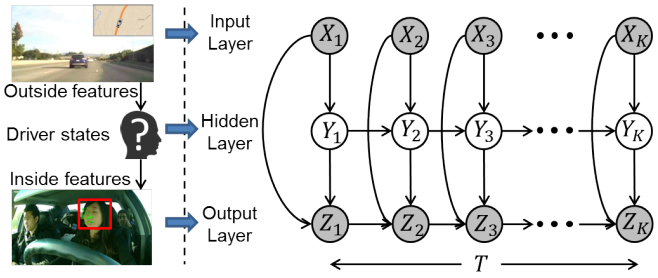


Fig. 4: **AIO-HMM.** The model has three layers: (i) Input (top): this layer represents outside vehicle features X; (ii) Hidden (middle): this layer represents driver's latent states Y; and (iii) Output (bottom): this layer represents inside vehicle features Z. This layer also captures temporal dependencies of inside vehicle features. T represents time.

specifically, the interactions between the outside events and the driver's latent states require discriminative modeling, while the interactions between the driver's latent states and the inside observations are best modeled generatively. Such interactions are well modeled by an Input-Output HMM (IOHMM) [5]. However, modeling the problem with IOHMM will not capture the temporal dependencies of the inside driving context. These dependencies are critical to capture the smooth and temporally correlated behaviours such as the driver's face movements. We therefore present Autoregressive Input-Output HMM (AIO-HMM) which extends IOHMM to model these observation dependencies. Figure 4 shows the AIO-HMM graphical model.

B. Modeling with Autoregressive Input-Output HMM

Given T seconds long driving context $\mathcal{C}$ before the maneuver M, we learn a generative model for the context $P(\mathcal{C}|M)$. The driving context $\mathcal{C}$ consists of the outside driving context and the inside driving context. The outside and inside driving contexts are temporal sequences represented by the outside features $X_1^K = \{X_1, .., X_K\}$ and the inside features $Z_1^K = \{Z_1, .., Z_K\}$ respectively. The corresponding sequence of the driver's latent states is $Y_1^K = \{Y_1, .., Y_K\}$. X and Z are vectors and Y is a discrete state.

$$\begin{aligned} P(\mathcal{C}|M) &= \sum_{Y_1^K} P(Z_1^K, X_1^K, Y_1^K|M) \\ &= P(X_1^K|M) \sum_{Y_1^K} P(Z_1^K, Y_1^K|X_1^K, M) \\ &\propto \sum_{Y_1^K} P(Z_1^K, Y_1^K|X_1^K, M) \end{aligned} \tag{1}$$

We model the correlations between X, Y and Z with an Autoregressive Input-Output HMM (AIO-HMM) as shown in Figure 4. The AIO-HMM models the distribution in equation (1). It does not assume any generative process for the outside features $P(X_1^K|M)$. It instead models them in a discriminative manner using equation (1). This captures the reasoning that events outside the vehicle affect the driver's state of mind, which then generates the events inside the vehicle before the driver performs the maneuver. The top (input) layer of the AIO-HMM consists of outside features X_1^K. The outside features then affect the driver's latent states Y_1^K, represented by the middle (hidden) layer, which then generates the inside features Z_1^K at the bottom (output) layer. The events inside the vehicle such as the driver's head movements are temporally correlated because they are generally smooth. The AIO-HMM handles these dependencies with autoregressive connections in the output layer.

Model Parameters. The AIO-HMM has two types of parameters: (i) state transition parameters $\mathbf{w}$; and (ii) observation emission parameters ($\boldsymbol{\mu}$,$\boldsymbol{\Sigma}$). We use set $\mathcal{S}$ to denote the possible latent states of the driver. For each state $Y = i \in \mathcal{S}$, we parametrize transition probabilities of leaving the state with log-linear functions, and parametrize the output layer feature emissions with normal distributions.

$$\text{Transition: } P(Y_t = j|Y_{t-1} = i, X_t; \mathbf{w}_{ij}) = \frac{e^{\mathbf{w}_{ij} \cdot X_t}}{\sum_{l \in \mathcal{S}} e^{\mathbf{w}_{il} \cdot X_t}}$$

$$\text{Emission: } P(Z_t|Y_t = i, X_t, Z_{t-1}; \boldsymbol{\mu}_{it}, \boldsymbol{\Sigma}_i) = \mathcal{N}(Z_t|\boldsymbol{\mu}_{it}, \boldsymbol{\Sigma}_i)$$

The inside (vehicle) features represented by the output layer are jointly influenced by all three layers. These interactions are modeled by the mean and variance of the normal distribution. We model the mean of the distribution using the outside and inside features from the vehicle as follows:

$$\boldsymbol{\mu}_{it} = (1 + \mathbf{a}_i \cdot X_t + \mathbf{b}_i \cdot Z_{t-1})\boldsymbol{\mu}_i$$

In the equation above, $\mathbf{a}_i$ and $\mathbf{b}_i$ are parameters that we learn for every state $i \in \mathcal{S}$. Therefore, the parameters we learn for state $i \in \mathcal{S}$ are $\boldsymbol{\theta}_i = \{\boldsymbol{\mu}_i, \mathbf{a}_i, \mathbf{b}_i, \boldsymbol{\Sigma}_i$ and $\mathbf{w}_{ij}|j \in \mathcal{S}\}$, and the overall AIO-HMM parameters are $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i|i \in \mathcal{S}\}$.

C. Learning AIO-HMM parameters

The training data $\mathcal{D} = \{(X_{1,n}^{K_n}, Z_{1,n}^{K_n})|n = 1, .., N\}$ consists of N instances of a maneuver M. The goal is to maximize the data log-likelihood.

$$l(\boldsymbol{\Theta}; \mathcal{D}) = \sum_{n=1}^{N} \log P(Z_{1,n}^{K_n}|X_{1,n}^{K_n}; \boldsymbol{\Theta}) \tag{2}$$

Directly optimizing equation (2) is challenging because parameters Y representing the driver's states are *latent*. We therefore use the iterative EM procedure to learn the model parameters. In EM, instead of directly maximizing equation (2), we maximize its simpler lower bound. We estimate the lower bound in the E-step and then maximize that estimate in the M-step. These two steps are then iteratively repeated.

E-step. In the E-step we get the lower bound of equation (2) by calculating the expected value of the *complete* data log-likelihood using the current estimate of the parameter $\hat{\boldsymbol{\Theta}}$.

$$\text{E-step: } Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) = E[l_c(\boldsymbol{\Theta}; \mathcal{D}_c)|\hat{\boldsymbol{\Theta}}, \mathcal{D}] \tag{3}$$

where $l_c(\boldsymbol{\Theta}; \mathcal{D}_c)$ is the log-likelihood of the *complete* data $\mathcal{D}_c$ defined as:

$$\mathcal{D}_c = \{(X_{1,n}^{K_n}, Z_{1,n}^{K_n}, Y_{1,n}^{K_n})|n = 1, .., N\} \tag{4}$$

$$l_c(\boldsymbol{\Theta}; \mathcal{D}_c) = \sum_{n=1}^{N} \log P(Z_{1,n}^{K_n}, Y_{1,n}^{K_n}|X_{1,n}^{K_n}; \boldsymbol{\Theta}) \tag{5}$$

We should note that the occurrences of hidden variables Y in $l_c(\boldsymbol{\Theta}; \mathcal{D}_c)$ are marginalized in equation (3), and hence Y need not be known. We efficiently estimate $Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}})$ using the forward-backward algorithm [31].

M-step. In the M-step we maximize the expected value of the complete data log-likelihood $Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}})$ and update the model

parameter as follows:

$$\text{M-step: } \boldsymbol{\Theta} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}; \hat{\boldsymbol{\Theta}}) \tag{6}$$

Solving equation (6) requires us to optimize for the parameters $\boldsymbol{\mu}$, $\mathbf{a}$, $\mathbf{b}$, $\boldsymbol{\Sigma}$ and $\mathbf{w}$. We optimize all parameters expect $\mathbf{w}$ exactly by deriving their closed form update expressions. We optimized $\mathbf{w}$ using the gradient descent. Refer to supplementary material for detailed E and M steps.

D. Inference of Maneuvers

We now describe how we use the learned models to anticipate maneuvers in driving scenarios not seen during training. Our learning algorithm trains separate AIO-HMM models for each maneuver. The goal during inference is to determine which model best explains the past T seconds of the driving context. We evaluate the likelihood of the inside and outside feature sequences (Z_1^K and X_1^K) for each maneuver, and anticipate the probability P_M of each maneuver M as follows:

$$P_M = P(M|Z_1^K, X_1^K) \propto P(Z_1^K, X_1^K|M)P(M) \tag{7}$$

Algorithm 1 shows the complete inference procedure. The inference in equation (7) simply requires a forward-pass [31] of the AIO-HMM, the complexity of which is $\mathcal{O}(K(|\mathcal{S}|^2 + |\mathcal{S}||Z|^3 + |\mathcal{S}||X|))$. However, in practice it is only $\mathcal{O}(K|\mathcal{S}||Z|^3)$ because $|Z|^3 \gg |S|$ and $|Z|^3 \gg |X|$. Here $|\mathcal{S}|$ is the number of discrete latent states representing the driver's intention, while $|Z|$ and $|X|$ are the dimensions of the inside and outside feature vectors respectively. In equation (7) $P(M)$ is the prior probability of maneuver M. We assume an uninformative uniform prior over all the maneuvers.

Algorithm 1 Anticipating maneuvers

input Driving videos, GPS, Maps and Vehicle Dynamics
output Probability of each maneuver
  Initialize the face tracker with the driver's face
  while *driving* **do**
    Track the driver's face [46]
    Extract features Z_1^K and X_1^K (Sec. V)
    Inference $P_M = P(M|Z_1^K, X_1^K)$ (Eq. (7))
    Send the inferred probability of each maneuver to ADAS
  end while

V. Features

We now describe the features we extract for anticipating maneuvers. We extract features by processing the inside and outside driving contexts. We denote the inside features as Z and the outside features as X.

A. Inside-vehicle features.

The inside features Z capture the driver's head movements. Our autonomous vision pipeline consists of face detection, tracking, and feature extraction modules. The face detection and tracking modules track the driver's face in the videos from the driver-facing camera and generate face tracks (described below). These face tracks are fed to the feature extraction module, which then extracts the head motion features per frame, denoted by ϕ(face). For AIO-HMM, we compute Z by aggregating ϕ(face) for every 20 frames, i.e., $Z = \sum_{i=1}^{20} \phi(\text{face}_i)/\| \sum_{i=1}^{20} \phi(\text{face}_i)\|$.

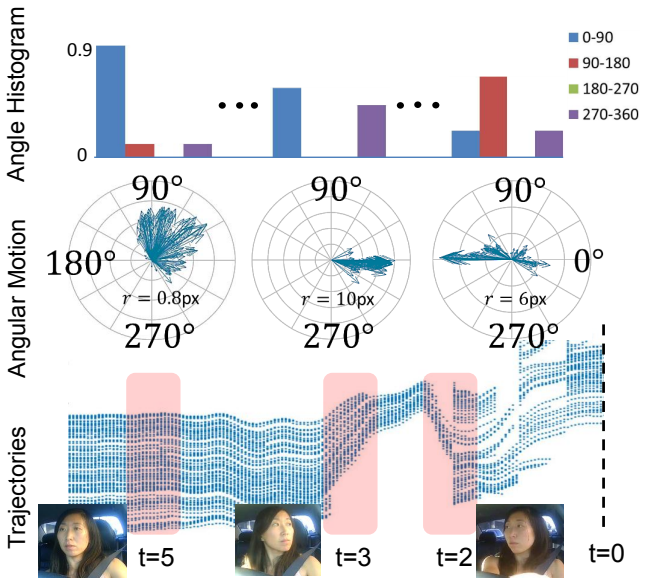


Fig. 5: **Inside vehicle feature extraction.** The angular histogram features extracted at three different time steps for a left turn maneuver. *Bottom*: Trajectories for the horizontal motion of tracked facial pixels 't' seconds before the maneuver. X-axis is the time and Y-axis is the pixels' horizontal coordinates. At t=5 seconds before the maneuver the driver is looking straight, at t=3 looks (left) in the direction of maneuver, and at t=2 looks (right) in opposite direction for the crossing traffic. *Middle*: Average motion vector of tracked facial pixels in polar coordinates. r is the average movement of pixels and arrow indicates the direction in which the face moves when looking from the camera. *Top*: Normalized angular histogram features.

Face detection and tracking. We detect the driver's face using a trained Viola-Jones face detector [46]. From the detected face, we first extract visually discriminative (facial) points using the Shi-Tomasi corner detector [38] and then track those facial points using the Kanade-Lucas-Tomasi tracker [26, 38, 44]. However, the tracking may accumulate errors over time because of changes in illumination due to the shadows of trees, traffic, etc. We therefore constrain the tracked facial points to follow a projective transformation and remove the incorrectly tracked points using the RANSAC algorithm. While tracking the facial points, we lose some of the tracked points with every new frame. To address this problem, we re-initialize the tracker with new discriminative facial points once the number of tracked points falls below a threshold [15]. Some tracking results and failure cases are available here: `https://sites.google.com/site/brainforcars/`

Head motion features. For maneuver anticipation the horizontal movement of the face and its angular rotation (*yaw*) are particularly important. From the face tracking module we obtain *face tracks*, which are 2D trajectories of the tracked facial points in the image plane. Figure 5 (bottom) shows how the horizontal coordinates of the tracked facial points vary with time before a left turn maneuver. We represent the driver's face movements and rotations with histogram features. In particular, we take matching facial points between successive frames and create histograms of their corresponding horizontal motions (in pixels) and angular motions in the image plane (Figure 5). We bin the horizontal and angular

Fig. 6: **Our data set** is diverse in drivers, landscape, and weather.

motions using $[\leq -2,\ -2$ to $0,\ 0$ to $2,\ \geq 2]$ and $[0$ to $\frac{\pi}{2},\ \frac{\pi}{2}$ to $\pi,\ \pi$ to $\frac{3\pi}{2},\ \frac{3\pi}{2}$ to $2\pi]$, respectively. We also calculate the mean movement of the driver's face center. This gives us $\phi(\text{face}) \in \mathbb{R}^9$ facial features for each frame.

B. Outside-vehicle features.

The outside feature vector X encodes the information about the outside environment such as the road conditions, vehicle dynamics, etc. In order to get this information, we use the road-facing camera together with the vehicle's GPS coordinates, the speed, and the street maps. More specifically, we obtain two binary features from the road-facing camera indicating whether a lane exists on the left side and on the right side of the vehicle. We also augment the vehicle's GPS coordinates with the street maps and extract a binary feature indicating if the vehicle is within 15 meters of a road artifact such as intersections, turns, highway exists, etc. In order to represent the influence of the vehicle's speed on the maneuvers, we encode the average, maximum, and minimum speeds of the vehicle over the last 5 seconds as features. This results in a $X \in \mathbb{R}^6$ dimensional outside feature vector.

VI. Experiment

In this section we present the evaluation of our approach on a driving data set. We first give an overview of our data set, the baseline algorithms, and our evaluation setup. We then present the results and discussion.

A. Experimental Setup

Data set. Our data set consists of natural driving videos with both inside and outside views of the car, its speed, and the global position system (GPS) coordinates.[1] The inside car video captures the driver and passengers, and the outside car video captures the view of the road ahead.

We collected this driving data set under fully natural settings without any intervention.[2] It consists of 1180 miles of freeway and city driving and encloses 21,000 square miles across two states.[3] We collected this data set from 10 drivers over a period of two months. The complete data set has a total of 2 million video frames and includes diverse landscapes and weather conditions. Figure 6 shows a few samples from our data set. We annotated the driving videos with a total of 700 events containing 274 lane changes, 131 turns, and 295 randomly sampled instances of driving straight. Each lane change or turn annotation marks the start time of the maneuver, i.e., before the car touches the lane or yaws, respectively. For all annotated events, we also annotated the lane information, i.e., the number of lanes on the road and the current lane of the car.

Baseline algorithms. We compare our method against the following baselines:

- *Chance:* Anticipations are chosen uniformly at random.
- *SVM [30]:* Support Vector Machine is a maximum margin discriminative classifier [9]. Morris et al. [30] takes this approach for anticipating maneuvers.[4] We train the SVM on 5 seconds of driving context by concatenating all frame features to get a $\mathbb{R}^{3840}$ dimensional feature vector.
- *Random-Forest [39]:* This is also a discriminative classifier that learns many decision trees from the training data, and at test time it averages the prediction of the individual decision trees. We train it on the same features as SVM with 150 decision trees of depth ten each.
- *HMM:* This models the contextual features with a Hidden Markov Model. We train the HMM on a temporal sequence of feature vectors that we extract every 0.8 seconds, i.e., every 20 video frames. We consider three versions of the HMM: (i) HMM E: with only outside features from the road camera, the vehicle's speed, GPS and street maps (Section V-B); (ii) HMM F: with only inside features from the driver's face (Section V-A); and (ii) HMM $E + F$: with both inside and outside features.

We compare these baseline algorithms with our input-output models, IOHMM and AIO-HMM. The features for our model are extracted in the same manner as in HMM $E + F$ method.

Evaluation setup. We evaluate an algorithm based on its correctness in predicting future maneuvers. In particular, we anticipate maneuvers every 0.8 seconds (20 video frames) where the algorithm processes the recent context and assigns a probability to each of the four maneuvers: {*left lane change, right lane change, left turn, right turn*} and a probability to the event of *driving straight*. These five probabilities together sum to one. After anticipation, i.e. when the algorithm has computed all five probabilities, the algorithm predicts a maneuver if its probability is above a threshold. If none of the maneuvers' probabilities are above this threshold, the algorithm does not make a maneuver prediction and predicts *driving straight*. However, when it predicts one of the four maneuvers, it sticks with this prediction and makes no further predictions for next 5 seconds or until a maneuver occurs, whichever happens earlier. After 5 seconds or a maneuver has occurred, it returns to anticipating future maneuvers.

[1]The inside and outside cameras operate at 25 and 30 frames/sec, and output frames of resolution 1920x1080 and 640x480 pixels, respectively. The distance between successive GPS coordinates is 2 meters on average.

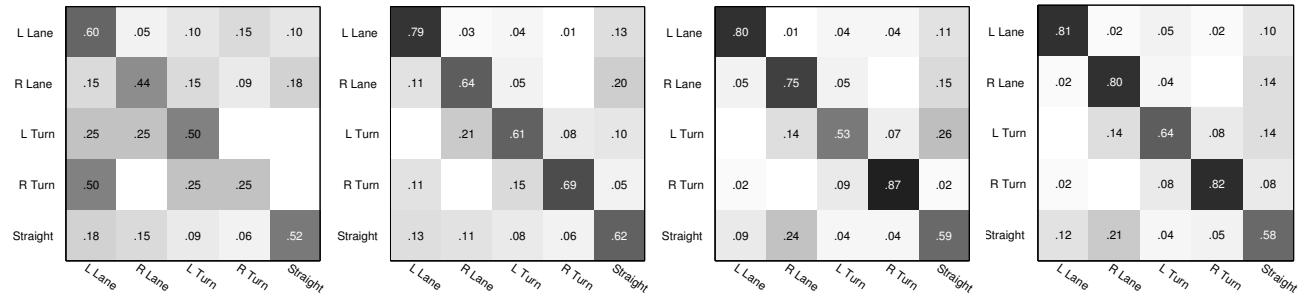
[2]**Collection protocol:** We set up cameras, GPS and speed recording device in subject's personal vehicles and left it to record the data for several weeks. The subjects were asked to ignore our setup and drive as they would normally.

[3]**Driving map in the supplementary material.**

[4]Morries et al. [30] considered binary classification problem (lane change vs driving straight) and used Relevance Vector Machine (RVM) [43]. However, we consider multiple maneuvers and for such multi-class problems RVM pseudo likelihoods are not comparable. In practice, for binary classification problems SVM and RVM give similar performance[31].

TABLE I: **Results on our driving data set**, showing average *precision*, *recall* and *time-to-maneuver* computed from 5-fold cross-validation. The number inside parenthesis is the standard error.

Algorithm	Lane change			Turns			All maneuvers		
	Pr (%)	Re (%)	Time-to-maneuver (s)	Pr (%)	Re (%)	Time-to-maneuver (s)	Pr (%)	Re (%)	Time-to-maneuver (s)
Chance	33.3	33.3	-	33.3	33.3	-	20.0	20.0	-
Morris et al. [30] SVM	73.7 (3.4)	57.8 (2.8)	2.40	64.7 (6.5)	47.2 (7.6)	2.40	43.7 (2.4)	37.7 (1.8)	1.20
Random-Forest	71.2 (2.4)	53.4 (3.2)	3.00	68.6 (3.5)	44.4 (3.5)	1.20	51.9 (1.6)	27.7 (1.1)	1.20
HMM E	75.0 (2.2)	60.4 (5.7)	3.46	74.4 (0.5)	66.6 (3.0)	4.04	63.9 (2.6)	60.2 (4.2)	3.26
HMM F	76.4 (1.4)	75.2 (1.6)	3.62	75.6 (2.7)	60.1 (1.7)	3.58	64.2 (1.5)	36.8 (1.3)	2.61
HMM $E+F$	80.9 (0.9)	79.6 (1.3)	3.61	73.5 (2.2)	75.3 (3.1)	4.53	67.8 (2.0)	67.7 (2.5)	3.72
(Our method) IOHMM	81.6 (1.0)	**79.6 (1.9)**	3.98	77.6 (3.3)	**75.9 (2.5)**	4.42	74.2 (1.7)	71.2 (1.6)	3.83
(Our final method) AIO-HMM	**83.8 (1.3)**	79.2 (2.9)	3.80	**80.8 (3.4)**	75.2 (2.4)	4.16	**77.4 (2.3)**	**71.2 (1.3)**	3.53



(a) SVM [30] *(b)* HMM $E+F$ *(c)* IOHMM *(d)* AIO-HMM

Fig. 7: **Confusion matrix** of different algorithms when jointly predicting all the maneuvers. Predictions made by algorithms are represented by rows and actual maneuvers are represented by columns. Numbers on the diagonal represent precision. (**More matrices in supplementary**.)

During this process of anticipation and prediction, the algorithm makes (i) true predictions (tp): when it predicts the correct maneuver; (ii) false predictions (fp): when it predicts a maneuver but the driver performs a different maneuver; (iii) false positive predictions (fpp): when it predicts a maneuver but the driver does not perform any maneuver (i.e. *driving straight*); and (iv) missed predictions (mp): when it predicts *driving straight* but the driver performs a maneuver. We evaluate the algorithms using their precision and recall scores:

$$Pr = \frac{tp}{\underbrace{tp+fp+fpp}_{\text{Total \# of maneuver predictions}}}; \quad Re = \frac{tp}{\underbrace{tp+fp+mp}_{\text{Total \# of maneuvers}}}$$

The precision measures the fraction of the predicted maneuvers that are correct and recall measures the fraction of the maneuvers that are correctly predicted. For true predictions (tp) we also compute the average *time-to-maneuver*, where time-to-maneuver is the interval between the time of algorithm's prediction and the start of the maneuver.

In our experiments we perform cross validation to choose the number of the driver's latent states in the AIO-HMM and the threshold on probabilities for maneuver prediction. For *SVM* we cross-validate for the parameter C and the choice of kernel from Gaussian and polynomial kernels. The parameters are chosen as the ones giving the highest F1-score on a validation set. The F1-score is the harmonic mean of the precision and recall, defined as $F1 = 2 * Pr * Re/(Pr + Re)$.

B. Results and Discussion

We evaluate the algorithms on maneuvers that were not seen during training and report the results using 5-fold cross validation. Table I reports the precision and recall scores under three settings: (i) *Lane change*: when the algorithms only predict for the left and right lane changes. This setting is relevant for highway driving where the prior probabilities of turns are low; (ii) *Turns*: when the algorithms only predict for the left and right turns; and (iii) *All maneuvers*: in this setting the algorithms jointly predict all four maneuvers. All three settings include the instances of *driving straight*.

As shown in Table I, the AIO-HMM performs better than the other algorithms. Its precision is over 80% for the *lane change* and *turns* settings. For jointly predicting all the maneuvers its precision is 77%, which is 34% higher than the previous work by Morris et al. [30] and 26% higher than the Random-Forest. The AIO-HMM recall is always comparable or better than the other algorithms. On average the AIO-HMM predicts maneuvers 3.5 seconds before they occur and up to 4 seconds earlier when only predicting turns. On the other hand, Morris et al. [30] predicts only 1.2 to 2.4 seconds in advance.

Figure 7 shows the confusion matrix plots for jointly anticipating all the maneuvers. AIO-HMM gives the highest precision for each maneuver. Modeling maneuver anticipation with an input-output model enjoys two benefits: (i) it models the more accurate reasoning that events outside the vehicle affect the driver's state of mind which then generates events inside the vehicle; (ii) it also allows for a discriminative modeling of the state transition probabilities using rich features from outside the vehicle. On the other hand, the HMM $E+F$ solves a harder problem by learning a generative model of the outside and inside features together. As shown in Table I, the precision of HMM $E+F$ is 10% less than that of AIO-HMM for jointly predicting all the maneuvers.

Table II compares the fpp of different algorithms. False positive predictions (fpp) happen when an algorithm wrongly predicts *driving straight* as one of the maneuvers. Therefore low value of fpp is preferred. HMM F performs best on this metric with a fpp of 11% as it mostly assigns a high probability to *driving straight*. However, due to this reason, it incorrectly predicts *driving straight* even when drivers perform

TABLE II: **False positive prediction** (fpp) of different algorithms. The number inside parenthesis is the standard error.

Algorithm	Lane change	Turns	All
Morris et al. [30] SVM	15.3 (0.8)	13.3 (5.6)	24.0 (3.5)
Random-Forest	16.2 (3.3)	12.9 (3.7)	17.5 (4.0)
HMM E	36.2 (6.6)	33.3 (0.0)	63.8 (9.4)
HMM F	23.1 (2.1)	23.3 (3.1)	11.5 (0.1)
HMM $E+F$	30.0 (4.8)	21.2 (3.3)	40.7 (4.9)
IOHMM	28.4 (1.5)	25.0 (0.1)	40.0 (1.5)
AIO-HMM	24.6 (1.5)	20.0 (2.0)	30.7 (3.4)

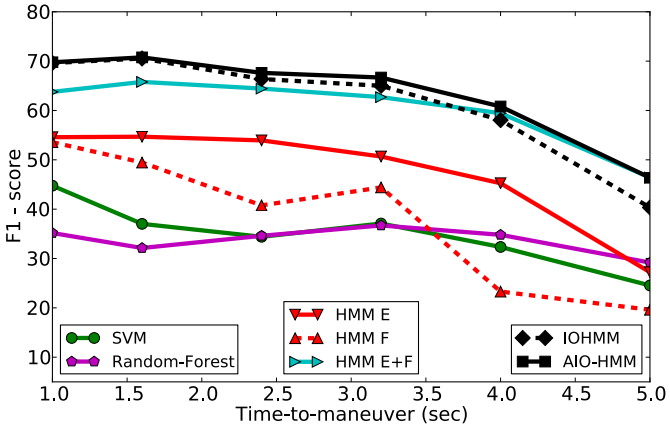


Fig. 8: **Effect of time-to-maneuver.** Plot comparing $F1$-scores when algorithms predict maneuvers at a fixed time-to-maneuver, and shows how the performance changes as we vary the time-to-maneuver.

a maneuver. This results in the low recall of HMM F at 36%, as shown in Table I. AIO-HMM's fpp is 10% less than that of IOHMM and HMM $E+F$ when predicting all the maneuvers.

Importance of inside and outside driving context. An important aspect of anticipation is the joint modeling of the inside and outside driving contexts. HMM F models only the inside driving context, while HMM E models only the outside driving context. As shown in Table I, the precision and recall values of both models is less than HMM $E+F$, which jointly models both the contexts. More specifically, the precision of HMM F on jointly predicting all the maneuvers in 3%, 10%, and 13% less than that of HMM $E+F$, IOHMM, and AIO-HMM, respectively. For HMM E this difference is 4%, 11%, and 14% respectively.

Modeling observation dependencies. AIO-HMM extends IOHMM by modeling the temporal dependencies of events inside the vehicle. Handling this is important because events such as human face movements are smooth and temporally correlated. This results in better performance: on average AIO-HMM precision is 3% higher than IOHMM, as shown in Table I. Also on three maneuvers, AIO-HMM offers a higher precision than IOHMM, as shown in Figure 7.

Effect of time-to-maneuver. In Figure 8 we compare F1-scores of the algorithms when they predict maneuvers at a fixed time-to-maneuver, and show how the performance changes as we vary the time-to-maneuver. As we get closer to the start of the maneuvers the F1-scores of the algorithms increase. As opposed to this setting, in Table I the algorithms predicted maneuvers at the time they were most confident, and achieved higher F1-scores. Under both the fixed and variable time prediction settings, the AIO-HMM consistently performs better than the others.

Anticipation complexity. The AIO-HMM anticipates maneuvers every 0.8 seconds (20 videos frames) using the previous 5 seconds of the driving context. The anticipation complexity mainly comprises of feature extraction and the model inference in equation (7). Fortunately both these steps can be performed as a dynamic program by storing the computation of the most recent anticipation. Therefore, for every anticipation we only have to process the incoming 20 video frames and not complete 5 seconds of the driving context. Furthermore, due to dynamic programming the inference complexity described in equation (7), $\mathcal{O}(K|\mathcal{S}||I|^3)$, no longer depends on K and reduces to $\mathcal{O}(|\mathcal{S}||I|^3)$. In our experiment on average we predict a maneuver under 3.6 milliseconds on a 3.4GHz CPU using MATLAB 2014b on Ubuntu 12.04 operating system.

C. Qualitative discussion

Common Failure Modes. When dealing with natural driving scenarios, wrong anticipations can occur for different reasons. These include failures in the vision pipeline and unmodeled events such as interactions with fellow passengers, overtakes, etc. In 6% of the maneuvers, our tracker failed due to changes in illumination (in supplementary we show some instances of failed tracking). Wrong anticipations are also common when drivers strongly rely upon their recent memory of traffic conditions. In such situations visual cues are partially available in form of eye movements. Similarly, when making turns from turn-only lanes drivers tend not to reveal many visual cues. With rich sensory integration, such as radar for modeling the outside traffic along with reasoning about the traffic rules, we can further improve the performance. Fortunately, the automobile industry has made significant advances in some of these areas [1, 3, 47] where our work can apply. Future work also includes extending our approach to night driving.

Prediction timing. In anticipation there is an inherent decision ambiguity. Once the algorithm is certain about a maneuver above a threshold probability should it predict immediately or should it wait for more information? An example of this ambiguity is in situations where drivers scan the traffic but do not perform a maneuver. In such situations different prediction strategies will result in different performances.

VII. Conclusion

In this paper we considered the problem of anticipating driving maneuvers a few seconds before the driver performs them. Our work enables advanced driver assistance systems (ADAS) to alert drivers before they perform a dangerous maneuver, thereby giving drivers more time to react. In order to anticipate maneuvers, we equipped a car with multiple cameras and a computing device to obtain the multi-modal driving context. We proposed an AIO-HMM model to jointly capture the driver's intention and the contextual information from both inside and outside of the car. The AIO-HMM accurately models both the temporal aspects and generative nature of the problem. It reasons how events outside the vehicle affect the driver's state of mind which results in events inside the vehicle. We extensively evaluated our approach on

1180 miles of driving data and showed improvement over many baseline algorithms. Anticipation using our approach, on average, took only a few milliseconds therefore making it suited for real-time use. We also publicly release the first data set of natural driving with inside and outside videos.

Acknowledgement. We thank Ozan Sener for helpful discussions. This research was funded in part by Microsoft Faculty Fellowship (to Saxena), NSF Career award (to Saxena) and Army Research Office.

REFERENCES

[1] Audi piloted driving. http://www.audi.com/content/com/brand/en/vorsprung_durch_technik/content/2014/10/piloted-driving.html. Accessed: 2014-12-03.

[2] Vehicle safety features. http://consumerreports.org/cro/2012/04/guide-to-safety-features/index.htm. Accessed: 2014-09-30.

[3] Google self driving car. http://en.wikipedia.org/wiki/Google_driverless_car. Accessed: 2014-10-11.

[4] 2012 motor vehicle crashes: overview. *National Highway Traffic Safety Administration, Washington, D.C., Tech. Rep.*, 2013.

[5] Y. Bengio and O. Frasconi. An input output hmm architecture. *NIPS*, 1995.

[6] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun. Learning motion patterns of people for compliant robot motion. *IJRR*, 2005.

[7] H. Berndt, J. Emmert, and K. Dietmayer. Continuous driver intention recognition with hidden markov models. In *IEEE ITSC*, 2008.

[8] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE PAMI*, 23(6), 2001.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3), 1995.

[10] B. Douillard, D. Fox, and F. T. Ramos. A spatio-temporal probabilistic model for multi-sensor object recognition. In *IROS*, 2007.

[11] A. Dragan and S. Srinivasa. Formalizing assistive teleoperation. In *RSS*, 2012.

[12] B. Frohlich, M. Enzweiler, and U. Franke. Will this car change the lane?turn signal recognition in the frequency domain. In *IEEE IVS*, 2014.

[13] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *CVPR*, 2003.

[14] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face detection. *IEEE PAMI*, 29(4), 2007.

[15] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Proc. ICPR*, 2010.

[16] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. Activity forecasting. In *Proc. ECCV*. 2012.

[17] H. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. In *RSS*, 2013.

[18] H. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *ICML*, 2013.

[19] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *RSS*, 2012.

[20] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu. A driver behavior recognition method based on a driver model framework. Technical report, SAE Technical Paper, 2000.

[21] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier. Learning-based approach for online lane change intention prediction. In *IEEE IVS*, 2013.

[22] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[23] C. Laugier, I. E. Paromtchik, M. Perrollaz, MY. Yong, J-D. Yoder, C. Tay, K. Mekhnacha, and A. Negre. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *ITS Magazine, IEEE*, 3(4), 2011.

[24] S. Lefevre, C. Laugier, and J. Ibanez-Guzman. Risk assessment at road intersections: Comparing intention and expectation. In *IEEE IVS*, 2012.

[25] M. Liebner, M. Baumann, F. Klanner, and C. Stiller. Driver intent inference at urban intersections using the intelligent driver model. In *IEEE IVS*, 2012.

[26] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, 1981.

[27] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *IROS*, 2013.

[28] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2), 2004.

[29] L. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR*, 2007.

[30] B. Morris, A. Doshi, and M. Trivedi. Lane change intent prediction for driver assistance: On-road design and evaluation. In *IEEE IVS*, 2011.

[31] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[32] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE PAMI*, 31(4), 2009.

[33] N. Oliver and A. P. Pentland. Graphical models for driver behavior recognition in a smartcar. In *IEEE IVS*, pages 7–12, 2000.

[34] L. Rabiner and B-H. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1), 1986.

[35] M. Rezaei and R. Klette. Look at the driver, look at the road: No distraction! no accident! In *CVPR*, 2014.

[36] T. Rueda-Domingo, P. Lardelli-Claret, J. Luna del Castillo, J. Jimenez-Moleon, M. Garcia-Martin, and A. Bueno-Cavanillas. The influence of passengers on the risk of the driver causing a car collision in spain: Analysis of collisions from 1990 to 1999. *Accident*

Analysis & Prevention, 2004.

[37] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, and K-D. Kuhnert. A lane change detection approach using feature ranking with maximized predictive power. In *IEEE IVS*, 2014.

[38] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[39] A. Criminisi J. Shotton and E. Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, 5(6), 2011.

[40] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[41] A. Tawari, S. Sivaraman, M. Trivedi, T. Shannon, and M. Tippelhofer. Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking. In *IEEE IVS*, 2014.

[42] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[43] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *JMLR*, 1, 2001.

[44] C. Tomasi and T. Kanade. Detection and tracking of point features. *IJCV*, 1991.

[45] M. Trivedi, T. Gandhi, and J. McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *IEEE Trans. on ITS*, 8(1), 2007.

[46] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004.

[47] D. Z. Wang, I. Posner, and P. Newman. Model-Free Detection and Tracking of Dynamic Objects with 2D Lidar. *IJRR*, 2015.

[48] S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *CVPR*, 2006.

[49] Y. Wang and Q. Ji. A dynamic conditional random field model for object segmentation in image sequences. In *CVPR*, 2005.

[50] Z. Wang, K. Mülling, M. Deisenroth, H. Amor, D. Vogt, B. Schölkopf, and J. Peters. Probabilistic movement modeling for intention inference in human-robot interaction. *IJRR*, 2013.

[51] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013.

[52] X. Xiong and F. De la Torre. Supervised descent method for solving nonlinear least squares problems in computer vision. *arXiv preprint arXiv:1405.0601*, 2014.

[53] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Tech. rep., Microsoft Research, 2010.

[54] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.