TOWARDS REAL-TIME RECOGNITION OF DRIVER INTENTIONS

Andrew Liu[1] and Alex Pentland[2]

[1]**Nissan Cambridge Basic Research,** [2]**MIT Media Laboratory**

Cambridge, MA USA

Keywords: Hidden Markov Models, Intelligent Control, Behavioral Science

ABSTRACT

Knowledge of an automobile driver's intended actions (e.g., to turn, change lanes, etc.) could facilitate the integration of intelligent vehicle systems with the driver. The actions can be inferred from the driver's control actions as she prepares to execute an action. Actions are modeled as a sequence of internal mental states, each with a characteristic pattern of driver control behavior. By observing the temporal pattern of the drivers' control behavior and comparing it to the action models, we can determine which action the drivers are beginning to execute.

INTRODUCTION

With the arrival of fully automated cars still in the distant future, "augmented control" (AC) systems [1] that enhance the driver's operation of the vehicle are becoming a focus of development. Familiar examples of primitive augmented systems include adaptive cruise control systems that help the driver maintain inter-vehicle distance and speed, and steering systems that help the driver maintain lane position. Future systems might dynamically adjust low-level vehicle operations, such as the shift timing in the transmission, with knowledge of driver intentions and thus improve the safety of actions such as passing. In the future, there will also be more information available to the driver (e.g., traffic and navigation information systems). The addition of information displays could increase the attentional load on the driver and potentially make driving more difficult. Without a proper understanding of the driver's mental state, it is conceivable that the concurrent activation of many such systems could be very distracting or even overwhelming. Therefore, the future smart car must serve the role of the driver's "valet", generally keeping out of sight and mind, but providing the necessary services when needed.

The difficulty, of course, is that we generally do not understand humans well enough to construct a useful mathematical model of their driving behavior. Early attempts to devise AC systems adopted the simplest possible model of the human, such as a noise source or filter. While this makes it relatively easy to obtain a mathematical specification of the vehicle control system, including the human, such simple models make it impossible to build a system that takes real advantage of the human's abilities.

Previous research

A thought experiment suggests that the driving situation may be sufficiently constrained that we can hope to infer drivers' intentions from their motor behavior. Imagine being a passenger sitting in a car and looking at the driver; do you think you could guess when the driver was preparing to turn left? to stop? to pass? Most people, after some reflection, answer yes to these questions; that is, they believe that drivers' intended actions can be guessed from observation of their movements. This belief is supported by experimental studies that suggest that some aspects of driving behavior (e.g., posture changes before braking [2], eye motion before turning at moderate to high speeds [3]) are sufficiently stereotyped to allow an action to be recognized well before its actual execution.

In recent years, research has focused on two areas in which to make the car more "intelligent": determining the intentions of the driver and determining the state of the environment in which the driver is operating. In either case, the central question has been whether driver behavior is a reliable indicator of either driver intentions or the external state.

Takahashi [4] has studied the driver's intention to decelerate while coasting on a downhill grade. This system predicts when the driver intends to decelerate and automatically shifts to a higher gear for greater engine braking. A quantitative submodel expresses the intention to decelerate as a function of measurable vehicle parameters. The model parameters were derived from case data in which many different drivers evaluated combinations of shift timing and coasting speed.

Detecting the nature of the vehicle/driver's surrounding environment has been approached in a similar manner, that is, by using driver behavior as an indicator. Both Takahashi [5] and Qiao [6] have used fuzzy techniques to predict the type of surrounding terrain (i.e., mountain roads versus flat roads) based on vehicle parameters such as speed and accelerator position.

HIDDEN MARKOV DYNAMIC MODELS

Our approach is to modeling human behavior is to consider the human as a Markov device with a (possibly large) number of internal 'mental' states, each with its own particular control behavior, and interstate transition probabilities (e.g., in a car the states might be centering in the lane, matching speed with a leading car, changing lanes, and centering in the new lane....which together constitute a lane change) A simple example of this type of human model would be a bank of standard quadratic controllers, each using different dynamics and measurements, together with a network of probabilistic transitions between them.

Single Dynamic Models

Consider the dynamic processes

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \Delta t) + \xi(t) \qquad (1)$$

where the function $\mathbf{f}$ models the dynamic evolution of state vector $\mathbf{X}_k$ at time k, and let us define an observation process

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{X}_k, \Delta t) + \eta(t) \qquad (2)$$

where sensor observations $\mathbf{Y}$ are a function $\mathbf{h}$ of the state vector and time. Both ξ and η are white noise processes having known spectral density matrices.

Using Kalman's result, we can then obtain the optimal linear estimate $\hat{\mathbf{X}}_k$ of the state vector $\mathbf{X}_k$ by use of the following *Kalman filter*:

$$\hat{\mathbf{X}}_k = \mathbf{X}_k^* + \mathbf{K}_k(\mathbf{Y}_k - \mathbf{h}(\mathbf{X}_k^*, t)) \qquad (3)$$

where $\mathbf{K}_k$ is the Kalman gain matrix [11]. At each time step k, the filter algorithm uses a state prediction $\mathbf{X}_k^*$, an error covariance matrix prediction $\mathbf{P}_k^*$, and a sensor measurement $\mathbf{Y}_k$ to determine an optimal linear state estimate $\hat{\mathbf{X}}_k$, error covariance matrix estimate $\hat{\mathbf{P}}_k$, and predictions $\mathbf{X}_{k+1}^*$, $\mathbf{P}_{k+1}^*$ for the next time step.

Given the state vector $\mathbf{X}_k$ at time k we can predict the measurements at time $k + \Delta t$ by

$$\mathbf{Y}_{k+\Delta t} = \mathbf{h}(\mathbf{X}_k, \Delta t) \qquad (4)$$

and the predicted state vector at time $k + \Delta t$ is given by

$$\hat{\mathbf{X}}_{k+\Delta t} = \mathbf{X}_k^* + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \qquad (5)$$

Multiple Dynamic Models

Human behavior, in all but the simplest tasks, is not as simple as a single dynamic model. The next most complex model of human behavior is to have *several* alternative models of the person's dynamics, one for each class of response. Then at each instant we can make observations of the person's state, decide which model applies, and then make our response based on that model. This is known as the *multiple model* or *generalized likelihood* approach, and produces a generalized maximum likelihood estimate of the current and future values of the state variables [12].

Mathematically, this is accomplished by setting up a set of states S, each associated with a Kalman filter and a particular dynamic model:

$$\hat{\mathbf{X}}_k^{(i)} = \mathbf{X}_k^{*(i)} + \mathbf{K}_k^{(i)}(\mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_k^{*(i)}, t)) \qquad (6)$$

where the superscript (i) denotes the i^{th} Kalman filter. The *measurement innovations process* for the i^{th} model (and associated Kalman filter) is then

$$\Gamma_k^{(i)} = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_k^{*(i)}, t) \tag{7}$$

The measurement innovations process is zero-mean with covariance $\mathcal{R}$.

The i^{th} measurement innovations process is, intuitively, the part of the observation data that is unexplained by the i^{th} model. The model that explains the largest portion of the observations is, of course, the model most likely to be correct. Thus, at each time step, we calculate the probability $Pr^{(i)}$ of the m-dimensional observations $\mathbf{Y}_k$ given the i^{th} model's dynamics,

$$Pr^{(i)}(\mathbf{Y}_k) = \frac{\exp\left(-\frac{1}{2}\Gamma_k^{(i)T}\mathcal{R}^{-1}\Gamma_k^{(i)}\right)}{(2\pi)^{m/2}\mathrm{Det}(\mathcal{R})^{1/2}} \tag{8}$$

and choose the model with the largest probability. This model is then used to estimate the current value of the state variables, to predict their future values, and to choose among alternative responses.

Hidden Markov Dynamic Models

In the above multiple dynamic model, all the processes have a fixed likelihood at each time step. However, this is uncharacteristic of most situations, where there is a fixed sequence of internal states each with its own dynamics. Consider driving through a curve; the driver may be modeled as having transitioned through a series of states $\lambda = (s_1, s_2, ...s_k)$, $s_i \epsilon S$, for instance, entering a curve, in the curve, and exiting a curve. Transitions between these states happened only in the order indicated, with a final transition from other to entering the curve.

Thus in considering state transitions among a set of dynamic models we should make use of our current estimate of the driver's internal state. We can accomplish this fairly generally by considering the Markov probability structure of the transitions between the different states. The input to decide the person's current internal state (e.g., which dynamic model currently applies) will be the measurement innovations process as above, but instead of using this directly in Equation 8 we will instead also consider the Markov inter-state transition probabilities.

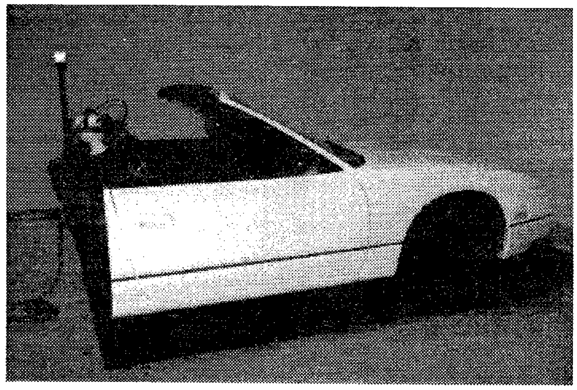

Figure 1: The Nissan Cambridge Basic Research 240SX simulator.

We call this type of model a *hidden Markov dynamic model* (HMDM) [1]. Using this approach one can recognize which action is occuring by comparing the observed pattern of driver behavior to each of several hidden Markov dynamic models (one for each action of interest). We use the well-known Viterbi algorithm [7] to match the innovations processes of the various dynamic models (Equation 7) with the action pattern proscribed by each of the HMDMs. This produces an estimate of which action is *most likely* given the observed pattern of driver behavior, and can be accomplished in real-time on current microprocessors.

DEVELOPMENT OF HMDMS FOR DRIVING

The goal of our research is to statistically characterize the sequence of steps within a driving action, and to use the first few preparatory steps to identify the action being initiated. We report our progress developing HMDMs for a range of common driving tasks as well as preliminary results of using the HMDMs in a real-time task recognition system.

Driving Simulation

The data collection for HMDM training was performed on the Nissan Cambridge Basic Research driving simulator. The simulator consists of the front half of a Nissan 240SX convertible and a 60 deg (horizontal) by 40 deg (vertical) image projected onto the wall facing the driver (Figure 1). The

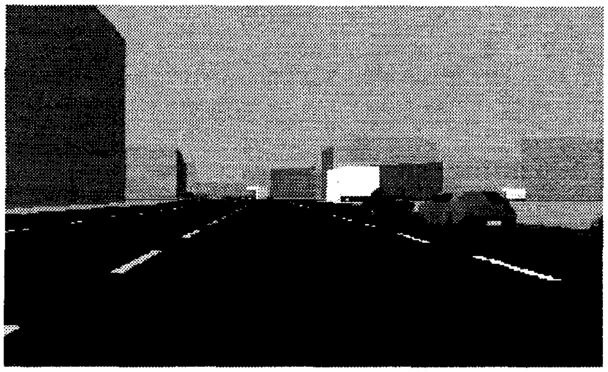

Figure 2: Scene from the simulated world.

240SX is instrumented to record driver control input such as steering wheel angle and parameters of the dynamic car model such as acceleration.

Using the simulator, eight adult male subjects drove through an extensive city-like virtual environment illustrated in Figure 2. This world contains a large number of buildings, many roads with standard markings, and other moving cars. Each subject completed three trials through the virtual world; each trial lasted approximately five minutes, during which time the driver's control of steering angle and steering velocity, car velocity and acceleration were recorded at approximately 0.1 second intervals.

From time to time during this drive text commands were presented on-screen, whereupon the subjects had to assess the surrounding situation, formulate a plan to carry out the command, and then execute the command. The total time needed to complete each command varied from 5 to 10 seconds, depending upon the complexity of both the action and the surrounding situation.

The commands included: (1) stop at next intersection, (2) turn left at next intersection, (3) turn right at next intersection, (4) change lanes, (5) pass the car in front of you, (6) follow the car in front of you. The pass and follow commands appeared only once per trial, while the change lane and stop commands appeared twice. The two turn commands were repeated three times at different locations, each time with different road, building, and traffic conditions.

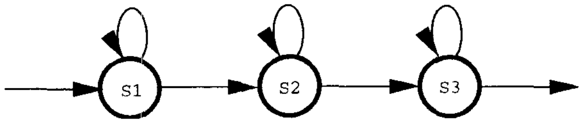

Figure 3: A driving action model with three intenal states (S1, S2, S3).

HMDM parameter estimation

The parameters of each Markov chain λ were estimated iteratively to maximize $Pr(\mathbf{Y}|\lambda)$ where the observation sequence $\mathbf{Y}$ consisted of the steering angle, steering speed, and acceleration data recorded as subjects carried out these commands. The initial models were identical three-state models (Figure 3) with equal transition probabilities to each of the following states. Using the tools provided by Entropic's HTK computer software, the parameter estimation was completed in two steps. A more formal discussion of these procedures can be found in [9].

First, an initial set of parameter values was obtained from the training data with a uniform segmentation of the training data. This was followed by repeated use of Viterbi alignment to re-segment the training data. The Viterbi algorithm can be used to recover the most likely state sequence which in turn determines the new segmentation of the observation sequence. The parameters of each state are re-estimated and the procedure is repeated until the likelihood of the training data no longer increases. A Baum-Welch re-estimation of individual HMDMs is performed to further refine the parameters.

Next, an embedded training version of the Baum-Welch algorithm is performed. In this case, the model parameters were simultaneously re-estimated from labelled but unsegmented training data (i.e., the sequence in which the training examples occurs is used, but the exact boundaries of each example are not needed). This procedure further tunes the models for continuous real-time recognition.

Evaluating recognition performance

To assess the classification accuracy of these models we combined them with the Viterbi recognition algorithm, and examined the stream of drivers' steering and acceleration innovations in order to detect and classify the driver's actions. We then examined the computer's classifications immediately after each command, and recorded whether or not the computer had correctly labeled the action.

To obtain unbiased estimates of recognition performance, we employed the "leaving one out" method. In this method HMDMs are trained on seven subjects and then tested on the eighth subject. This is then repeated eight times, each time leaving out a different one of the eight subjects, and the recognition statistics are averaged.

Recognition results were tabulated at one, two, and three seconds after the presentation of a command to the subject. Note that the minimum response time to a command is approximately 0.5 seconds, so that the one second point is at most one-half second after the beginning of the driver's action. At one second after the command presentation ($\approx$0.5 seconds after the beginning of the action) mean recognition accuracy was 88.3% $\pm$ 4.4% (SD). At two and three seconds ($\approx$1.5 and $\approx$2.5 seconds after the beginning of the action, respectively), mean recognition accuracy was 89.4% $\pm$ 4.3% and 87.5% $\pm$ 2.8%. The chance of randomly guessing the correct task is only 17% (1 in 6). These results demonstrate that many types of driving behavior are sufficiently stereotyped that they are reliably recognizable from observation of the driver's initial movements.

To assess whether our sample was sufficiently large to adequately encompass the range of variation in the driving styles of our subjects, we compared these previous results to the case in which we train on all subjects and then test the training data. In this case, the recognition accuracies were 91.0%, 88.6%, and 88.2%, indicating that we have a sufficiently large sample.

It should be noted that the performance of the current models is slightly reduced compared to the results reported in [1]. The models in [1] were trained on four parameter observation vectors (vehicle velocity was the fourth parameter) that were also subsampled to 20Hz. Due to hardware limitations in our real-time implementation, it was not possible to supply 20Hz data to the recognizer, so the models in the present study were trained from 10Hz data. The velocity parameter was eliminated to make recognition of the actions velocity-independent.

REAL-TIME RECOGNITION

We have implemented a version of a real-time task recognition system on the 240SX simulator adapting the real-time recognition capabilities of the HTK software to work with non-speech signal sources. The simulator acts as the signal source and sends a 3-parameter observation vector (acceleration, steering angle, steering wheel velocity) 10 times per second via an RPC connection to another computer running the HTK Viterbi recognizer. The HTK host reads the data and performs continuous task recognition on the signal. A frame-by-frame record of the recognizer's classification (the action HMDM with the highest likelihood of generating the observed sequence) is compared to the log file produced by the simulator to assess the recognition performance. In an actual system, the recognizer would send the current classification back to the vehicle, to initiate the appropriate adjustment or display the needed information.

In a preliminary test of the implementation of the real-time system, three subjects drove through the virtual city following the on-screen commands as they appeared. One of the three subjects was also included in the training pool. We examined the number of correct guesses in 0.5 second windows up to three seconds after the command presentation and found that recognition performance varied widely for the tasks. For left turns, the recognition rate was fairly constant between 50-60%. One second after the command, passing recognition was up to 60-70% and after two seconds, right turn recognition was over 60%. Both stopping and following recognition only reached 40-50% after 2.5 seconds. Interestingly, lane change recognition started around 50% and decreased with time. This may reflect the faster completion time of lane changes compared to the other tasks.

WORK IN PROGRESS

The current procedure of giving on-screen commands to the driver is obviously unrealistic. In addition, the point at which the driver decides to begin execution is an indeterminate time after the command presentation. The resulting variation in the segmentation of the training set reduces the recognition performance of the HMDMs. Under normal circumstances, drivers choose when to execute a task, albeit constrained to varying degrees by the surrounding traffic, road signs, etc.

Therefore, we are testing a new protocol for collecting training data. The subjects now drive through a more extensive city-like environment with the freedom to choose when and where to execute the maneuvers. They verbally report the initiation and completion of a maneuver (e.g., they will say "I am turning right") while the experimenter records the time of the report in the log file produced by the simulator. This will hopefully provide a more consistent segmentation of the training data. We are also recording the state of the surrounding traffic since this is information which the driver might use in deciding a course of action.

Using this data set, it will also be possible to observe what the drivers are doing just *before* initiating a maneuver. If they demonstrate consistent behavior before their execution of the maneuver, then it may be possible to create a model of the driver's preparatory actions. In this case, it will be possible to *predict* the driver's intention rather than simply detecting the action after it has begun.

ACKNOWLEDGMENTS

We thank Nuria Oliver and Janis Stipins for their work in developing software to enable real-time recognition, Marcos Fernandez for the development of the traffic simulation, and Isaac Murakami for assistance with the data analysis.

REFERENCES

[1] A. Pentland and A. Liu. Towards Augmented Control Systems. *Proc. of the 1995 IEEE Intelligent Vehicles Symposium*, pp. 350-355, Detroit, MI, Sept, 1995.

[2] M. Helander. Applicability of Drivers' Electrodermal Response to the Design of the Traffic Environment, *J. Applied Psychology*, Vol. 63, No. 4, pp. 481-488, 1978.

[3] M. Land and D.N. Lee. Where we look when we steer. *Nature*, Vol. 369, pp. 742-744, 1994.

[4] H. Takahashi and K. Kuroda. A Study on Automated Shifting and Shift Timing Using a Driver's Mental Model, *Proc. of the 1996 IEEE Intelligent Vehicles Symposium 96*, pp. 300-305, Tokyo, Japan, Sept. 19-20, 1996.

[5] H. Takahashi. Human Sensor System that Predicts Driving Environment Using Driver's Fuzzy Operational Data. *Japanese J. of Fuzzy Theory and Systems*, Vol. 4, No. 6, pp. 709-734, 1992.

[6] L. Qiao, M. Sato, and H. Takeda. Learning Algorithm of Environmental Recognition in Driving Vehicle. *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 25, No. 6, pp. 917-925, 1995.

[7] L. Rabiner and B. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, p. 4–16, Jan. 1986.

[8] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh Univ. Press, Edinburgh, 1990.

[9] S. Young, J. Jansen, J. Odell, D. Ollasen, and P. Woodland. *HTK: Hidden Markov Model Toolkit V2.0*. Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, 1996.

[10] L. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes. *Inequalities*, Vol. 3, pp. 1–8, 1972.

[11] R.E. Kalman and R.S. Bucy. New Results in Linear Filtering and Prediction Theory. In *Transaction ASME (Journal of basic engineering)*, 83D, 95-108, 1961.

[12] A.S. Willsky. Detection of Abrupt Changes in Dynamic Systems. In M. Basseville and A. Benveniste, (Ed.). *Detection of Abrupt Changes in Signals and Dynamical Systems*, Lecture Notes in Control and Information Sciences, No. 77, pp. 27-49, Springer-Verlag, 1986.