UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title
A soft barrier model for predicting human visuomotor behavior in a driving task

Permalink
https://escholarship.org/uc/item/3v86d63c

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 35(35)

ISSN
1069-7977

Authors
Johnson, Leif
Sullivan, Brian
Hayhoe, Mary
et al.

Publication Date
2013

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

A soft barrier model for predicting human visuomotor behavior in a driving task

Leif Johnson (leif@cs.utexas.edu)
Department of Computer Science, University of Texas at Austin, USA

Brian Sullivan (brians@ski.org)
Smith-Kettlewell Eye Research Institute, San Francisco, CA, USA

Mary Hayhoe (mary@mail.cps.utexas.edu)
Department of Psychology, University of Texas at Austin, USA

Dana Ballard (dana@cs.utexas.edu)
Department of Computer Science, University of Texas at Austin, USA

Abstract

We present a task-based model of human gaze allocation in a driving environment. When engaged in natural tasks, gaze is predominantly directed towards task relevant objects. In particular in a multi-task scenario such as driving, human drivers must access multiple perceptual cues that can be used for effective control. Our model uses visual task modules that require multiple independent sources of information for control, analogous to human foveation on different task-relevant objects. Building on the framework described by Sprague and Ballard (2003), we use a modular structure to feed information to a set of PID controllers that drive a simulated car and introduce a barrier model for gaze selection. The softmax barrier model uses performance thresholds to represent task importance across modules and allows noise to be added to any module to represent task uncertainty. Results from the model compare favorably with human gaze data gathered from subjects driving in a virtual environment.

Keywords: Visual attention; eye movements.

Introduction

Humans routinely interact with complex, noisy, dynamic environments to accomplish tasks in the world. For example, while driving a car, a person navigates to a desired destination (e.g., grocery store) while paying attention to different types of objects in the environment (pedestrians, vehicles, etc.) and obeying traffic laws (speed limit, stop signs, etc.). Humans are able to balance competing task demands while simultaneously gathering information from the world through a foveated visual system, which must be actively moved to different targets to obtain high-resolution imagery.

During the deployment of attention, in particular overt eye movements towards an object, humans are sensitive to bottom-up salience (color, motion, etc.) as well as top-down task priority and the rewards associated with a task (Knudsen, 2007; Wolfe, Butcher, Lee, & Hyle, 2003). In particular when engaged in "natural" tasks, eye movements are largely directed towards task relevant objects (Hayhoe & Ballard, 2005; Land & Hayhoe, 2001). Typically in natural environments, there are multiple task relevant objects spread over space and time that require active visual strategies to properly gather information. While human vision research has often focused on models of visual saliency, i.e., a stimulus based controller of attention (Bruce & Tsotsos, 2009; Itti & Koch, 2001; Zhang, Tong, Marks, Shan, & Cottrell, 2008), such models are inappropriate to address task-based behavior because they do not incorporate information about the state of the agent whose vision is being modeled. An alternative approach is to consider vision as part of a control process where information from the senses is used to guide motor behavior (Butko & Movellan, 2010; Nunez-Varela, Ravindran, & Wyatt, 2012; Senders, 1980; Sprague & Ballard, 2003; Sullivan, Johnson, Ballard, & Hayhoe, 2011). Both stimulus and task-based approaches have led to a variety of formulations concerning how eye movements should be selected, e.g., using energy models, information theoretic measures, or measures of reward and uncertainty. In the present work, we focus on how selection of eye movement targets may be controlled in part by task related uncertainty and reward.

We present a model of visual processing and control that simultaneously takes into account the reward and uncertainty in multiple tasks associated with a dynamic, noisy driving environment. The model successfully accounts for variations in gaze deployment seen in humans driving in a virtual reality driving environment. Additionally, we discuss future research allowed by inversion of the soft barrier model. Inversion allows human data to be mapped into parameters in the model space so that it can be understood and compared quantitatively within the model framework.

Model

The model proposed in this paper follows the modular architecture of Sprague and Ballard (2003) by factoring complex behaviors like driving into a set of simple control modules that each focus on a well-defined task—for example, a module to follow the road and another to avoid oncoming cars. Intuitively, a module is an abstract black-box controller that can be used alone to guide an agent through a single task. More interestingly, modules can be used together dynamically to engage in multiple ongoing behaviors. While the human visual system is highly parallel, processing and attentional focus are largely biased towards the fovea, meaning humans typically get information in a serial fashion by foveating different objects over time. In our model, multiple task modules run concurrently; however, to incorporate the foveation constraint, only one module at a time actively gains new percep-

tual information.

At a high level, modules are responsible for gathering and updating information about specific aspects of the state of the world, and for using that information to generate control signals for the agent. A central component of the model is that it requires a usable control policy even in the absence of updating its state information. Human short term memory decays with time, so to simulate this we allow the state information upon which the actions are computed to be corrupted by noise. We incorporate these into our model using simple scalar values for each module. Formally, we define a module as a tuple $M = (\mathcal{S}, \mathcal{A}, \pi, \mathbf{s}^*, \rho, \varepsilon)$, where:

- $\mathcal{S} = \{s_1, \ldots, s_n\}$ is a set of the n state variables that are relevant to the module,
- $\mathcal{A} = \{a_1, \ldots, a_k\}$ is a set of the k action variables that are relevant to the module,
- $\pi : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is a control policy that maps state values onto actions,
- $\mathbf{s}^* \in \mathbb{R}^n$ is a vector of target state values,
- ρ is a scalar uncertainty threshold value for the module, and
- ε is a scalar noise value for the module.

The first three elements of M are common to typical Markov decision process (MDP) scenarios. The state space, spanned by elements of $\mathcal{S}$, represents all possible combinations of world state that are relevant for the task. The action space, spanned by elements of $\mathcal{A}$, describes all possible actions that the agent can take. The control policy maps states to actions; an optimal policy maps states to the best action for each state. The fourth element of the tuple, $\mathbf{s}^*$, is a vector of target values for each state variable. These target values are used in place of the more traditional formulation of scalar reward; this is discussed in further detail below. Finally, each module incorporates explicit values for both task priority $1/\rho$ and task uncertainty ε, which are also explained below.

A learning agent is equipped with N individual modules $M^{(1)}, \ldots, M^{(N)}$ that each specialize in one task and can be used in conjunction to control behavior in the world. To simplify the control problem, in our model all modules share a common set of action variables. In the driving environment described in this paper, there are two action variables: one represents changes in the vehicle's speed and another represents changes in the vehicle's heading.

State estimates

Each module depends on a set of world-state variables that are relevant to the module's specific task. When driving, relevant state variables for a car-following task, for example, could include the agent-centric distance and angle to the leader car, the speed and heading relative to the leader car, etc. Relevant state variables for a target-speed task might be as simple as monitoring the absolute speed of the agent.

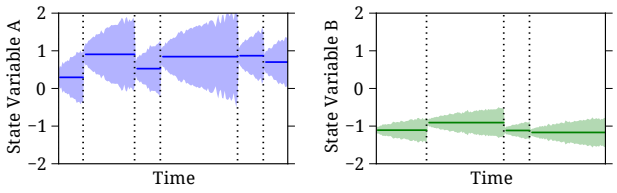

Figure 1: Evolution of state variable and uncertainty information for two single-variable modules. On the left, the solid blue lines represent the observed values of the state variable A over time, while the shaded blue regions represent the region in which the true value of the state variable is likely to occur. On the right, the observed values and uncertainty regions are shown in green for a different state variable, B. Vertical dashes in each plot indicate times where the state estimates are updated with a new observation of the true value; these updates also reduce the magnitudes of the uncertainties in each estimate towards zero. The ε parameter in this scenario is greater for the module tracking variable A than for the module tracking variable B.

In MDP scenarios, agents are assumed to have constant access to accurate state variable information. Humans, on the other hand, have a foveated visual system that often requires active serial collection of updated state information. We assume that this serial process requires that when one visual task is accessing new information all other tasks must rely on noisy memory estimates.

To incorporate this state uncertainty into the model, each module $M^{(i)}$ maintains an explicit *estimate* of the current value of each of its state variables, $\hat{\mathbf{s}}^{(i)}(t)$. (We will henceforth omit the module superscript except to resolve ambiguities.) This estimate could be designed to incorporate many sorts of prior information about the evolution of the world, but the model in its current state simply treats state estimates as samples drawn IID from a spherical normal distribution

$$\hat{\mathbf{s}}(t) \sim \mathcal{N}\left(\mu(t), \sigma^2(t) I\right)$$

where $\mu(t) = [\mu_1(t) \ldots \mu_n(t)]^T$ is a vector of the most recently observed state values, and $\sigma(t)$ is the standard deviation for the state variable estimates in the module. Figure 1 shows the state updates over time for a simple, hypothetical system containing two modules, each tracking one state variable.

Uncertainty propagation Uncertainty propagates over time within each module by maintaining a small set of J "uncertainty particles" $\mathcal{E} = \{\beta_1(t), \ldots, \beta_J(t)\}$. Each particle represents one potential path of deviation that the true state value might have taken from the last-observed state value. At every time step in the simulation, all uncertainty particles are displaced randomly by a step drawn from $\mathcal{N}(0, \varepsilon)$, thus defining a random walk for each particle. The root-mean-square value of the uncertainty particles is then used to define the standard

deviation

$$\sigma(t) = \sqrt{\frac{1}{J}\sum_{j=1}^{J}\beta_j^2(t)}$$

of state estimates for this module. Periodically, a module will be updated with accurate state information from the world (described below); when this happens, the magnitude of each uncertainty particle for the module is reduced according to $\beta_j(t+1) = (1-\alpha)\beta_j(t)$. After an informal parameter search, we set $\alpha = 0.7$ for all modules; with $\alpha = 1$, the model tends to produce many short updates because uncertainty is instantly reduced to 0, but with $\alpha < 1$ the uncertainty increase due to noise competes with the uncertainty reduction from the updated state information. Figure 2 shows the uncertainties over time for the hypothetical two-module system shown in Figure 1.

The state estimation approach described here can be seen as a sort of particle filter (e.g., Arulampalam et al., 2002), using an uninformed proposal distribution and equal weights for all particles. Interestingly, the behavior of the simulation was largely unaffected by the choice of J; for our simulation, we used $J = 10$.

Control policy

Each module relies on a policy to determine which action to take when the world is in a particular state. There are multiple ways an MDP may be solved for a control policy, e.g. in reinforcement learning a Q–table can be learned, which explicitly represents the expected future reward for each possible state and action combination; the policy is then given by a simple maximum over available actions for each state.

For a task like driving, however, continuous variables are the most natural representation of state (distance to another car, current speed, etc.) and action (change in speed, change in steering) variables. Although MDP algorithms can converge on policies for tasks in continuous spaces, for many real-world tasks the resulting policies can be more easily modeled using a simple parametric function. In addition, many algorithms for solving MDPs require significant training time to arrive at these regularly-shaped policies. The model described here instead uses a continuous proportional-integral-derivative (PID) control strategy.

PID controllers A PID controller $C(e)$ is a feedback control functional that maps state errors $e(t)$ onto control signals $u(t)$. Formally,

$$C(e) = K_P\, e(t) + K_I \int_0^t e(v)\, dv + K_D\, \dot{e}(t)$$

where K_P, K_I, and K_D are parameters that affect the convergence speed and stability of the PID controller output when encountering a step change in error. In our model, these parameters are tuned manually for each module in isolation (O'Dwyer, 2006) to produce qualitatively appropriate driving behavior.

Each module in the model uses one PID controller for each state variable. Given estimates $\hat{\mathbf{s}}(t)$ of the current values of each variable and a vector of target state values $\mathbf{s}^*$, the control policy becomes

$$\pi(\hat{\mathbf{s}}(t)) = U\,[\, C_1(\hat{s}_1(t) - s_1^*) \ldots C_n(\hat{s}_n(t) - s_n^*)\,]^T$$

where U is a $k \times n$ mixing matrix that combines control policy recommendations from each PID controller into a final control output for each action variable. Note that, in this model, the control policy π does not have access to the true state values $\mathbf{s}(t)$, but rather to the module's estimates of those state values $\hat{\mathbf{s}}(t)$.

The composition of U is determined by the needs of the modeling task. For the driving task, for example, each module generally has one state variable corresponding to a desired distance, and another corresponding to a desired heading. For this case, U is set to the 2×2 identity matrix, since the PID controller that is monitoring a distance variable provides a natural control signal for vehicle speed, and the PID controller that monitors an angle variable provides a control signal for the vehicle heading. The exception to this is the module focusing on maintaining a target speed; this module only monitors current speed in the world, so it always provides a zero-output control value for the change-of-heading action variable.

Priority

Modules can be prioritized by increasing their importance relative to one another, to allow modular agents to perform one task (for example, following a leader car) in preference to another (like achieving a target speed). In a traditional MDP scenario, this is modeled by controlling the ratio of reward values between two subsets of world states. In the present model, module priority is manipulated through the ρ parameter: as ρ increases, the module's relative priority decreases.

This relative priority value is incorporated into the model as a soft bound on the diffusion of uncertainty for each module. The specifics of this integration of uncertainty and priority are described in more detail next, as part of the perceptual arbitration process.

Simulation

In simulation, an agent is placed in a two–dimensional virtual driving world. The world contains a single road with multiple lanes. Several non-agent cars are placed on the lanes at random locations, and one of the non-agent cars is designated as the leader car.

The basic simulation loop updates the state of the world at a fixed frequency f_s (set to 60Hz to match experimental conditions from (Sullivan, Johnson, Rothkopf, Ballard, & Hayhoe, 2012)) according to an elementary physics simulation. At each time step, each car in the world moves ahead proportionally to its speed, in a direction given by its heading. For the non-agent cars, the simulator constrains these speed and heading values so that the cars always follow the lanes in the world.

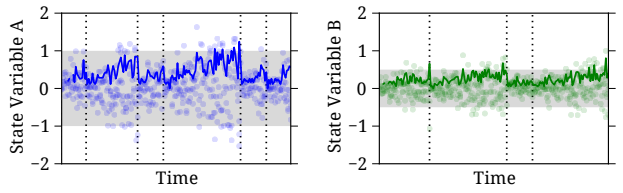

Figure 2: Example random walks for uncertainty particles in the two modules shown in Figure 1. The individual particles are shown as small dots in each plot; their RMS uncertainty value $\sigma(t)$ is shown with a solid blue (left) and green (right) line. Vertical dotted lines indicate time steps when each module received an update; these updates reduce the magnitudes of the uncertainty particles towards 0. The uncertainty threshold ρ for each module is indicated by the shaded gray region in the center of the plot; in this example, the module tracking state variable B (right) has higher priority (lower uncertainty threshold ρ) than the module tracking variable A. However, because the module on the left has a higher noise parameter ε, it receives more updates than the module on the right in the same duration of time.

After moving all vehicles in the world, the simulation additionally requests a control update from the learning agent, which changes the heading and speed of the agent before the next frame begins. Every time the simulator requests a control update for the learning agent, the modules are also updated by displacing their uncertainty particles according to each module's noise parameter ε.

Perceptual Arbitration

If the simulation only performed the steps above, the agent's performance would become increasingly erratic over time, because the uncertainty particles would drift further away from zero. The resulting erroneous state value estimates would produce poor PID controller outputs, and the resulting actions chosen by the agent would further compound the uncertainty in the state estimates. In a human driver, this behavior would be analogous to taking one look at the world when getting into the car, and then driving with a blindfold thereafter.

Clearly this is not what humans tend to do when driving. Instead, people continually and regularly reposition their gaze toward objects in the environment as the driving task progresses. The final step in our model is to incorpate a scheduler to arbitrate between task modules, such that updated sensory information is delivered to the PID controllers dependent on task uncertainty and priority. Like Sprague and Ballard (2003), we hypothesize that this repositioning serves to reduce uncertainty about the state of relevant variables in the world—distance to a leader car, current speed, etc. To capture this behavior, the simulator periodically selects a module for receiving updated state information through a perceptual arbitration mechanism. This selection process happens at a constant frequency f_p (set to 3Hz for the results reported here to approximate the frequency of human gaze behavior).

The perceptual arbitration process incorporates priority and uncertainty in the following way. We first define, for each module $M^{(i)}$, a weighted uncertainty at time t that incorporates both the RMS uncertainty and the scalar priority of the module:

$$\zeta^{(i)}(t) = \sigma^{(i)}(t) - \rho^{(i)}.$$

We also define a global variable $\phi(t)$ to represent the index of the module that gets updated at time t. Then the soft barrier model defines the probability that module $M^{(i)}$ is selected for update at time t as a Boltzmann distribution over each of the priority-weighted module uncertainties:

$$p(\phi(t) = i | \zeta^{(1)}(t), \ldots, \zeta^{(N)}(t)) = \frac{\exp\left(\zeta^{(i)}(t)\right)}{\sum_{j=1}^{N} \exp\left(\zeta^{(j)}(t)\right)}$$

Intuitively, if the uncertainty in $M^{(i)}$ is currently above the threshold for that module—that is, if $\sigma^{(i)}(t) > \rho^{(i)}$—then $M^{(i)}$ is much more likely to be selected for update than another module, especially if none of the other modules have uncertainties exceeding their thresholds. However, the softmax selection process allows for nondeterminism: even if $\zeta^{(i)}(t) > \zeta^{(j)}(t)$ for $j \neq i$, there is some nonzero probability that i will not be selected for update at time t. Finally, because module updates are always selected at frequency f_p by sampling from the above distribution at the appropriate time, a module might be selected for update even if none of the agent's task modules have exceeded their uncertainty boundary (i.e., if $\zeta^{(i)} < 0$ for all i).

Although inspired by diffusion models of decision making, this model contrasts somewhat with traditional models. Many diffusion models with "hard" bounds were developed for forced-choice, two-alternative tasks (e.g., Carpenter & Williams, 1995); our model, in comparison, is designed to incorporate a wider variety of tasks. The "soft" barrier, driven at a fixed frequency, can incorporate more than two choices into the model simultaneously, while accounting for biologically realistic delays in planning and executing saccades.

As described above, and illustrated in Figures 1 and 2, when a module is selected for update, it is provided with the true state of each world state variable in $\mathcal{S}^{(i)}$, and each of its uncertainty particles β_j is reduced towards zero for every simulation frame until another module is selected for update.

Simulation results

We implemented the model described above[1] and ran several simulations to assess its qualitative behavior. Our simulated driving environment was identical in layout to the virtual environment used by Sullivan et al. (2012), so that we could directly compare our results to human performance. Our implementation consisted of three modules: a "speed" module

[1] http://github.com/lmjohns3/driving-simulator

$M^{(s)}$ that attempted to drive at a particular target speed; a "follow" module $M^{(f)}$ that attempted to follow a lead car, and a "lane" module $M^{(l)}$ that attempted to steer so as to follow the nearest lane on the road. All cars in the simulation drove in a simulated 2–dimensional world, described above. Each time gaze was allocated to a new module, we recorded the module that received the gaze, as well as several behavioral measurements (e.g., distance to leader car, current speed, etc.) to verify that the agent was driving appropriately.

Categorizing looks

The gaze selection process in our model is Markovian, meaning that each selected module is independent of the previously-selected modules; more formally, $p(\phi(t)|\phi(t-n),\cdot) = p(\phi(t)|\cdot)$ for all $n > 0$. Thus, it is possible that multiple consecutive module updates are directed at the same module, or $\phi(t) = \phi(t-n)$. Similar refixation behavior exists in human gaze during complex tasks; presumably observers use the visual information across multiple fixations for a continuous control signal for a single task. To make analysis simpler and more consistent between simulation and human results, we grouped multiple consecutive updates for a given module into a single "look." For instance, in the example shown in Figures 1 and 2, updates are provided first to module A, then B, then A twice, then B twice. In this example, each module receives two "looks," with the second look for each module being twice as long as the first.

Comparison with human results

Sullivan et al. (2012) instructed subjects driving in a virtual environment to follow a leader car and maintain a certain speed, but the priority of which of the two tasks was most important was varied so that one was high and the other was low. Additionally, subjects drove in some conditions where noise was added to the speed of the car, with the intent of disrupting the maintenance of a constant speed. These manipulations resulted in four conditions where either following a leader or maintaining a constant speed was most important, and velocity noise was either present or absent. They found that task priority increased fixation behavior on task-related objects. Additionally, an interaction between priority and uncertainty was found, whereby uncertainty alone did not guarantee increased fixation behavior. Instead, only if a task related object had sufficiently high priority did the addition of uncertainty further increase fixation behavior. Look duration histograms for this experiment are replicated in the top row of Figure 3.

We ran a set of simulations with our model attempting to replicate this behavior using parameters set to mimic the orginal human driver conditions. We used a simple grid search to locate these parameters. Because all of the parameters taken together can present a scaling ambiguity (e.g., if all $\varepsilon^{(i)}$ and $\rho^{(i)}$ are multiplied by 2, then the same qualitative behavior will result) we fixed $\rho^{(f)} = 1$ and explored only settings for the other parameters.

Once we identified the parameter settings corresponding to the experimental conditions, we evaluated our model by running it in each of these conditions 10 times, with each simulation run for approximately 4000 steps. The sequence of module updates for each simulation run was stored and labeled as looks as described above, then normalized to form a probability distribution. These results were compared the distributions of look durations from the human data. The model was able to capture several important aspects of the human data, including a sensitivity to both noise and priority, but also a gating effect whereby noise in low-priority tasks had a smaller effect than noise in high-priority tasks. Our results, shown under the human data in Figure 3, are qualitatively similar to the human performance in a virtual driving environment.

In addition to our scheduling model, a baseline fixation scheduler was run in the simulation. This scheduler incorporated only the priority of each task in selecting modules for update, but uncertainty was not incorporated. The results from this baseline scheduler are shown in the bottom row of Figure 3. The probability distributions from our scheduler and the baseline compared against the human data via the Kullback-Leibler (KL) divergence. Over all the conditions, our model had an average KL divergence of 2.20, versus 4.43 for the baseline scheduler (lower numbers are better).

Discussion

This paper described a modular, "soft" barrier approach for modeling eye movements in human drivers. The model includes explicit measurements of an agent's estimates of external world state, and uses a random walk to model the uncertainty in these estimates over time. Uncertainty, modulated by the priority of a task, is then used to arbitrate gaze allocations among competing modules. Our priority-plus-uncertainty model provides a better fit of a set of human driving data than a priority-only baseline fixation scheduling model. We are currently working on comparing this model to predictions from a standard salience model (Itti & Koch, 2001), a central bias model (Tatler & Vincent, 2009) and the original scheduling model that inspired our work (Sprague & Ballard, 2003). In addition, the softmax approach to selecting modules for update permits a clean inversion of the model; that is, given human eye fixation behavior, the model can be inverted to provide the most likely set of parameter settings to explain those data. We plan to develop this inversion more fully so as to replace the grid search described in this paper. Once the inversion process is in place, we can use this model to recover the task priorities and uncertainty levels that human drivers appear to be experiencing in these conditions.

References

Arulampalam, M., Maskell, S., Gordon, N., Clapp, T., Sci, D., Organ, T., & Adelaide, S. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

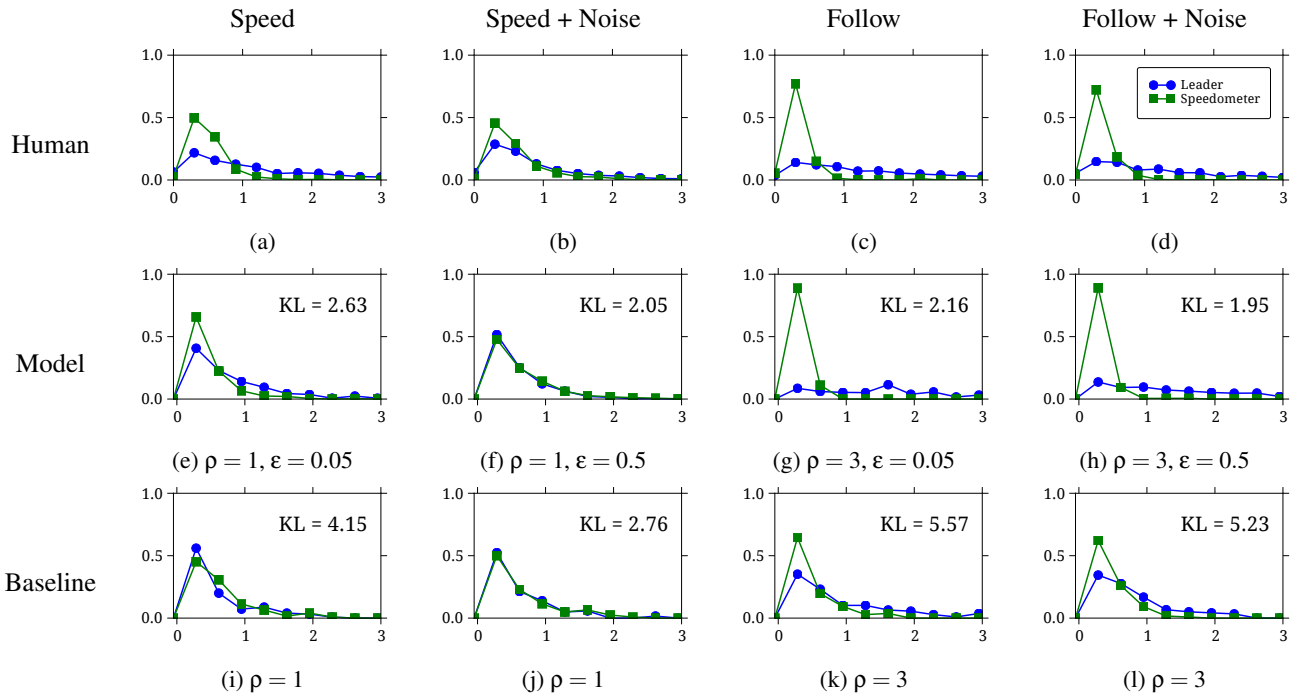

Figure 3: Distribution of look durations for human subjects (a-d; from Sullivan et al., 2012), model predictions (e-h), and baseline predictions (i-l). In all plots, look duration (in seconds) is shown along the abscissa, with the proportion of looks indicated on the ordinate. Looks to the speedometer are plotted with green squares; looks to the leader car are plotted with blue circles. (a, b) In conditions where driving at a target speed was emphasized, human looks at the speedometer were approximately matched in duration to looks at the leader car. (c, d) In conditions where following a leader car was emphasized, human looks at the speedometer were brief. Noise added to the car's speed (b, d) affected human looks in the speed task more than looks in the following task. Similar results hold for our model (e-h), but not for a baseline model that incorporates task priority but ignores the effects of uncertainty (i-l).

Bruce, N., & Tsotsos, J. (2009). Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, *9*(3).

Butko, N., & Movellan, J. (2010). Detecting contingencies: An infomax approach. *Neural Networks*, *23*(8), 973–984.

Carpenter, R., & Williams, M. (1995). Neural computation of log likelihood in control of saccadic eye movements. *Nature*, *377*(6544), 59–62.

Hayhoe, M., & Ballard, D. (2005). Eye movements in natural behavior. *Trends in cognitive sciences*, *9*(4), 188–194.

Itti, L., & Koch, C. (2001). Computational modeling of visual attention. *Nature reviews neuroscience*, *2*(3), 194–203.

Knudsen, E. (2007). Fundamental components of attention. *Annu. Rev. Neurosci.*, *30*, 57–78.

Land, M., & Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? *Vision research*, *41*(25), 3559–3565.

Nunez-Varela, J., Ravindran, B., & Wyatt, J. (2012). Gaze allocation analysis for a visually guided manipulation task. *Proc. From Animals to Animats*, 44–53.

O'Dwyer, A. (2006). *Handbook of PI and PID controller tuning rules* (Vol. 2). World Scientific.

Senders, J. (1980). *Visual scanning processes*. Soest, Netherlands: Drukkerij Neo Print.

Sprague, N., & Ballard, D. (2003). Eye movements for reward maximization. *Advances in neural information processing systems*, *16*.

Sullivan, B., Johnson, L., Ballard, D., & Hayhoe, M. (2011). A modular reinforcement learning model for human visuomotor behavior in a driving task. In *Proc. AISB Symposium* (pp. 33–40).

Sullivan, B., Johnson, L., Rothkopf, C., Ballard, D., & Hayhoe, M. (2012). The role of uncertainty and reward on eye movements in a virtual driving task. *Journal of Vision*, *12*(13).

Tatler, B., & Vincent, B. (2009). The prominence of behavioural biases in eye guidance. *Visual Cognition*, *17*(6-7), 1029–1054.

Wolfe, J., Butcher, S., Lee, C., & Hyle, M. (2003). Changing your mind: on the contributions of top-down and bottom-up guidance in visual search for feature singletons. *Journal of Experimental Psychology: Human Perception and Performance*, *29*(2), 483.

Zhang, L., Tong, M., Marks, T., Shan, H., & Cottrell, G. (2008). SUN: A bayesian framework for saliency using natural statistics. *Journal of Vision*, *8*(7).