Infomax Control of Eye Movements

Nicholas J. Butko and Javier R. Movellan

***Abstract*—Recently, infomax methods of optimal control have begun to reshape how we think about active information gathering. We show how such methods can be used to formulate the problem of choosing where to look. We show how an optimal eye movement controller can be learned from subjective experiences of information gathering, and we explore in simulation properties of the optimal controller. This controller outperforms other eye movement strategies proposed in the literature. The learned eye movement strategies are tailored to the specific visual system of the learner—we show that agents with different kinds of eyes should follow different eye movement strategies. Then we use these insights to build an autonomous computer program that follows this approach and learns to search for faces in images faster than current state-of-the-art techniques. The context of these results is search in static scenes, but the approach extends easily, and gives further efficiency gains, to dynamic tracking tasks. A limitation of infomax methods is that they require probabilistic models of uncertainty of the sensory system, the motor system, and the external world. In the final section of this paper, we propose future avenues of research by which autonomous physical agents may use developmental experience to subjectively characterize the uncertainties they face.**

***Index Terms*—Eye movement, face detection, infomax control, object detection, policy gradient, visual search.**

I. Introduction

IN DAILY life, we constantly seek information that makes us more certain about questions of interest. We might check Wikipedia to regain certainty about the answer to "Who was the 17th president?" or we might look at the sky to help predict whether it will rain soon. But not all information gathering is conscious. When I play tennis, my eyes move to regions of the visual scene that answer the question, "How should I swing my arm to hit ball the way I want?" As you read, your eyes automatically saccade to words and letters that help you answer the question, "What is this author trying to convey?"

Humans make over 150 000 saccades per waking day, spending about 1.5–2 h in saccadic flight, during which useful vision is very poor [1]. Every second of every minute of our waking lives, we make unconscious decisions about where to look; we decide which photons to sense in order to help us get the information we need to make it through our day and accomplish our goals. Some of these eye movement decisions may have life-and-death consequences: if we look the wrong way when crossing a road, we may be killed.

In this paper, we consider the problem "How should an agent direct its eyes to best gather information?" from a computational, or optimality point of view. We make the following contributions.

1) We present several existing models of eye movements and relate them to the approach based on optimal information gathering. We review other domains where optimal information gathering techniques have been applied.
2) We analyze the question of where to look as a problem in stochastic optimal control. This requires that we characterize the uncertainties in our sensors (eyes), actuators (muscles), and target dynamics. Once we have characterized these uncertainties, we can quantify the information provided by eye movements. We show how the optimal eye movements change depending on the sensor characteristics. For example, we show that a robot may want to move its cameras differently from how a human moves her eyes.
3) We show that information can be used as a reward signal to learn efficient eye movement behavior.
4) We follow the approach above to build versatile "digital eye" that efficiently scans images to find objects of interest.
5) We discuss the remaining steps necessary to account for a fully autonomous developmental model: how do infants and robots use statistical regularities among sensors and actuators to characterize uncertainties in unsupervised, self-contained, and verifiable terms?

A. Different Views of Eye Movement

Many researchers have considered the problem of why humans move their eyes the way they do. An important class of models explain eye movements from the point of view of visual salience. We call these *salience* models. Salience models typically attempt to predict eye fixation histograms, i.e., the relative probability with which people will look at particular regions of an image. An image region is considered salient in an experimental condition if people tend to saccade to that area with high frequency in that condition.

Among salience models, a distinction can be made between *descriptive* models, *structural* models, and *computational* models (Fig. 1). Descriptive models are agnostic about why people look at certain places, and just attempt to predict where they will look. In their purest form these models are just functions whose inputs are images, and whose outputs are pixel by pixel probabilities that the pixels will be looked at. These probabilities are learned from examples of images and the locations where people look at in those images [2]. Structural

Manuscript received January 31, 2010; revised May 04, 2010; accepted May 14, 2010. Date of publication May 24, 2010; date of current version June 30, 2010. This work was supported by the National Science Foundation (NSF) under Grant IIS INT2 0808767, and by the Infomax Neural Networks for Real-Time Learning and Control under Grant ECCS 0622229.

N. J. Butko is with the Cognitive Science Department, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: nbutko@cogsci.ucsd.edu).

J. R. Movellan is with the Institute for Neural Computation, La Jolla, CA 92093 USA (e-mail: movellan@mplab.ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2010.2051029

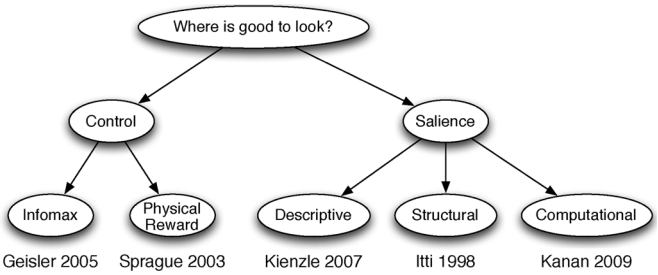


Fig. 1. Taxonomy of eye movement models with example references, which are not exhaustive. More references and discussion can be found in the text.

models appeal to neural mechanisms [3], or mental mechanisms [4], [5] to explain why some image regions are more salient than others. Computational models explain observed behaviors as solutions to some problem or objective [6]. For example, some recent visual salience algorithms, e.g., [7]–[10] propose that when people view images they are implicitly trying to maximize the chance of looking directly at a visual search target [11]. According to these models, the eyes move to regions of the image that were most likely rendered by one such object. Other models propose that people have the computational objective of moving their eyes to "surprising" locations, which are defined as locations that contain the most information about local image statistics [12], [13]. We review the relationship between surprise based information models of eye movements and infomax control models of eye movements shortly (Section I-C).

In this paper, we present a computational analysis of eye motion from the point of view of the theory of stochastic optimal control. Before we do so, we wish to clarify some crucial differences between the *salience* models described above, and the *control* models of the type we pursue in this paper.

(*1*) Salience models are designed to predict eye fixation histograms, i.e., the frequency with a typical person fixates regions of a given image. As such, by definition, salience models do not provide reasons to look at things that are not currently visible. In contrast, control models describe optimal policies to move visual sensors of known characteristics so as to best achieve given tasks. In control models, it is often valuable to look at regions that are not currently visible so as to gain more information about those regions.

(*2*) Visual search based computational salience models assume that there is a low resolution (e.g., periphery) and a high resolution (e.g., fovea) processing system. The low resolution system chooses the region that most probably contains a target of interest and triggers a saccade to that region. The foveal system then proceeds to process the local region that was just fixated. While this is a reasonable story, it has not been justified from an optimality point of view. An optimality approach requires evaluation of the expected information gain that the high resolution system would provide if the eye were to fixate on that pixel. This evaluation requires a full specification of the high resolution process, in addition to an integration over the possible outcomes of the high resolution process to compute an expected reward.

In fact, salience models give no specification for the properties of the high resolution process or its reliability of inferring target presence, nor do they integrate over potential consequences of the eye movement. They cannot be evaluated from an optimality point of view because the benefit to the organism of the eye movement cannot be computed. Instead, we are led to believe that optimal eye motion is independent of these parameters. We can assert that it is a good idea to try to look directly where you think a search target is to confirm its presence, but this assertion is of no consolation to a tiger who does not want to spook his prey, or to the astronomer trying to see faint stars.

In contrast, in control models the foveal-peripheral characteristics of the visual sensors need to be specified. This allows evaluation of the expected information gain of an eye movement prior to making the movement, thus orienting the eyes in an optimal manner. As we will see in this paper, the characteristics of the foveal and peripheral systems do affect the way in which the eyes should move. In some cases, optimal eye motion entails looking away from the regions that most probably contain the target of interest, in direct violation of the stated computational objective of some visual salience models.

(*3*) Since salience models are designed to explain fixation histograms, they are agnostic about the sequencing of eye movements and about how the information observed up to time t influences the decisions to move our eyes to other locations. To explain sequencing effects, like the fact that people are less likely to look at previously scanned locations, salience models appeal to notions such as "inhibition of return." While useful at a descriptive or structural level of analysis, inhibition of return is not justified from the point of view of salience algorithms' stated computational objectives.

Control models on the other hand need to be explicit about the information collected after each fixation. In control models, after each eye movement, information is gathered and changes the opinion and sense of certainty about how the world is. In turn, these new opinions and sense of certainty combine to direct the eyes to a new location to help achieve some specific task. Control models give a computationally grounded justification for an effect that looks like inhibition of return: to achieve most tasks, you do not want to just look in the same place always [14]. This task can be something physical, e.g., "pick up and throw away garbage" [15] and "track a moving cursor with an unreliable joystick" [16], or it could be purely exploratory, gathering information as quickly as possible, which we call infomax (Fig. 1) [14]. Purely exploratory eye movements may have evolved to be intrinsically rewarding because they are useful in learning strategies to achieve a variety of goals in a variety of environments [17].

(*4*) Finally, a common distinction made in the literature is "top–down" versus "bottom–up" salience. Some papers make this distinction from a functional perspective. Bottom–up salience is supposed to be governed only by the characteristics of the stimulus alone. Top–down salience is supposed to be modulated by the current goals and tasks of the individual [18]. A fundamental problem with this functional distinction is that there is no such thing as a taskless condition. When subjects are asked to freely look at an image they are consciously or unconsciously performing a task. Some papers avoid this problem

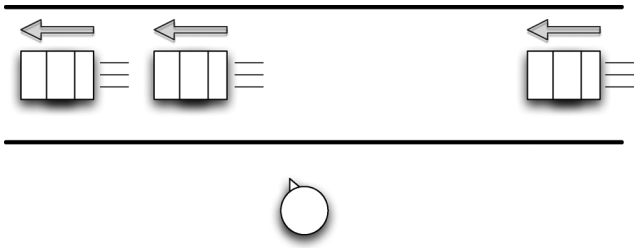


Fig. 2. We get more information about whether it is safe to cross this one-way street by looking to the right than by looking to the left.

by applying a mechanistic point of view: bottom–up salience is supposed to refer to the output of mechanisms (mental or neural) that transmit information in a unidirectional manner from peripheral to central processing systems. However, this notion is also problematic. The brain is fundamentally an interactive system: visual information has an effect on the activity of auditory cortex [19], [20]. Beliefs and expectations modulate primary visual cortex [21]. Moreover psychological laws that were supposed to be the signature of feedforward, bottom–up processing, can be reproduced in interactive processing systems in which the notion of bottom–up and top–down processing does not apply. Thus in this paper, we abandon the top–down versus bottom–up terminology.

B. Notation Standards

We leave implicit the probability space over which random variables are defined. Capital letters typically represent random variables and vectors. Lower case letters represent specific values taken by random variables. For example, $X = x$ indicates that the random variable X has taken the specific value x, technically a set of outcomes. We leave implicit the distinction between probability mass functions (for discrete random variables) and probability density functions (for continuous random variables). When possible we identify probability functions by their arguments. For example $p(x)$ represents the probability mass (if X is discrete) or probability density (if X is continuous) of the random variable X evaluated at the specific value x. We use colons to represent sequences of random variables. For example $X_{1:t} = (X_1, \cdots, X_t)$.

C. The Value of Information

Consider the problem of crossing a one-way street like the one shown in Fig. 2. The faster we manage to cross safely to the other side of the road the better we have accomplished our goal. The world can be in one of two states: $S_t = 0$, indicates that it is unsafe to cross at the current time t, and $S_t = 1$ means that crossing is safe. $H_{t-1} = (O_{1:t-1}, A_{1:t-1})$ represents the history of actions and observations up to time t. $p(S_t = 1|h_{t-1})$ is our belief, based on the history of observations h_{t-1} as to whether or not it is safe to cross. We can take three actions: $A_t = l$ means that we look left, $A_t = r$ means that we look right (where the cars are coming from), and $A_t = c$ means that we cross. Our beliefs about S_t are shaped by the observations provided by our visual system. For simplicity, assume the system tells us whether or not a car is present in the field of view: $O_t = 0$ if no car is visible and $O_t = 1$ if some car is visible.

If we look to the left, we will see the cars that just passed (because the cars come from the right). This will give us some information, for example, how busy the street generally is. If for the last minute we only saw one car, then this is a pretty safe street to cross, but if we saw fifty, we know it is a heavily travelled highway that is quite perilous. However, we will not get any indication about *what cars are currently coming*, and so we will always be somewhat uncertain about whether it is safe to cross. If we look right, we will see the cars that are about to come, and can be much more certain about when exactly is a safe time to cross (Fig. 2). Thus given the task at hand, the information gained by looking right is more valuable than the information gained by looking left.

The key here is that looking right provides more information than looking left about a key state of the world. Mathematically, the *information* that a specific observation o_t provides *about* a variable of interest S_t is the reduction of uncertainty about S_t due to that observation

$$\mathcal{I}(S_t, o_t|h_{t-1}) = \mathcal{H}(S_t|h_{t-1}) - \mathcal{H}(S_t|h_{t-1}, o_t) \quad (1)$$

$$= \int p(s_t|h_{t-1}, o_t) \log p(s_t|h_{t-1}, o_t) ds_t - \int p(s_t|h_{t-1}) \log p(s_t|h_{t-1}) ds_t. \quad (2)$$

In infomax control problems, we evaluate potential actions in terms of the information gain we expect them to provide. Thus given an action a_t, we need to take expected values across all possible observations following the action a_t

$$\mathcal{I}(S_t, O_t|h_{t-1}, a_t)$$

$$= \int p(o_t|h_{t-1}, a_t) \mathcal{I}(S_t, o_t|h_{t-1}) do_t$$

$$= \int p(s_t, o_t|h_{t-1} a_t) \log p(s_t|h_{t-1}, o_t, a_t) ds_t do_t - \int p(s_t|h_{t-1}) \log p(s_t|h_{t-1}) ds_t \quad (3)$$

$$= \mathcal{H}(S_t|h_{t-1}) - \mathcal{H}(S_t|O_t, h_{t-1}, a_t) \quad (4)$$

where the Shannon entropy $\mathcal{H}$ is a measure of uncertainty. Note that $\mathcal{H}(S_t|h_{t-1})$ is constant with respect to a_t and therefore, in order to maximize the expected information gain, we need to choose action that minimizes the expected entropy of the posterior probability distribution of S_t. In our example, $\mathcal{H}(S_t|O_t, h_{t-1}, A_t = r)$ will be smaller than $\mathcal{H}(S_t|O_t, h_{t-1}, A_t = l)$ and thus, we choose to look right. Equation (3) reveals the details that are needed to be able to quantify information: we need to have both a concrete notion of how probable each state of the world is, $p(s_t|h_t)$, and also a model of how everything we've done and seen has changed that probability, $p(s_t|h_t, o_t, a_t)$.

Some authors in the visual salience literature [13] have promoted the idea of "Bayesian surprise" as a way to evaluate the salience of a visual region. The Bayesian surprise provided by an observation o_t is defined as the KL divergence between the

prior distribution of a state and the posterior distribution of the state given the observation

$$\mathcal{B}(S_t, o_t|h_{t-1}) = \int p(s_t|h_{t-1}, o_t) \log \frac{p(s_t|h_{t-1}, o_t)}{p(s_t|h_{t-1})} ds_t. \quad (5)$$

The *expected surprise* of an observation turns out to be the mutual information that the observation gives about the state provided by an action a_t

$$\mathcal{B}(S_t, O_t|h_{t-1}, a_t) = \int p(o_t|h_{t-1}a_t)\mathcal{B}(S_t, o_t|h_{t-1}) do_t \quad (6)$$

$$= \int p(s_t|h_{t-1})p(o_t|h_{t-1}s_t, a_t) \times \log \frac{p(s_t|h_{t-1}, o_t, a_t)}{p(s_t|h_{t-1})} ds_t do_t \quad (7)$$

$$= \mathcal{I}(S_t, O_t|h_{t-1}, A_t = a_t). \quad (8)$$

Thus, expected surprise and information gain are equivalent metrics for evaluating the value of actions.

So what separates a surprise based salience model from an infomax control model? First, the state of interest in [13] is "parameters of local image statistics." With this state space, surprise is only defined for the image which was already seen, and so there is no reason to look at something that is not currently observed. Second, surprise models are *reactive*: They only react to what has already been seen, as in equation (5). Control models consider (sometimes implicitly) the consequence of future actions and observations, as in equation (6), making them *proactive*. They act in the way that will best help achieve some future goal. This highlights two main differences between salience models and control models.

D. Infomax in Other Domains

Maximization of expected information gain was proposed by Lindley [22] as a sensible criterion for designing experiments. Stone [23] and Fedorov [24] applied this idea to the efficient estimation of parameters in linear regression and ANOVA models. Bernardo [25] used a Bayesian framework to show that information gain can be used as a utility function in the context of optimal control. While exact solutions to infomax control were found for linear problems, they proved difficult for even the simplest nonlinear problem. For this reason, information maximization approaches languished for a number of years.

Recent years have seen a flourishing of approximate solutions to stochastic optimal control problems, some of which can be applied to difficult infomax control problems. Lewi *et al.* found a very efficient approach to find approximate infomax solutions to the problem of parameter estimation in generalized linear models. They used the approach to choose which stimuli to present to a neuron so as estimate the properties of its receptive field. They showed that the approach could reduce the total experiment time by an order of magnitude [26].

Infomax approaches have also been used to develop unsupervised learning algorithms. Bell and Sejnowski showed that when this learning algorithm is applied to artificial neural networks exposed to natural images they develop Gabor receptive fields similar to those found in simple cells in primary visual cortex [27]. Movellan *et al.* [28] showed that information maximization could be used to model how humans ask questions in active concept learning tasks. Movellan and Butko [29], [30] showed that nine-month-old infants schedule vocalizations so as to optimally detect contingent social interaction. They also showed that information gain could be used as a reward for reinforcement learning algorithms and explain the developmental trajectories observed in infants.

Cakmak *et al.* showed that robot learning improved when robots asked human teachers questions that would give the robots most information, and also that the teaching interactions were more motivating to the human teachers [31].

A recent class of approaches uses the submodular property of information to approximate optimal information gathering. This property describes mathematically the diminishing information returns of subsequent probes of nearby areas. These approaches have been used to optimally deploy sensors to effectively monitor environmental factors in lakes [32], and in active-learning scenarios to quickly learn how to accurately diagnose health conditions from medical images [33].

II. Problem Statement

To think systematically about information and information gathering, it is useful to formulate eye movement problems as partially observable markov decision processes (POMDPs). To make this more concrete, consider a control-based model of eye movement in which our goal is to play "Where's Waldo?", a popular children's game where the goal is to find a visually distinct man named Waldo as quickly as possible from among a wide field of distractors [34]. This game is analogous to a situation in which an observer moves her eyes in order to search a 2-D image plane of bounded size for a target that is not moving.

A. POMDP Problem Formulation

A POMDP is defined by the following elements [35] (with their correspondences in the Where's Waldo? control model).

- S_t is a random variable that represents the state of the world at time t. In this paper, the bounded area in which the target can appear is covered by a grid of N total elements, which we refer to as the *visual array*. In the Waldo example, $S_t = i$ means that Waldo is at location i, at time t.
- A_t is random variable that represents the action taken by the agent at time t. In the Waldo example, $A_t = k$ means that the agent fixated location k at time t.
- O_t is a random variable that represents the sensor outputs (observations) available at time t. In the general case, the sensors are noisy and provide only partial evidence about the state of the world.
- $p(s_{t+1}|s_{1:t}, a_{1:t}, o_{1:t}) = p(s_{t+1}|s_t, a_t)$: Markovian system dynamics—How the state changes naturally over time, and also based on the agent's actions. In Where's Waldo?, Waldo does not move so $p(s_{t+1}|s_t, a_t) = 1$ if $s_{t+1} = s_t$, 0 otherwise.
- $p(o_t|s_{1:t}, a_{1:t}) = p(o_t|s_t, a_t)$: Markovian observation model—How objects appear at different points in the fovea or periphery. Red and white stripes in your periphery could possibly be Waldo; a man with a camera, striped shirt and blue pants in your fovea is definitely Waldo.

B. Belief State

A critical concept in POMDPs is the "Belief State" $B_t = (B_t^1, \cdots, B_t^N)$, where B_t^i is the probability that the target is at location i at time t, given all the actions taken and observations received up to time t

$$B_t^i \stackrel{\text{def}}{=} p(S_t = i | A_{1:t}, O_{1:t}). \tag{9}$$

It is easy to show that the belief state vector at time t is a function of A_t, O_t and B_{t-1}. Specifically, given a sequence of actions $a_{1:t}$ and observations $o_{1:t}$, then

$$\begin{aligned} b_t^j &= p(S_t = j | a_{1:t}, o_{1:t}) \propto p(S_t = j, o_t | a_{1:t}, o_{1:t-1}) && (10)\\ &= p(o_t | S_t = j, a_t) p(S_t = j | a_{1:t}, o_{1:t-1}) \\ &= p(o_t | S_t = j, a_t) \sum_{i=1}^{N} p(S_t = j | S_{t-1} = i, a_t) b_{t-1}^i. && (11) \end{aligned}$$

Waldo never moves, so this becomes

$$b_t^j = \frac{p(o_t | S_t = j, a_t) b_{t-1}^j}{\sum_{k=1}^{N} p(o_t | S_t = k, a_t) b_{t-1}^k}. \tag{12}$$

Thus, the belief state B_t encodes all the relevant history of an agent's actions and observations. In the control model of visual search presented below, the belief state representation is the same size as a single observation. This speaks against arguments about the "cost" of memory. For example, [14] argues that subjects forget what they've seen because it is simply too costly to remember many observations. But equation (11) tells us that there is practically no cost to memory: to remember everything *that is relevant* about the entire history of observations you just need to store your current belief, which in the visual search case requires exactly $N - 1$ real valued numbers. A computational level explanation is that events are "forgotten" because doing so improves task performance. If Waldo is likely to move, it is almost completely irrelevant where he was or was not five minutes ago. Since the POMDP belief state only encodes relevant information, the agent would appear to an outside observer to have forgotten where Waldo was five minutes ago. This would lead to an effect that looks like forgetting, even though you still remember all that is relevant about everything you've seen up to this point.

An aspect of the POMDP approach is that it prescribes a level of remembering and forgetting that is optimal for the statistics of movement of relevant search targets. The amount of forgetting observed in psychophysical experiments such as those gathered in [14], is in fact an indication about the implicit beliefs implemented by the brain. These implicit beliefs may reflect (be optimal for) the statistics of the environment in which the brain operates.

C. Information Reward

Infomax control problems are ones in which we wish to act in such a way as to optimally gather information about some unknown thing in the environment. Gathering information about the unknown answer to a question like "Where's Waldo?" is equivalent to minimizing the entropy (uncertainty) of belief vector B about Waldo's location. In the language of optimal control, we let the instantaneous reward R_t to be a decreasing function of the entropy of the state belief

$$\begin{aligned} R_t &= -w_t \mathcal{H}(S_t | A_{1:t}, O_{1:t}) \\ &= w_t \sum_{j=i}^{N} B_t^j \log B_t^j \end{aligned} \tag{13}$$

where $w_t \geq 0$ is a constant that determines the relative value of being certain at time t. A policy π is a function that maps beliefs into actions, i.e., $A_{t+1} = \pi_t(B_t)$. The value of a specific belief state b_t given a specific policy π is a weighted sum of expected rewards up to a terminal time point T conditioned on that policy

$$V_t^{\pi}(b_t) = \sum_{s=t}^{T} E[R_t | b_t, \pi]. \tag{14}$$

The goal of infomax is to find a policy π^* that maximizes the overall value

$$\pi_t^*(b_t) = \underset{\pi}{\operatorname{argmax}}\, V_t^{\pi}(b_t). \tag{15}$$

At first sight, the Infomax reward function appears peculiar in that it is based on our own beliefs. For example, "It doesn't matter to me where Waldo is; it just matters that I am sure of where he is." This is in fact a typical of POMDP problems, not just infomax problems. Kaelbling *et al.* observesthat the POMDP reward function is strange in that the agent appears to derive reward from belief rather than the environment. However, the beliefs in POMDPs are not arbitrary. They are constrained by correct Bayesian inference based on observation from the environment. Thus, is not possible to pursue a strategy of self-delusion to achieve reward. Rather, the agent's expectation of reward is the true expectation of reward, and so the experienced reward will (on average) meet the agent's expectation when planning [35].

D. Components of Uncertainty

In order to develop optimality models of visual search, we must specify both an observation model in the form of a family of distributions $p(o_t | s_t, a_t)$ specifying how the world may look like, and the system's dynamics model in the form of a family of distributions $p(s_t | s_{t-1}, a_t)$ specifying how the world may change in the future. In [15] and [30], these probability distributions were constructed by creating simulated worlds. Since the researchers constructed the worlds, they knew precisely the uncertainties in those worlds. In [14], psychophysical stimuli were carefully created to constrain the observation model to be a linear filter with Gaussian noise, and the parameters of the Gaussian noise model for human eyes were fit psychophysically at different points of retinal eccentricity.

Without specifying these probability models and their associated uncertainty, we cannot compute the belief update in (11), or the information reward in (13). The following are examples of sources of uncertainty that may be considered in modeling eye movement (Fig. 3).

- **Target Uncertainty**: How are objects likely to move on their own, when my eyes do not move? Can my eye movements affect the motion of external objects?
- **Action Uncertainty**: How reliably can my eyes move?

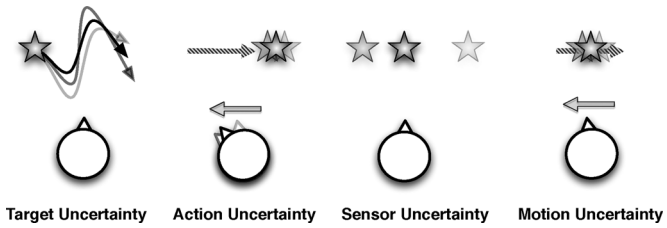


Fig. 3. Different factors introduce uncertainty in visual search targets localization. A few examples of these many factors are: how targets will move, the reliability of our own muscles, loss of reliability at visual eccentricity, and motion blur or distortion.

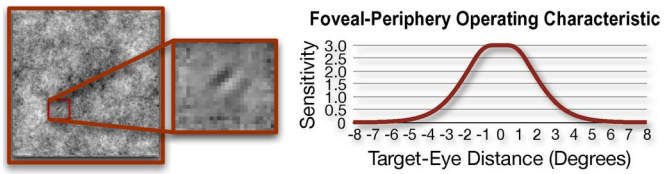


Fig. 4. **Left**: A wavelet is "hidden" in a pink noise background. **Right**: Najemnik & Geisler measured subjects' ability to detect these targets as a function of how far away they were looking.

- **Sensor Uncertainty**: How does the appearance of an object change based on its distance to my center of gaze?
- **Motion Induced Uncertainty**: How does the appearance of an object change while my eye is in motion? For example, things may be blurry, distorted, or completely invisible while the eye is in motion, depending on the physical characteristics of the occulomotor system.

III. A Control Model of Visual Search

In this section, we present a psychophysical model of visual search developed by Najemnik & Geisler (N&G) [14], and reformulate it from the point of view of stochastic optimal control and later extend it so as to overcome two of its limitations: (1) the fact that the model achieves optimality with respect to a single fixation rather than a sequence of fixations; (2) the fact that the model assumes Gaussian sensors and nonmoving targets.

In N&G's model, the task is to find a target stimulus (a Gabor wavelet) in a correlated Gaussian noise background (Fig. 4). The optimal procedure to infer the target's location is to filter the image with a linear filter matched to the target stimulus. In N&G's model, the sensitivity of the matched filter decreases with the eccentricity from the fixation point. This foveal-perihperal sensitivity is measured empirically using psychophysical experiments to determined how likely subjects are to detect such a wavelet at different eccentricities. An example of the foveal-peripheral operating characteristic (FPOC) curves measured in this fashion by N&G is shown in Fig. 4.

In terms of the sources of uncertainty described in Fig. 3, N&G's model can be summarized as the following.

- **Target Uncertainty**: None (the search target never moves).
- **Action Uncertainty**: None (the eye moves reliably).
- **Retinal Uncertainty**: Signal plus eccentricity-dependent Gaussian noise, detailed below.

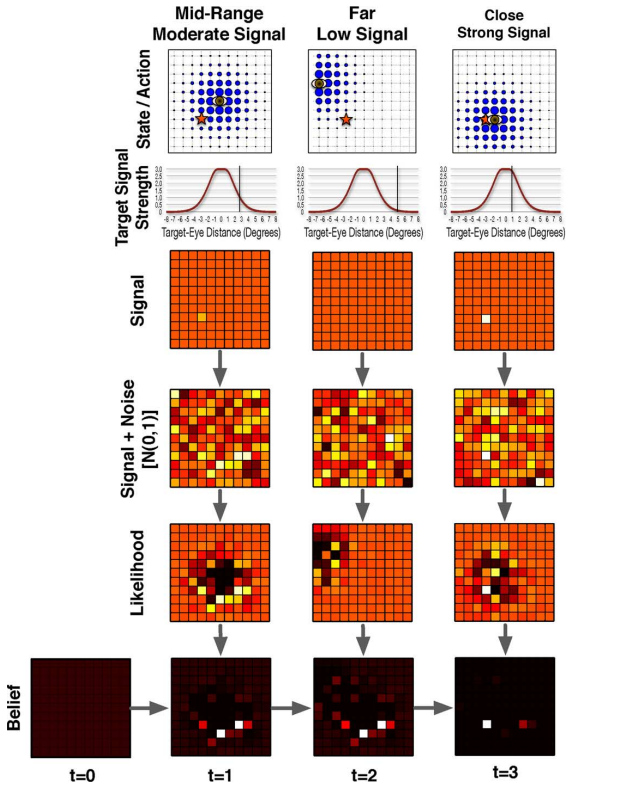


Fig. 5. I-POMDP model of eye movement: A target is located at a visual location previously unknown to the subject. After making several fixations, the subject comes to know with high confidence the location of the visual target. See text for further description.

- **Motion Uncertainty**: None (eye movements are instantaneous, so there is no chance for motion blur or shear, etc.)

An illustration of a typical trial of this model is show in Fig. 5. A noisy observation $o_t^j \in \mathcal{R}$ is sampled at each potential target location j at each timestep t. This noisy observation is illustrated in the "Signal+Noise" row of Fig. 5. In locations without a target, the observation is drawn from a baseline Gaussian distribution, which has zero-mean and standard deviation 1. These zero-mean locations are shown as darker regions in the "Signal" row of Fig. 5. Only the single observation directly at the target location is drawn from the "target" Gaussian distribution. The standard deviation of the target distribution is always 1. The mean of the target increases as the target approaches the foveal region (the brightest location in the "Signal" row) and converges towards zero as the eccentricity increases. Thus, the equation for the observation at location j, given that the target is at location S_t and the eye is focused on location k, is as follows:

$$O_t^j = \delta(S_t, j) d_{j,k} + Z_t^i \tag{16}$$

where Z_t^i is independent and identically distributed (i.i.d.) zero mean, unit variance Gaussian random noise, $\delta(S_t, j) = 1$ if $S_t = j$, zero, otherwise and $d_{j,k}$ is the discriminability of a target at location j given that the fovea is centered at location k. We call this the FPOC of location j given that the retina is centered at k. In humans, the FPOC $d_{j,k}$ decreases with increased distance of location j from the current point of fixation k, meaning farther from the point of fixation, it becomes harder to discriminate an observation caused by target-based activity

from one caused by noise alone. This is illustrated in the "Target Signal Strength" row of Fig. 5.

Under the model, the individual observations O_t^j are conditionally independent given the external scene,[1] and so the likelihood of an entire vector of observations $o_t = (o_t^1, \cdots, p_t^N)$ given that the target is at location i and the eye is focusing on location k is as follows:

$$\begin{aligned} p(o_t|S_t = i, A_t = k) &= \prod_{j=1}^{N} p\left(o_t^j|S_t = i, A_t = k\right) \\ &= 1/\sqrt{2\pi}\exp\left(\left(o_t^i - d_{i,k}\right)^2/2\right) \\ &\quad \times \prod_{j\neq i} 1/\sqrt{2\pi}\exp\left(\left(o_t^j\right)^2/2\right) \\ &= 1/\sqrt{2\pi}\frac{\exp\left(\left(o_t^i - d_{i,k}\right)^2/2\right)}{\exp\left(\left(o_t^i\right)^2/2\right)} \\ &\quad \times \prod_{j} 1/\sqrt{2\pi}\exp\left(\left(o_t^j\right)^2/2\right) \\ &= \frac{\exp\left(\left(o_t^i - d_{i,k}\right)^2/2\right)}{\exp\left(\left(o_t^i\right)^2/2\right)}Z \\ &= \exp(\alpha_{i,k}d_{i,k})K; \\ &\quad \alpha_{i,k} \stackrel{\text{def}}{=} \left(o_t^i - d_{i,k}/2\right) \end{aligned} \tag{17}$$

where K is identical for all i, k. This gives a likelihood that the Signal+Noise observation was generated by each possible target location ("Likelihood" row of Fig. 5). Combining this with (12) yields the proportional belief update ("Belief" row of Fig. 5)

$$b_{t+1}^i \propto \exp(\alpha_{i,k}d_{i,k})b_t^i. \tag{18}$$

Note the simplicity of the belief update. Even though the model has a large state, observation, and action space, updating beliefs is computationally efficient. To calculate the relative probability that an entire observation vector was caused by a state, we need constant time (only a single element of that observation vector is considered). Thus, the process of computing the belief update for all beliefs grows linearly. The belief about location of the search target could be updated with simple neural circuitry and strictly local update rules.

IV. Learning Where to Look

N&G [14] modeled visual search as a control strategy designed to detect the location of a visual target under sensor uncertainty. The observer plans one saccade at a time. At each saccade, the observer chooses to fixate the location that best improves the chances of being correct after that fixation. In this Section, we reformulate N&G model from the point of view of the theory of stochastic optimal control (in particular the theory of POMDPs). We find optimal infomax policies, show how these policies change with the FPOC of the observer, and show that using information as a reward signal leads to a better search strategy than N&G's ideal observer. Hereafter, we refer to the infomax POMDP version of N&G's model as I-POMDP.

First we explore whether a simulated agent could use information gain as a training signal to learn efficient eye movement policies. Our goal is to learn a policy $\pi(B_t) \rightarrow A_{t+1}$ that approximates the optimal policy defined in (15). Algorithms for learning exactly optimal policies in POMDPs exist, but are only feasible with few states, actions, and observations [35]. Point-based approximation methods can learn approximately optimal policies for POMDPs with many states and actions, but require a small observation space [36]. The I-POMDP model has an $\mathcal{R}^N$ observation space, which is very large. Moreover, these algorithms capitalize on the guarantee of traditional POMDPs that the reward function be linear in the belief vector b_t; I-POMDPs allow nonbelief-linear reward functions like (13).

A. Policy Gradient

Due to the limitations of these approaches, here we consider function approximation methods which find locally optimal policy functions over a family of functions parameterized by a vector θ. Each setting of θ corresponds to a specific policy. It is possible to derive the gradient of the value function in (14) with respect to θ [37].

An unbiased estimate of this gradient can be obtained by sampling a finite set of belief trajectories and collecting the corresponding rewards. This results in a simple update procedure, derived in [37].

1) Choose an initial value for θ.
2) Set $t = t_0$; Get an initial belief state b_t. Set $\vec{z} = \vec{0}$.
3) Run the system for one time step: take an action using the policy θ, make an observation, update the belief, from b_t to b_{t+1} and collect the reward r_{t+1} corresponding to that belief.
   - If b_{t+1} is a final state or $t = T$, go to 2.
   - Set $\vec{z} \leftarrow \vec{z} + \beta(\nabla_\theta p(b_{t+1}|b_t, \theta)/p(b_{t+1}|b_t, \theta))$.
   - Set $\theta \leftarrow \theta + \gamma_t r_{t+1}\vec{z}$.
   - Set $t \leftarrow t + 1$.
   - Set $b_t \leftarrow b_{t+1}$.
4) Go to 3).

where γ_t is a learning rate which can anneal over time, and β is a "bias-variance trade-off" parameter. Arguably the most challenging aspect of policy gradient methods is computing the quantity, $\nabla_\theta p(b_{t+1}|b_t, \theta)/p(b_{t+1}|b_t, \theta)$. In Appendix 1 and 2, we show how this can be done for logistic policies of the type described below.

B. Policy Gradient With Logistic Policies

We parameterize the policy as a logistic function. Let the parameter θ be a matrix with ith row θ^i. For a given θ, the probability of choosing an action k given a belief b_t takes the following form

$$p(A_{t+1} = k|b_t, \theta) = \frac{\exp\left(\theta^k \cdot \phi(b_t)\right)}{\sum_{i=1}^{n}\exp\left(\theta^i \cdot \phi(b_t)\right)} \tag{19}$$

where $\phi(\cdot)$ is a feature function that takes the belief vector as input and outputs another vector. Logistic policies can

[1]Note this does not require that the observations are independent, only that the sensors are noisy and the noise in each sensory element were independent.

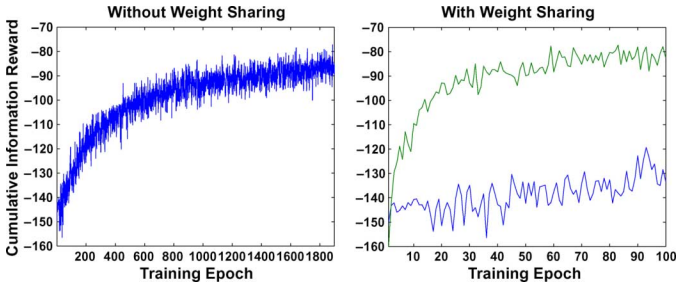


Fig. 6. **Left**: Policy gradient enables learning even when there are 14 641 parameters. **Right**: Learning is 20 times faster when we use weight sharing to exploit invariances, reducing the number of parameters to 61. The original learning curve is duplicated in blue in "With Weight Sharing" to highlight this timescale difference.

be thought of as a neural network, with an input layer (the featurized belief vector) projecting to an output layer in which each output unit represents the probability of fixating a given location. In the current work, we used $\phi(b_t) \triangleq b_t$, i.e., the input was just the current belief vector. The model is parameterized by θ, an $N \times N$ matrix, where N is the size of the visual array. Logistic policies extend many of the policies assumed in previous models (e.g., greedy search, random search) while allowing an intuitive examination of the learned policy. For example, a reasonable policy might be "look directly where the probability of the target is largest." We could verify whether this policy was optimal by examining the learned parameter matrix for very large values on the diagonal. This would mean that high belief that a target is at a given location leads to a high probability of fixating that location. Meanwhile connections to nodes at farther distances would taper off.

C. Convolutional Policies

The policy model in (19) can have many parameters. For an 11×11 visual array, there are 14 641 parameters. Fig. 6 shows that it is indeed possible to learn a good policy in such a situation, but it takes a long time. The search space can be reduced to 61 parameters by exploiting the shift- and rotation-invariances of most visual search problems.[2] This approach results in a convolutional policy which is defined by a rotationally symmetric, two-dimensional kernel. Under convolutional policies of this type the value of a belief map is obtained by filtering with a filter whose impulse response equals the policy's two-dimensional kernel.

Gradients for a convolutional policy can be learned via weight-sharing, by tying the parameters of all connections to locations equidistant from the point of fixation. This involves computing the full gradient, and then adding the gradients from each tied parameter to get the gradient for the tied parameter. Learning converges much faster (Fig. 6). For the remainder of this paper, we use convolutional logistic policies learned by policy gradient with weight-sharing. We learn similar control laws regardless of initial parameters and visual array size, and so the approach seems robust to local minima in parameter space.

[2]For a 7×7 visual array, the number of free parameters is reduced from 2401 to 28.

D. Eye Movement Learning Experiments

To compute policies, we used a time-horizon T that was the same as the number of states N; the reward went to 0 long before T, approximating undiscounted infinite horizon. The parameter β was 0.75, γ was 0.02, and gradients were pooled across 150 episodes per epoch. We manipulated the following.

- **Size of visual array**: The visual array size was 7×7 or 11×11, with $N = 49$ and $N = 121$, respectively.
- **Reward Function**: We compared the Infomax reward function with that postulated in Saliency literature [18].
- **Visual System Properties**: In addition to using an FPOC from psychophysical data [14], we studied what would happen in systems with different FPOCs.

Our results were analyzed in two ways.

- **Performance**: Performance was measured as "% Correct on an N-Alternative Forced Choice task (N-AFC)". That is, in an 11×11 visual array, if the location of the target had higher belief than all 120 other locations, the agent was right, otherwise it was wrong.
- **Control Law**: A policy is defined by a convolution kernel. If the kernel has a high value at eccentricity e, the agent wants to look toward some location k when there are high beliefs at locations e units away from k. If the kernel has a negative value at eccentricity e, the agent wants to look away from location k if there are high beliefs at locations e units away from k.

E. Results: Performance & Policy

We first compared three policies that we expected to perform well.

1) **Learned Infomax**: A convolutional policy, learned from experiences with information as a reinforcing signal, as described above.
2) **Percent-Correct-Greedy**: Choose the action that yields the highest expected-percent-correct after the observation, i.e., that maximizes $R_{t+1} = \max_i E[B^i_{t+1}]$ (proposed in [14]). Computing a single action from this policy is $O(KN^3)$, where N is the size of the visual array and K is a very large constant. Because of the difficulty in computing this policy for each action, we used small 7×7 visual arrays.
3) **Fixate Target Greedy**: Choose the action k that maximizes the immediate chance of looking directly at the target. This policy is implicit in visual salience models like [3], [11].

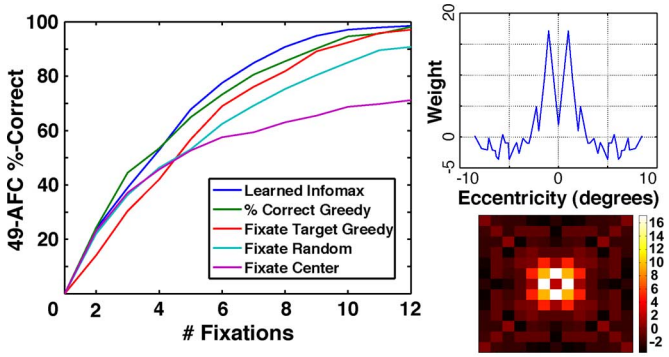
We also evaluated the performance of two policies that we expected to perform poorly.

1) **Fixate Random Locations**.
2) **Fixate Center of Visual Array**: The eye remains fixed in the central location and never moves. This policy discovers targets in the foveal region quickly, in the parafoveal region slowly, and in the peripheral region never.

The experimental conditions were simulated search tasks using the statistical model presented above, of which Fig. 5 is a typical example. On each trial, the target was moved to a new location, hidden from the searcher. In all, each location was chosen exactly 100 times. The size of the visual array was 11×11 or 7×7, depending on the experiment. For the latter,

TABLE I
FIXATIONS TO REACH 90% CORRECT (49-AFC)

Learned Infomax	% Cor. Greedy	Fixate Target	Fixate Random
7.86	8.96	9.25	11.33



(a) (b)

Fig. 7. (a) The Learned Policy performs better than four alternative policies described in Section IV-E. Policy "% Correct Greedy", proposed in [14], outperforms the learned policy in only the first four fixations. This reflects the classic tradeoff between greedy and long-term planning. (b) The "receptive field" of the learned policy. *Top*: 1-D kernel function that was learned: The learned strategy looks *next to* places of high probability. *Bottom*: Rotating this kernel radially gives the radially symmetric 2-D convolution filter that defines the policy. (a) Comparison of policies; (b) learned policy.

there were 4900 total evaluation trials for each policy, and for the former there were 12 100.

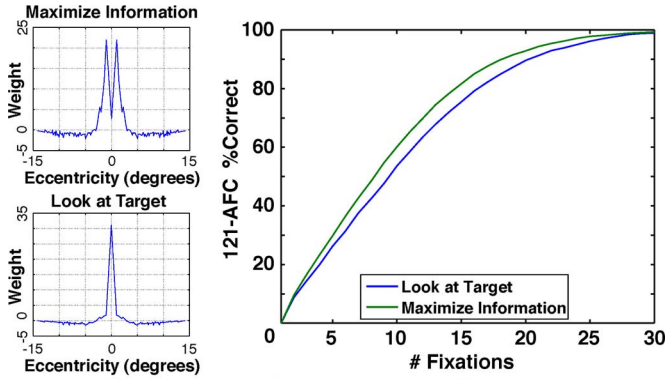
The learned Infomax optimal controller reached high levels of accuracy (90% correct on the 49-AFC task) about 1.1 fixations earlier than the Percent-Correct-Greedy policy and about 3.5 fixations earlier than the Random policy (Table I). The performance of all policies is shown in Fig. 7(a).

The policy that achieves this high performance is visualized in Fig. 7(b). Interestingly, this policy chooses to foveate *next to*, but *not at* locations where the target is likely to be. This appears to ensure that the target remains in the foveal region, while gathering extra information about the periphery. It is improper to claim that the learned policy avoids looking directly at the target—the target location is unknown. Rather, plausible target locations are kept at the edge of the fovea. Especially during the first eye movements, these are only weak hypotheses, and usually turn out to be wrong. By keeping weak but plausible target locations at the edge of the fovea, the agent is able to confirm them if they turn out to be correct, while simultaneously testing many alternate hypotheses if the current plausible hypotheses turn out to be wrong.

F. Results: Comparison to Previous Approaches

The control law that optimizes the infomax reward function avoids looking directly at plausible target locations, preferring to look just to the side of them. It is commonly assumed that ideal searchers should directly fixate locations most likely to contain the search target. Such a strategy turns out to be suboptimal when more than one eye movement is possible. How much benefit does the ideal controller get by avoiding looking directly at the target?

When we evaluated the "Fixate Target" strategy previously, we did so in a greedy way after the fashion of the salience literature. In order to be more fair to this strategy, we trained a controller that could maximize its long-term probability of looking at the target. It was given reward of 1 for looking directly at the target and 0 otherwise. Since the controller did not have direct access to the state, it received expected reward based on its belief state after the fashion of POMDPs [35], and so was linear in the belief state. This reward was the probability that it was looking at the target, $R_t = B_t^k$, where $A_t = k$.



(a) (b)

Fig. 8. Performance loss from directly fixating the target; the visual array is 11 × 11. (a) Learned "receptive fields." *Top*: The Infomax policy closely resembles the policy in Fig. 7(b) which was trained on a smaller visual array. *Bottom*: A different policy is learned when the goal is to look directly at the target. (b) Maximizing information performs noticeably better than trying to look directly at the target.

We trained Infomax and fixate target controllers on an 11×11 visual array I-POMDP. The learned control laws are visualized in Fig. 8(a). The shape of the Infomax control law is similar to that of the 7×7 task, preferring to look next to the target. This indicates that the ideal strategy remains constant with problem size. The ideal fixate target strategy looks very similar to an impulse response, and so is very similar to the greedy fixate target strategy in the previous section. Fig. 8(b) indicates that this is a reasonable, but suboptimal strategy. Controllers optimized to fixate target require 20 fixations to reach 90% accuracy on a 121-AFC tasks, while those optimizing information-gain require 18 fixations.

This quantifies the expected performance boost achievable over previous Saliency approaches in robots [11], which attempted to look at search targets. Instead, our results suggest that a better strategy is to look *near but not at* visual targets. This presents avenues for psychophysical study, to see whether indeed people prefer to look near but not at visual targets.

G. Dependence on Visual System

So far, we showed that information is a sufficient reinforcing signal to learn highly effective looking behavior from experience searching for targets. However, this was done using a single example model of uncertainty, the FPOC curve shown in Fig. 5. Is it possible that the resulting looking behavior somehow generalizes to all eyes? Or is it necessary to take into account the specific uncertainty characteristics of each system in planning optimal eye movement strategies?

The I-POMDP framework allows us to investigate how an ideal oculomotor law may change if the FPOC of the sensory mechanism changes. This question is relevant to roboticists be-

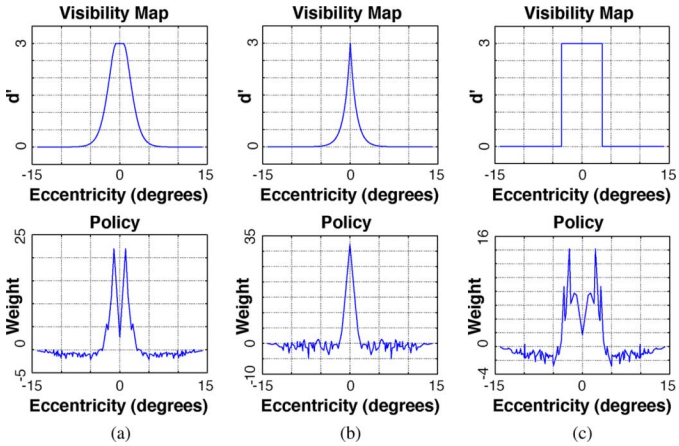


Fig. 9. Optimal policies (bottom) given different FPOCs (top). The visual array is 11 × 11. Each policy is the average of the parameters of 10 learned policies. (a) FPOC based on human data from [11], which was used in this paper's previous experiments. (b) Exponential falloff of acuity. In this case, looking next to the target does not give reliable information about its presence, and so the learned policy prefers to look directly at the target. (c) A camera can locate objects reliably in its field of view, but not outside. The learned policy attempts to keep the object toward the edge of its field of view.

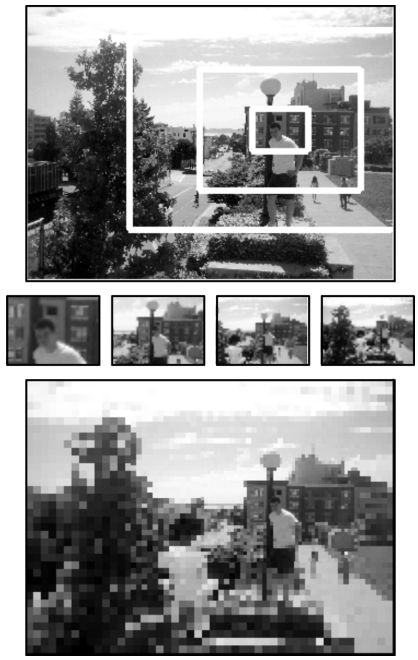


Fig. 10. Digital fovea: Several concentric IPs (*Top*) are arranged around a point of fixation. The image portions contained within each rectangle are reduced to a common size (*Middle*). In a reconstruction from the downsampled images, detail is preserved around the fixation point, but decreases with eccentricity (*Bottom*).

cause robotic cameras do not typically have the same properties as a human eye. The question is also relevant to developmental scientists and clinicians that may study the development of visual search in infants and in adults with clinical eye conditions.

Here we considered two additional FPOCs. One is an exponential function that is sparser than the human FPOC: it has a sharp initial fall-off of acuity, but then has slightly higher acuity in the periphery [Fig. 9(b)]. The other is modeled after a standard camera with uniform acuity throughout its entire visual sensor and none elsewhere, resulting in a step-function FPOC [Fig. 9(c)].

The resulting control laws are strikingly different from the original [Fig. 9(a)], suggesting that the ideal visual search strategy depends heavily on the specific FPOC of the visual system. This provides a warning against the usefulness of models of visual search derived from typical adult humans when trying to make claims about how infants, robots, or adults with certain visual disorders should move their eyes.

V. Creating & Controlling a Digital Eye

Detecting objects quickly and at low computational cost is important for a wide variety of domains, such as security applications, traffic analysis, clinical diagnosis, satellite image processing, and robotics. While progress in recent years has been dramatic, there are still two challenging cases: 1) physical scanning of scenes using active cameras; and 2) digital scanning of very large images. Scanning very large images can be seen as a special case of scanning world scenes. Thus, it is reasonable to expect that the approaches that biology has found useful for scanning the world may also be useful for scanning high resolution images.

However, the results from Section IV-G caution us to be deliberate and thoroughly characterize any system that we build to attempt to follow a biologically inspired path. In this section, we consider how the lessons we learned from studying visual search in the context of human vision can be effectively applied to make a computer program that can learn to become more efficient at a similar task.

As in the previous case, the main challenge will be to be explicit about what the informational consequence of each eye movement is. What does it mean for a computer program to "look at" a part of an image? We explore this idea by digitally simulating in software a foveal camera. The sequential placement of the digital fovea is then controlled using a policy designed to maximize the information gathered about the location of the target of interest.

The proposed approach is plug-and-play: it can be applied to standard object detectors in a modular manner. The visual search program that we present eventually learns to search scenes twice as fast as the object detection algorithms commonly used in practice. In this section, we mainly focus on finding a single face in a static image, but the model extends easily to searching for and tracking a moving face in a dynamic video, which we briefly discuss. The source code needed to reproduce the results in this section are provided online as part of Nick's Machine Perception Toolbox [38].

A. A Digital Eye

Key to the proposed approach is the idea of scanning images using a simulated fovea, which is created by cropping and scaling the image several times around a central fixation point, yielding pyramid of Image Patches (IPs) [39] (see Fig. 10). Each IP is then shrunk to a common reference size that is much smaller than the original image, typically 1/100th of the size. These different patches will lose information about the image in different ways. Large IPs may cover most of the image, but they will lose resolution when scaled down, so they will only contain

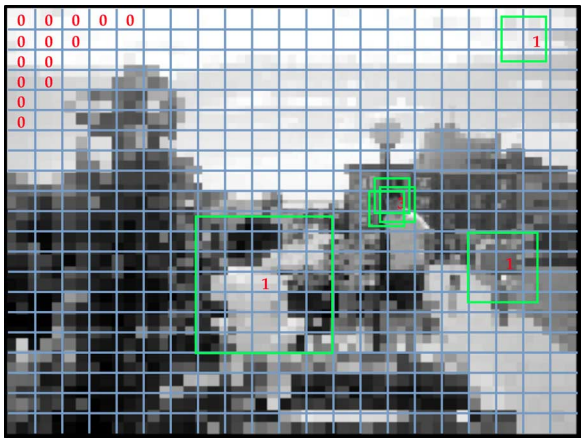


Fig. 11. Object detector returns candidate locations of the search target. In each grid cell, we count the candidates up to some maximum (above, empty cells have an observation of "0"). We model the counts as being generated by independent draws from many multinomial distributions, with parameters that vary with the distance to the point of fixation, and also whether the search target is actually centered at that grid cell.

information about low spatial frequencies. Small IPs will maintain resolution and high spatial frequency information, but only around a small region of the image.

Fig. 10 shows an example of the digital fovea at work. In this case, we used four IPs per fixation, operating at four scales. To search for the target object at that fixation point, we can apply any off-the-shelf object detection algorithm to each of these IPs. The object detector will search each of the IPs exhaustively for the target object. As long as the scaled size of the IPs is small, this exhaustive search will be quick.

For example, if any IP is scaled to 10% of the height and width of the image, its area is 1% of the original image. Since all four IPs are shrunk to the same small size, an object detector with linear complexity will search all four IPs in 4% of the time it would take to search the whole image. If the search target's location can be inferred after scanning IPs at fewer than 25 successive fixations, this foveated approach will be faster than exhaustively applying object detection to a high resolution image.[3]

B. The Multinomial I-POMDP Model

In I-POMDP, the wavelet search target could be located in one of N discrete locations, arranged in a grid. This grid formed the basis for the state space, the action space, and the observation vector.

To reproduce this behavior in the digital eye, we cover the image with a grid, and assume that the location of the object's center is inside one of those grid locations. A natural tradeoff arises in choosing how fine to make the grid: A finer grid groups fewer pixels into each grid cell, improving the ability to localize the object in the image; but this increases the number of hypotheses that must be entertained and locations that can be searched. This discretization can be seen in Fig. 11. Depending on the size of the image, more or fewer pixels may be grouped into each grid cell. This allows us to have the same state and action space as our previous investigations.

[3]This is a simple illustration and assumes no overhead for inference and planning. In practice, the break-even point will be slightly lower.

A probabilistic model of observations and how they are generated is important for deducing the target location with Bayesian inference, and for quantifying information. A major challenge for the digital eye is how to turn the output of the object detector into a suitable observation vector.

We treat object detectors as black-box algorithms that take an image as input, and output a list of pixels that are likely to be the centers of the search target. These detectors often fire in clusters around the object (hits), but also have false alarms, misses, and correct rejections (Fig. 11).

We generate the observation from the total number of objects returned by the object detector in each grid cell (up to some maximum count value, $c_{\max}$), after searching all IPs. The observation vector generated is $\vec{O}_t \in \{0, 1, \ldots, c_{\max}\}^N$.

Because information is lost in the digital eye, there is uncertainty about whether the object detector will find the object (false negative); given that an object detector finds an object, it is uncertain whether this is actually the object (false positive). We represent this uncertainty by modeling the generation of each grid cell's contribution to the observation vector as an independent draw from a different multinomial distribution conditioned on: 1) the presence or absence of an object in that grid cell; 2) the distance (x-distance and y-distance) to the center of fixation from that grid cell. Practically, this means for an $L \times M$ grid of target locations, each observation is drawn from one of $2LM$ multinomial distributions with different parameters for each combination of $\mathrm{x-distance} \in [0, 1, \ldots, M-1]$, $\mathrm{y-distance} \in [0, 1, \ldots, L-1]$, and object presence/absence. We refer to the I-POMDP with this modified multinomial observation model as the multinomial I-POMDP (MI-POMDP).

In images, the target we are searching for does not move, and the POMDP belief update equation in (12) can be used. In active cameras or video streams, the target might move between each fixation. In this case, the dynamics are modeled by $p(S_t = i | S_{t-1} = h)$, and the belief update becomes

$$B_t^i \propto \frac{p\left(o_t^i | S_t = i, A_t = j\right)}{p\left(o_t^i | S_t \neq i, A_t = j\right)} \sum_{h=1}^{N} p(S_t = i | S_{t-1} = h) B_{t-1}^h. \tag{20}$$

C. Fitting the Multinomial Observation Model

In order to estimate the information properties of the digital eye, we had the eye scan each grid location in a database of images with known face location, and measured its performance in terms of hits, misses, correct rejections, and false alarms at each possible distance from a known face location.

The image dataset contained 3500 images in which faces were present in equal amounts across all scales. Specifically, one fifth were $<10\%$ of the image major axis, and one fifth each were 10–20%, 20–30%, 30–40%, and $40\%+$ of the image major axis. The full images varied in size from 104×120 to 972×477 with an average size of 225×243. This data set is freely available as the size-scale normalized subset (GENKI-SZSL) of the GENKI dataset [40].

The observation model presented above consists of $2LM$ multinomial distributions, each with $c_{\max} + 1$ differently weighted outcomes. To fit the model, we estimated the weights for each outcome for each distribution, using $c_{\max} = 9$.

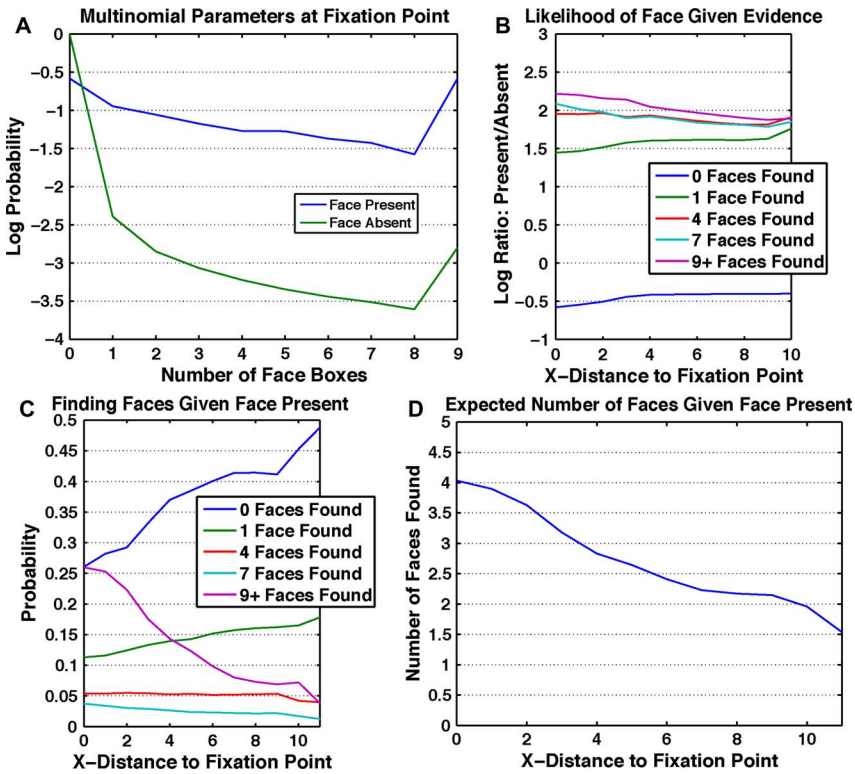


Fig. 12. Parameters of the multinomial observation model inferred from data: A: Probability of counting 0, 1, . . . faces at the point of fixation if the face is there, and if it is not there. (In A&C, boundary effects can be seen where all observations of size nine and greater are binned together.) B: Relative likelihood that a face is located N grid cells from the point of fixation, given that M face boxes were observed there. C: Probability of seeing M face boxes at a location N grid cells away from fixation, if the face is located there. D: Mean number of face boxes N grid cells away from fixation if the face is located there.

We started with a $2 \times 21 \times 21 \times 10$ table T filled with ones. For each image in the dataset, we fixated the digital fovea on every grid point k, and computed C, the count of found face boxes centered in each grid cell up to $C_{\max} = 9$. On each fixation, for each of the 440 locations j without a face, we computed $|dist_x(j,k)|$ and $|dist_y(j,k)|$, the absolute x- and y-distance from that location to the point of fixation, and incremented the table element $T[0, |dist_x(j,k)|, |dist_y(j,k)|, c]$. For the one location i with a face, we incremented the table element $T[1, |dist_x(i,k)|, |dist_y(i,k)|, c]$.

After this procedure, the estimates

$$P(O^j = c | S \neq j, A = k) = \frac{T\left[0, |dist_x(j,k)|, |dist_y(j,k)|, c\right]}{\sum_{c'=0}^{c_{\max}} T\left[0, |dist_x(j,k)|, |dist_y(j,k)|, c'\right]} \quad (21)$$

$$P(O^i = c | S = i, A = k) = \frac{T\left[1, |dist_x(i,k)|, |dist_y(i,k)|, c\right]}{\sum_{c'=0}^{c_{\max}} T\left[0, |dist_x(i,k)|, |dist_y(i,k)|, c'\right]} \quad (22)$$

correspond to the Bayesian MAP parameter estimates of the multinomial parameters, starting with a uniform Dirichlet conjugate prior [41].

Fig. 12 shows a subset of the parameters that we fit using our entire image data set. The average number of face boxes found decreases with the face's distance to the digital fovea, showing that the face is harder to find. When there is no face, it is more likely that the face finder gives zero face counts than if there is a face. Smaller numbers of face boxes are more likely than larger numbers regardless of whether there is a face. These results indicate that MI-POMDP matches our intuition about a foveated digital eye.

D. Comparison to Other Multiresolution Approaches

The search strategies proposed here relate to recent work on optimal image search, like efficient subwindow search (ESS) [42]. Our approach is data driven and detector independent, where the ESS approach is more analytic. We chose Viola Jones as a backend algorithm because it is standard, and freely available to all researchers. However, any object detector can be used. The cost of this flexibility is that our approach requires a dataset of labeled images to build a statistical model of the performance of a given object detector.

Algorithms like ESS are more restrictive on the object detector that they encapsulate, so they are not plug-and-play. Specifically, they require an upper bounding function f that must be constructed analytically for each family of object detectors for the guarantees of the algorithm to hold. Only some object detectors are amenable to such a construction. The efficiency of the ESS algorithm depends on the tightness of the upper bound that f computes and the computational overhead of evaluating f.

As in ESS, if it is known that there is more than one face in the image, our algorithm will find and report the location of one of them. As in ESS, we can search for subsequent faces by removing the location of the face we just found from consideration, and repeating the search process.

E. Implementation Details

The MI-POMDP model is framed in general formalisms that are agnostic to the object being searched for, or for the detector given. We tested it with the OpenCV 1.0 face detector, a Viola-Jones style face detector [43], [44]. For this paper, we chose to tile all images with a 21 × 21 grid, meaning the face could be localized to any of 441 locations.[4] We used IPs with diameters of 3, 9, 15, and 21 grid-cells. When the smallest IP was smaller than 60 × 45 pixels, it was not used. The downsampled image size was always the same number of pixels as the smallest IP used. The full source code needed to implement this model is provided online as part of Nick's Machine Perception Toolbox [38].

In the previous section, we fit the 8820 parameters of the multinomial detector output model to our full dataset of images. In this and following sections, all results were gathered using seven-fold cross-validation. The images were randomly assigned to seven groups of 500 images. In each fold, six groups were used to fit the multinomial parameters, and one group was used to test performance. All performance results were averaged by repeating this procedure across all seven folds. All timing experiments were done on Quad-Core Intel Xeon processors at 2.8 GHz . Absolute (wall clock) time was used, with a precision of one μs. Timing of each approach includes all the computation needed for those approaches. For MI-POMDP, this includes the time needed for image cropping and downsizing, object detection, inference, and control.

F. Default Performance

The OpenCV 1.0 Viola–Jones Face Finding implementation has a performance parameter that controls how it searches across scales for faces. Using the default scaling parameter of 1.1, we evaluated the difference in runtime and accuracy for applying Viola–Jones to a whole image, and for using multinomial I-POMDP, which calls Viola–Jones as a subroutine.

To plan fixations in a way that gathered information close to optimally, we used a convolutional logistic policy, as above in (19). We used a heuristic stopping criterion of the first repeated fixation. The maximum a-posteriori face location was then returned as the face location.

Even when there is one face image, the Viola–Jones approach generates many face boxes, both from false alarms, and multiple detections of the true face. To measure performance, we must make a single decision about the face location from this profusion of face boxes. One possibility is to take the center of mass of all boxes, but this may give a result that is close to none of the proposed locations. The approach we took was to count the number of face boxes centered in each grid cell, and take the grid cell with the highest count as the face location.

For both approaches, we measured error as the Euclidean grid-cell distance from the returned face and its true location. Fig. 13 shows an example of the algorithm in action. In this case, the final estimation of the face location is one grid-cell diagonal from the labeled location, giving a Euclidean distance error of 1.4.

[4]Anecdotally, we did not notice variation in performance with somewhat finer and coarser grids

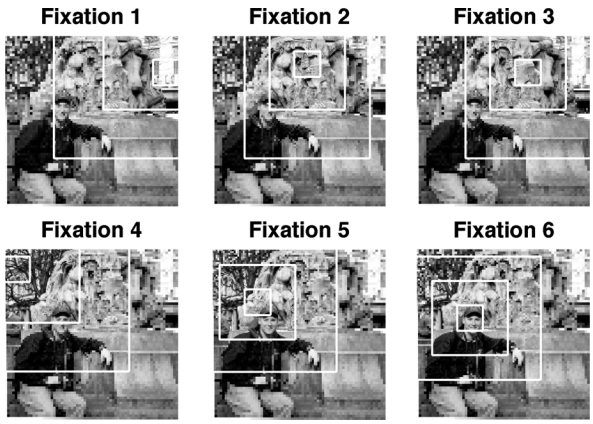


Fig. 13. Successive fixation choices by the MI-POMDP policy. The face is found in six fixations. The final estimation of the face location is one grid-cell diagonal from the labeled location, giving a Euclidean distance error of 1.4 grid-cells.

TABLE II
MI-POMDP DOUBLES THE SPEED OF VIOLA–JONES WITH A SMALL DECREASE IN ACCURACY

Measure	MI-POMDP	Viola Jones
Mean Runtime (ms)	**37.9**	73.4
Scaling (ms/1000px)	**0.57**	1.25
Error (grid-cells)	1.59	**1.26**

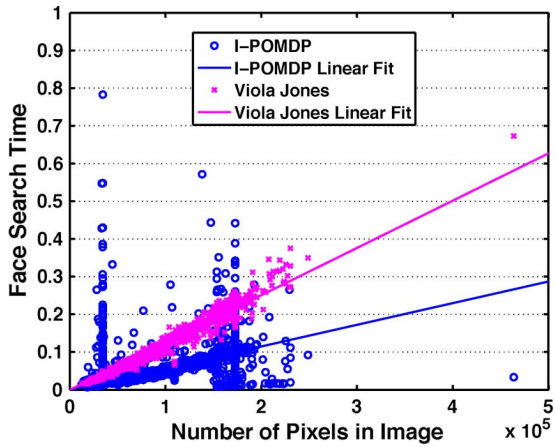


Fig. 14. Time needed to search for faces, as a function of image size. A mode of the dataset image sizes was 180 × 190 (2300/3500 images), explaining apparent spike at 34 000 pixels. Similar modes explain the other spikes.

The runtime of both algorithms as a function of image size is shown in Fig. 14. The runtime needed for Viola–Jones is empirically linear in the number of image pixels. On our computers, it took about 1.25 ms per 1000 pixels to analyze a given image. MI-POMDP is more variable. Mostly it was linear, taking .57 ms per 1000 pixels to analyze a given image (a 2.18× speed-up). Sometimes it was very quick—much quicker than this. For a few images it was slower than Viola–Jones. However, on average the real speedup (including every sub process of our algorithm) was about two-fold.

This speed increase comes at the price of a small decrease in accuracy, as shown in the Table II. Both methods on average placed the face between one and two grid-cells off the true face location.

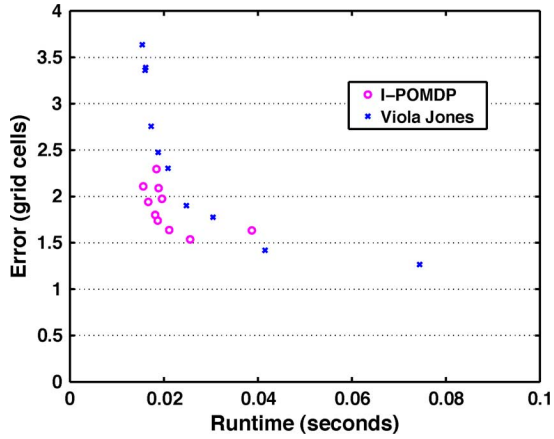


Fig. 15. By changing the Viola–Jones scaling factor, both Viola–Jones and I-POMDP become faster and less accurate. MI-POMDP is usually closer to the origin on a time-error curve, showing that it gives a better speed-accuracy tradeoff than just applying Viola–Jones.

G. Speed-Accuracy Tradeoff

While MI-POMDP sped up the OpenCV Face detector by a factor of two, it slightly reduced its accuracy. We thus investigated the speed-accuracy tradeoff function in OpenCV and compared it with the tradeoff provided by MI-POMDP. A speed-accuracy tradeoff function for the OpenCV classifier can be obtained by varying its scale parameter. This parameter controls the granularity of the search [43]. By default, this parameter is 1.1, but we changed it to 1.2, 1.3, ..., 2.0 and investigated the effect on speed and accuracy performance. Recall that MI-POMDP calls an object detector as a subroutine, so making that object detector faster also makes MI-POMDP faster.

Fig. 15 shows that MI-POMDP on top of a Viola–Jones style object detector gives a lower runtime for a given level of error than using Viola–Jones alone. Thus the MI-POMDP speed increase does not need to come with an accuracy tradeoff.

H. Discussion

We created a digital eye that leverages a principled model of visual search to substantially optimize the performance of generic object detectors. The computational cost added by this approach is more than compensated by the efficiency of the search. Speed ups of a factor of two can be expected with very little loss in accuracy. The approach proposed here lends itself to some natural extensions.

1) The approach is not at odds with salience-based search strategies, and in fact can be integrated with such approaches, like those taken in [45]. By leveraging the Pyramid of IPs digital fovea, salience can be computed for the foveal image representation much more quickly than for the entire image. Combined with recent fast salience methods like [11], [38], we might expect considerable gains.
2) Our digital eye is naturally parallelizable: by simulating several fixations at once, we can gather more information more quickly. By processing all IPs at once, each fixation takes less time. A challenge will be developing optimal parallel search strategies: If you have the computational resources to search 10 locations simultaneously, which 10 would give you the best long term information gathering?
3) Extension to active cameras in robots: While a parallel implementation of Viola–Jones could consider all Image Patches at once, a robot can only aim one camera at one spatial location at a time, and so it has a rigid informational bottleneck. The challenges in this extension will be in maintaining a reliable mapping from image coordinates to world coordinates, and in evaluating the foveal properties (fitting a multinomial observation model) for the robot's particular vision system.
4) More sophisticated system dynamics can be applied to search through high resolution video streams. Since the location of an object changes only a little bit frame to frame, inferences made in one frame are very informative for the next. Rather than searching the whole image for the target, we can apply one digital fixation to a frame and make inferences about where the target is (and is not) located. Since only one fixation is needed per frame, the per-image runtime will be much faster than in the current approach. While the object will not be correctly localized in every frame, once it is found, it can be easily tracked. We have already begun to explore this approach to object detection in high definition video, although at time of writing we have not quantified it thoroughly.

VI. On the Role of Learning and Development

The main focus of this paper was the computational analysis of eye movements. This involved formulating eye motion as a problem in stochastic optimal control and analyzing the type of solutions one finds under idealized models of the eyes. We showed that information gain can be a very powerful reward signal to develop efficient visual search policies. We also showed how these policies change as a function of some key characteristics of the visual sensory system. We showed that the approach could be used to engineer versatile and useful object search algorithms.

While our work elucidated the computational limitations of current saliency models of eye movement, namely the fact that they are not sufficiently specified to be considered valid optimality models, our own models are likewise too idealized. They ignore critical sources of uncertainty. For example, we assumed that the eyes move instantaneously, and with perfect fidelity. In real organisms and engineered systems, this is not the case. For example, for physical robot like the Einstein Robot (Fig. 16), there are important sources of uncertainty that cannot be ignored;

1) the relation between servo commands and motion of pixels across the retina;
2) the time course from execution to completion of an eye movement;
3) the size of the robot's instantaneous field of view (visual angle), relative to its total field of view, from one limit of its eye movement to the other;
4) the quality of image frames collected during an eye movement;
5) the likelihood that objects in the robot's environment will move spontaneously.

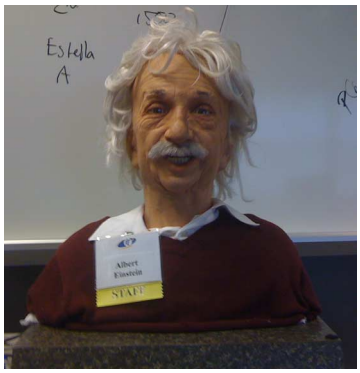


Fig. 16. Einstein robot.

Each of these parameters, and their associated uncertainties, must be quantified in order to better understand the problems faced by the brain when scheduling eye movements.

This brings an even more important issue. The computational models we investigated here assumed that we have characterized sensorimotor interaction, body morphology, and the statistical regularities and information structure they induce, as in [14] and [46]. In our case, in order to develop an optimal policy we first had to develop quantitative models of the properties of the sensory motor system. Only after we knew, for example, the probability distribution of the observations given actions and states, could we formulate an information based reward signal. This makes sense when the goal is to understand the visual search policies observed in organisms. A computational analysis of the type performed in this paper helps us get a better sense of why humans move their eyes the way they do and why we may want robots that move their eyes differently. However the analysis also raises important developmental questions: How do organisms acquire the knowledge of their own sensory motor systems that would be needed to develop optimal policies?

Organisms cannot construct their world and bodies to have desirable mathematical characteristics. Most importantly they do not have access to objective truth with which to characterize the uncertainties in their world and bodies. All their knowledge is "subjective" in the sense that it is mediated by their own sensors and inference mechanisms. How can humans, computer programs, and robots characterize uncertainties subjectively? One place to start is Sutton's verification principle [47]:

> *"An AI system can create and maintain knowledge only to the extent that it can verify that knowledge itself."*

An important area of future development for infomax models will be using simple statistical relations among sensors and actuators, and their evolution over time, to infer the above quantities. For example, simple low level cues like optical flow can be used to characterize both external motion distributions (how objects in the world are likely to move) and internal motion distributions (the time course of pixels moving across the retina after issuing a servo command).

Following optical flow approaches, a robot can compute frame-differences in pixel position across a trajectory of frames collected after an eye movement command is issued. In ongoing work, the Einstein robot was able to use this technique to determine that his servos do not start to move until about 200 ms after a motor command is requested, that there is rapid movement from about 200–300 ms, and small jitter and position refinements until about 500 ms.

Understanding the problems faced by organisms is key to ultimately understanding the solutions biology has chosen, and also for engineering intelligent systems such as robots. An example of understanding the choices of biology is why we have many different types of eye movements (smooth pursuit, saccades, vestibular stabilization, optokinetic stabilization). Characterizing the uncertainties inherent in eye movement may help us understand each type of eye movement in terms of its information costs and benefits. If Einstein takes 500 ms to complete a saccade, what is the information tradeoff between keeping his eyes stationary and receiving diminishing information returns on the things he already sees versus moving his eyes and sacrificing information now for new information later? Can he quantify the information cost-and-benefit of a saccade?

In infomax approaches, information is a fundamental currency that can be used to analyze tradeoffs faced in biological systems like the decision to saccade or use smooth pursuit. Infomax approaches make the role of each eye movement explicit in terms of its ability to decrease our uncertainty about relevant questions in the world. Finally, they give us a framework for building intelligent artificial systems that explore as quickly as possible, leading to faster machine perception.

APPENDIX

1) General Policy Gradients: Ultimately, in all policy gradient methods, the quantity we care about is

$$\frac{\nabla_\theta p(b_{t+1}|b_t,\theta)}{p(b_{t+1}|b_t,\theta)}$$

where

$$p(b_{t+1}|b_t,\theta)=\sum_O\sum_A p(b_{t+1}|o_t,a_t,b_t)p(o_t|a_t,b_t)p(a_t|b_t,\theta).$$

Note that the belief update is deterministic, and so $p(b_{t+1}|o_t,a_t,b_t)$ always 1 or 0. Let $\hat{O}_a$ be the set of observations that cause a state transition to b_{t+1} from state b_t under action a. Then

$$p(b_{t+1}|b_t,\theta)=\sum_{a_t\in A}\sum_{o_t\in\hat{O}_a} p(o_t|a_t,b_t)p(a_t|b_t,\theta)$$
$$\nabla_\theta p(b_t|b_t,\theta)=\sum_{a_t\in A}\sum_{o_t\in\hat{O}_a} p(o_t|a_t,b_t)\nabla_\theta p(a_t|b_t,\theta).$$

Thus, the policy gradient update rule can be written as

$$\frac{\nabla_\theta p(b_{t+1}|b_t,\theta)}{p(b_{t+1}|b_t,\theta)}=\frac{\sum_{a_t\in A}\sum_{o_t\in\hat{O}_a} p(o_t|a_t,b_t,\theta)\nabla_\theta p(a_t|b_t,\theta)}{\sum_{a_t\in A}\sum_{o_t\in\hat{O}_a} p(o_t|a_t,b_t,\theta)p(a_t|b_t,\theta)}.$$

The only extra information beyond the POMDP model that is required to make policy gradient updates is the gradient of the action probabilities with respect to the parameters.

If we assume that each new belief state can be reached by exactly one observation (this is often not true), which is the observation we've just made, then the set $\hat{O}_a$ is empty for all o_t other than the observation we actually just made, obviating the

inner sum over the possibly large number of observations. Then we have

$$\frac{\nabla_\theta p(b_{t+1}|b_t,\theta)}{p(b_{t+1}|b_t,\theta)} = \frac{\sum_{a_t\in A} p(o_t|a_t,b_t)\nabla_\theta p(a_t|b_t,\theta)1(o_t\in\hat{O}_a)}{\sum_{a_t\in A} p(o_t|a_t,b_t)p(a_t|b_t,\theta)1(o_t\in\hat{O}_a)}$$

where $1(o \in \hat{O}_a)$ simply indicates whether it would be possible for the same belief update to occur if observation o were made under a different action than the one we just saw. Even without the above assumption, this gradient should be *on average* correct, because each observation is made with the correct probability, and so over time the correct weighting will be given to each observation. However, computing the full sum gives a better, less variable estimate of the true gradient.

2) Gradients in Logistic Policies: In this section, we consider policies that are logistic mappings from continuous belief states to discrete action multinomial probabilities. Specifically, we have

$$\begin{aligned} p(A_t = k|b_t,\theta) &= \frac{\exp(\theta^k \cdot b_t)}{\sum_{j=1}^{M}\exp(\theta^j \cdot b_t)} \\ &= \frac{1}{1+\sum_{j\neq k}\frac{\exp(\theta^j\cdot b_t)}{\exp(\theta^k\cdot b_t)}} \\ &= \left[1+\exp(-\theta^k\cdot b_t)C_k\right]^{-1} \\ &\quad C_k \stackrel{\text{def}}{=} \sum_{j\neq k}\exp(\theta^j\cdot b_t) \qquad (23) \\ &= \left[1+K_k\exp(\theta^j\cdot b_t)+C_j\right]^{-1} \\ &\quad K_k \stackrel{\text{def}}{=} \exp(-\theta^k\cdot b_t) \\ &\quad C_j \stackrel{\text{def}}{=} K_k\sum_{i\neq k, i\neq j}\exp(\theta^i\cdot b_t) \qquad (24) \end{aligned}$$

where C_k is constant with respect to k, and C_j is constant with respect to both j, which is useful for computing derivatives. For this logistic formulation, the derivative with respect to the weights i leading to the chosen action k, i.e., the element (k,i) of the parameter matrix θ, can be written as

$$\begin{aligned} &\nabla_{\theta^{k,i}} p(A_t = k|b_t,\theta) \\ &\quad = \nabla_{\theta^{k,i}}\left[1+\exp(-\theta^k\cdot b_t)C_k\right]^{-1} \\ &\quad = b_t^i\left[1+\exp(-\theta^k\cdot b_t)C_i\right]^{-2}\exp(-\theta^k\cdot b_t)C_i \\ &\quad = b_t^i\left[1+\exp(-\theta^k\cdot b_t)C_k\right]^{-1} \\ &\qquad \times\left[1-\left[1+\exp(-\theta^k\cdot b_t)C_k\right]^{-1}\right] \\ &\quad = b_t^i p(A=k|b_t,\theta)\left[1-p(A_t=k|b_t,\theta)\right] \qquad (25) \end{aligned}$$

By a very similar argument, for the rows j of the parameter matrix θ that are not associated with action k, i.e. $j \neq k$, the derivative can be written as

$$\begin{aligned} \nabla_{\theta^{j,i}} p(A=k|b_t,\theta) &= \nabla_{\theta^{j,i}}\left[1+K_k\exp(\theta^j\cdot b_t)+C_j\right]^{-1} \\ &= -b_t^i p(A=k|b_t,\theta) \\ &\quad\times\left[1-(1-C_j)p(A=k|b_t;\theta)\right]. \qquad (26) \end{aligned}$$

REFERENCES

[1] C. Harris, "Does saccadic undershoot minimize saccadic flight-time? A monte-carlo study," *Vis. Res.*, vol. 35, no. 5, pp. 691–701, 1995.

[2] W. Kienzle, F. A. Wichmann, B. Schölkopf, and M. Franz, "A nonparametric approach to bottom-up visual saliency," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007.

[3] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[4] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cogn. Psychol.*, vol. 12, no. 1, pp. 97–136, Jan. 1980.

[5] J. Wolfe, K. Cave, and S. Franzel, "Guided search: An alternative to the feature integration model for visual search," *J. Exp. Psychol.: Human Perception Perform.*, vol. 15, no. 3, pp. 419–433, 1989.

[6] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. New York: Holt, 1982.

[7] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search," *Psychol. Rev.*, vol. 113, no. 4, pp. 766–786, 2006.

[8] N. D. B. Bruce and J. K. Tsotsos, "Saliency based on information maximization," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 155–162.

[9] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007.

[10] L. Zhang, M. H. Tong, N. J. Butko, J. R. Movellan, and G. W. Cottrell, A Bayesian Framework for Dynamic Visual Saliency Attracts Attention: A Probabilistic Account of the Cross-Race Advantage in Visual Search (In Preparation), 2007.

[11] N. J. Butko, L. Zhang, G. W. Cottrell, and J. R. Movellan, "Visual saliency model for robot cameras," in *Proc. Int. Conf. Robot. Autom.*, Pasadena, CA, 2008.

[12] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2005.

[13] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2006, pp. 1–8.

[14] J. Najemnik and W. S. Geisler, "Optimal eye movement strategies in visual search," *Nature*, vol. 434, pp. 387–391, Mar. 2005.

[15] N. Sprague and D. Ballard, "Eye movements for reward maximization," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[16] A. Simpkins, R. de Callafon, and E. Todorov, "Optimale trade-off between exploration and exploitation," in *Proc. Amer. Conf. Control*, Seattle, WA, Jun. 2008, pp. 33–38.

[17] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Trans. Autonom. Mental Develop.*, vol. 2, no. 2, pp. 70–82, Jun. 2010.

[18] C. M. Kanan, M. H. Tong, L. Zhang, and G. W. Cottrell, "Sun: Top-down saliency using natural statistics," *Visual Cogn.*, vol. 17, no. 6 & 7, pp. 979–1003, 2009.

[19] N. Sam, R. Hari, O, S. Lu, and J. Simola, "Seeing speech: Visual information from lip movements modifies activity in the human auditory cortex," *Neurosci. Lett.*, vol. 127, pp. 141–145, 1991.

[20] R. Mottonen, C. M. Krause, K. Tiippana, and M. Sams, "Processing of changes in visual speech in the human auditory cortex," *Brain Res. Cogn. Brain Res.*, vol. 13, no. 3, pp. 417–25, May 2002.

[21] J. H. R. Maunsell and S. Treue, "Feature-based attention in visual cortex," *Trends Neurosci.*, vol. 29, no. 6, pp. 317–322, Jun. 2006.

[22] D. V. Lindley, "On a measure of the information provided by an experiment," *Ann. Math. Statist.*, vol. 27, pp. 985–1005, 1956.

[23] M. Stone, "Application of a measure of information to the design and comparison of regression," *Annals Math. Statist.*, vol. 30, pp. 55–70, 1959.

[24] V. V. Fedorov, *Theory of Optimal Experiments*. New York: Academic, 1972.

[25] J. M. Bernardo, "The Use of Information in the Design and Analysis of Scientific Experimentation," Ph.D. dissertation, Univ. London, London, U.K., 1976.

[26] J. Lewi, R. Butera, and L. Paninski, "Sequential optimal design of neurophysiology experiments," *Neural Comput.*, vol. 21, no. 3, pp. 619–687, 2009.

[27] B. A. J. and T. J. Sejnowski, "Edges are the independent components of natural scenes," in *Advances in Neural Information Processng Systems*. Cambridge, MA: MIT Press, 1996, vol. 9.
[28] J. D. Nelson, J. B. Tenenbaum, and J. R. Movellan, "Active inference in concept learning," in *Proc. 23rd Annu. Conf. Cogn. Sci. Soc.*, Edinburgh, Scotland, 2001, pp. 692–697.
[29] J. R. Movellan, "An infomax controller for real time detection of contingency," in *Proc. Int. Conf. Develop. Learn.*, Osaka, Japan, 2005.
[30] N. J. Butko and J. R. Movellan, "Learning to learn," in *Proc. Int. Conf. Develop. Learn.*, London, U.K., 2007.
[31] M. Cakmak, C. Chao, and A. Thomaz, "Designing interactions for robot active learning," *IEEE Trans. Autonom. Mental Develop.*, vol. 2, no. 2, pp. 108–118, June 2010.
[32] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.
[33] S. Hoi, R. Jin, J. Zhu, and M. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. Int. Conf. Mach. Learn.*, Pittsburgh, PA, 2006, pp. 417–424.
[34] M. Handford*, Where's Waldo*. New York: Candlewick, 1987.
[35] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, pp. 99–134, 1998.
[36] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for pomdps," in *Proc. Int. Joint Conf. Artif. Intell.*, Acapulco, Mexico, 2003, pp. 1025–1032.
[37] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *J. Artif. Intell. Res.*, vol. 15, pp. 319–350, Nov. 2001.
[38] N. J. Butko, Nick's Machine Perception Toolbox [Online]. Available: http://mplab.ucsd.edu/~nick/NMPT 2008
[39] R. B. Gomes, L. M. G. Gonalves, and B. M. de Carvalho, "Real time vision for robotics using a moving fovea approach with multi resolution," in *Proc. Int. Conf. Robot. Autom.*, Pasadena, CA, May 2008.
[40] [Online]. Available: http://mplab.ucsd.eduThe MPLab GENKI Database, GENKI-SZSL Subset
[41] C. M. Bishop*, Pattern Recog. Mach. Learn.*, M. Jordan, J. Kleinberg, and B. Schölkopf, Eds. New York: Springer-Verlag, 2006.
[42] C. H. Lampert, M. B. Blaschko, and T. Hoffman, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008.
[43] [Online]. Available: http://www.cs.indiana.edu/cgi-pub/oleykin/website/OpenCVHelp/The OpenCV 1.0 API
[44] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Honolulu, HI, 2001.
[45] J. Vogel and N. de Freitas, "Target-directed attention: Sequential decision-making for gaze planning," in *Proc. Int. Conf. Robot. Automation*, Pasadena, CA, May 2008.
[46] M. Lungarella and O. Sporns, "Mapping information flow in sensorimotor networks," *PLoS Comput. Biol.*, vol. 2, no. 10, 2006.
[47] R. S. Sutton, Verification, the Key to AI [Online]. Available: http://webdocs.cs.ualberta.ca/~sutton/IncIdeas/KeytoAI.html Nov. 2001

Nicholas J. Butko received the B.Sc. degrees in computer science and cognitive science from the University of California, San Diego, in 2004. He is currently pursuing the Ph.D. degree at the University of California, San Diego.

His thesis research, conducted in the Machine Perception Laboratory (MPLab) since 2005, addresses the computational problems facing intelligent agents as they perceive, act, and interact in an uncertain and changing world. The emphasis here is to create theories that work in practice. To that end, he worked as an in intern at Intel in 2008 and 2009, to develop machine perception technologies for use in socially and physically aware perceptive tutoring systems.

Javier R. Movellan received the B.Sc. degree from the Universidad Autonoma de Madrid, Spain, in 1983, and the Ph.D. degree from the University of California, Berkeley (UC Berkeley), in 1990.

He founded the Machine Perception Laboratory (MPLab) at the University of California, San Diego (UCSD), where he is currently a research professor. Prior to his UCSD position, he was a Fullbright Scholar at UC Berkeley and a research associate at Carnegie Mellon University, Pittsburgh, PA, from 1990 to 1993. The mission of the MPLab is to learn about intelligent behavior by developing systems that operate in the uncertain, but sensory rich conditions typically faced by the brain. His research interests include machine learning, machine perception, automatic analysis of human behavior, and social robots.