UNIVERSITY OF CALIFORNIA, SAN DIEGO

Active Perception

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Cognitive Science

by

Nicholas J. Butko

Committee in charge:

Virginia de Sa, Chair
Garrison W. Cottrell
Richard J. Krauzlis
Javier R. Movellan
Terrence J. Sejnowski
Angela J. Yu

2010

The dissertation of Nicholas J. Butko is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2010

DEDICATION

To the robots. One day, you'll thank us.

EPIGRAPH

Exploring using the POMDP framework is often not such a good idea. This is because in many exploration problems, the number of unknown state variables is huge, as is the number of possible observations.... In fact, given the huge number of *possible* values for the unknown state variables in exploration, any algorithm that integrates over all possible such values will inevitably be inapplicable to high dimensional exploration problems, simply for computational reasons.

—Sebastian Thrun, Wolfram Burgard, and Dieter Fox,
Probabilistic Robotics, 2005 [1]

I do not now believe that this is at all a correct analysis of color vision or of the retina, but it showed the possible style of a correct analysis. Gone are the ad hoc programs of computer vision; gone is the restriction to a special visual miniworld; gone is any explanation *in terms of neurons*–except as a way of implementing a method. And present is a clear understanding of what is to be computed, how it is to be done, the physical assumptions on which the method is based, and some kind of analysis of algorithms that are capable of carrying it out.

—David Marr,
Vision, 1982 [2]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First I thank Javier Movellan, my mentor, my critic, and my advocate. I thank Virginia de Sa, my committee chair, for enabling me to pursue my passions. I thank my other committee members, Terry Sejnowki, Gary Cottrell, Angela Yu, and Rich Krauzlis, for their support, attention, and advice. I thank Ian Fasel, Lingyun Zhang, and Jochen Triesch, co-authors on work in this thesis.

I am thankful to the rest of my co-authors, on works that I've published that aren't included in this thesis: Ferid Bajromovic, Frank Mattern, Joachim Denzler, Andreas Wiratanaya, Michael Lyons, Shinji Abe, Georgios Theocharous, Richard Beckwith, Matthai Philipose, Tingfan Wu, Paul Ruvolo, Marni Bartlett, Nicole Schmiedt, and Vimal Mathew.

I am thankful to the remaining members of the Machine Perception Lab, Gwen Littlewort, Jacob Whitehill, Walter Talbott, Josh Susskind, Luis Palacios, Andrew Salamon, Dave Deriso.

I thank Cornelius Weber, Erik Murphy-Chutorian, and three anonymous reviewers for comments on earlier drafts of Chapter 2. I thank Matthew H. Tong for reproducing the results in Itti & Baldi's salience algorithm for Chapter 3, and the teachers, parents, and children in Room 1 of the Early Childhood Education Center for their invaluable continued support and involvement in the RUBI project.

I would like to thank those agencies that have funded myself and my collaborators for the work in this thesis, including the Hertie foundation, NSF grant ECCS 0622229, NSF IGERT training grant #DGE-0333451, UC Discovery Grant 10202, NSF Science of Learning Center grant #SBE-0542013, NIMH grant R01 MH57075, and NSF IIS INT2 0808767.

Finally, I would like to thank my closest friends and family: Cameron Chrisman, John Enquist, Jan Schellenberger, Aleksandr Simma, Edward Wu, Margaret Butko, George Butko, Barbara Butko, Emerald Butko, and Zachary Butko.

The text of Chapter 1 is almost entirely new text, although some of it was taken from the introductions of papers that comprise the chapters that follow. I

am the sole author of this chapter.

The text of Chapter 2, with some modification, is a reprint of the material as it appears in N.J. Butko and J. Triesch, "Exploring the Role of Intrinsic Plasticity for the Learning of Sensory Representations," *Neurocomputing*, 70(7-9):1130–1138 (2007) [3]. I was the primary author of this publication; the co-author supervised the research that forms the basis for this chapter.

The text of Chapter 3, with some modification, is a reprint of the material as it appears in N.J. Butko, L. Zhang, G.W. Cottrell, and J.R. Movellan, "Visual Saliency for Robot Cameras," *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, 2398–2403 (2008) [4]. I was the primary author in this publication.

The text of Chapter 4, with some modification, is a reprint of the material as it appears in N.J. Butko and J.R. Movellan, "Detecting Contingencies: An Infomax Approach," *Neural Networks* 23(8–9):973–984 (2010) [5]. Both authors shared in writing the paper. My main research role in this project was the implementation of the reinforcement learning developmental model.

The text of Chapter 5, with some modification, is a reprint of the material as it appears in N.J. Butko and J.R. Movellan, "Infomax Control of Eye Movements," *IEEE Transactions on Autonomous Mental Development*, 2(2):91–107 (2010) [6]. I was the primary author of this publication; the co-author supervised the research that forms the basis of this chapter.

The text of Chapter 6, with some modification, is a reprint of the material as it appears in N.J. Butko and J.R. Movellan, "Learning to Look," *Proceedings of the 2010 IEEE International Conference on Development and Learning*, 70–75 (2010) [7]. I was the primary author of this publication; the co-author supervised the research that forms the basis of this chapter.

The text of Chapter 7 is unpublished work, to be submitted with authors N.J. Butko, I.R. Fasel, and J.R. Movellan. I was the primary author and researcher on this project, designing the experiments, obtaining results, and drafting the manuscript. Fasel developed the software systems used, and Movellan supervised the research that forms the basis for this chapter.

VITA AND PUBLICATIONS

VITA

2010	Ph.D., Cognitive Science, University of California, San Diego
2004	B.S., Computer Science, University of California, San Diego
	B.S., Cognitive Science, University of California, San Diego

PUBLICATIONS

Butko, N.J., Movellan, J.R., "Detecting Contingencies: An Infomax Approach," *Neural Networks* 23(8–9):973–984 (2010)

Butko, N.J., Movellan, J.R., "Infomax Control of Eye Movements," *IEEE Transactions on Autonomous Mental Development, Special Issue on Active Learning and Intrinsically Motivated Exploration in Robots*, 2(2):91–107 (2010)

Butko, N.J., Movellan, J.R., "Learning to Look," *Proceedings of the 2010 IEEE International Conference on Development and Learning*, 70–75 (2010)

Theocharous, G., **Butko, N.J.**, Philipose, M., "Designing a Mathematical Manipulatives Tutoring System Using POMDPs," *Proceedings of the POMDP Practitioners' Workshop: Solving Real-world POMDP Problems, International Conference on Automated Planning and Scheduling (ICAPS)*, Toronto, May 2010.

Beckwith, R., Theocharous, G.T., Philipose, M., **Butko, N.J.**, Schmied, N., Mathew, V., "IRL: Virtual Reality and Pysical Manipulatives," *Proceedings of the Workshop on Next Generation of HCI and Education, CHI (ACM Conference on Human Factors in Computing Systems)*, Atlanta GA, April 2010

Butko, N.J., Movellan, J.R., "Optimal Scanning for Faster Object Detection," *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2751–2758 (2009)

Wu, T., **Butko, N.J.**, Ruvolo, P., Bartlett, M., Movellan, J.R., "Learning to Make Facial Expressions," *Proceedings of the 2009 IEEE International Conference on Development and Learning,* 1–6 (2009)

Theocharous, G., Beckwith, R., **Butko, N.J.**, Philipose, M., "Tractable POMDP Planning Algorithms for Optimal Teaching in SPAIS," *IJCAI Workshop on Plan, Activity, and Intent Recognition (PAIR)*, (2009)

Butko, N.J., Movellan, J.R., "IPOMDP: An Infomax Model of Eye-Movement," *Proceedings of the 2008 IEEE International Conference on Development and Learning*, 139–144 (2008)

Butko, N.J., Zhang, L., Cottrell, G.W., Movellan, J.R., "Visual Saliency for Robot Cameras," *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, 2398–2403 (2008)

Butko, N.J., "Generative & Discriminative Naive Bayes," *Technical Report*, MPLAB.UCSD-1 (2007).

Butko, N.J., Movellan, J.R., "Learning to Learn," *Proceedings of the 2007 IEEE International Conference on Development and Learning*, 151–156 (2007)

Butko, N.J., Triesch, J., "Exploring the Role of Intrinsic Plasticity for the Learning of Sensory Representations," *Neurocomputing*, 70(7-9):1130–1138 (2007)

Wiratanaya, A., Lyons, M., **Butko, N.J.**, Abe, S. "iMime: An Interactive Character Animation System for use in Dementia Care," *Intelligent User Interfaces* (2007)

Bajramovic, F., Mattern, F., **Butko, N.J.**, Denzler, J., "A Comparison of Nearest Neighbor Search Algorithms for Generic Object Recognition," *Advanced Concepts for Intelligent Vision Systems* (2006)

Butko, N.J., Triesch, J., "Exploring the Role of Intrinsic Plasticity for the Learning of Sensory Representations," *European Symposium on Artificial Neural Networks* (2006)

ABSTRACT OF THE DISSERTATION

Active Perception

by

Nicholas J. Butko
Doctor of Philosophy in Cognitive Science
University of California San Diego, 2010
Virginia de Sa, Chair

Action is indelibly tied to perception, and good perception is vital to our survival. Action helps to inform organisms about the world they inhabit, and to accrue information that aids in achieving their goals in a timely manner. In many computer vision applications, action plays no role in understanding an image; yet in natural situations, action and perception are inextricably linked across all levels. This thesis is about building machines that operate in real time in the real world, using actions to aid perception. In the chapters of this thesis, there are many examples of active perception domains, each with challenging problems. We analyze these problems mathematically, propose solutions, and evaluate their performance in real world conditions. In the process, we explore a robust mathematical foundation for understanding active perception, and develop many techniques for practical analysis.

Chapter 1

Introduction

This thesis is about active perception: how action helps to inform organisms about the world they inhabit, and how they act to accrue information to achieve their goals in a timely manner. In many computer vision applications, action plays no role in understanding an image; yet in natural situations, action and perception are inextricably linked across all levels: modulating attention enhances perception as measured by decreasing response time to saccade targets [8]; where we move our eyes has a critical effect on how we understand a scene [9]; bats vocalize to perceive the obstacles around them [10]; infants actively probe new objects to discover if they are responsive [11, 12]; expert Tetris players make many more movements than necessary to help discover the best move [13]; scientists design experiments to test hypotheses [14]; multiple sailors take different measurements and combine the output to discover the current bearing of a ship [15].

Our retinas are bombarded with thirteen thousand trillion photons every second of every minute that we are outside on a clear sunny day.[1] But this is only a small fraction of the visual data available: in a ten meter radius, there are forty-five million times as many photons. Humans make over 150,000 saccades per waking day, spending about 1.5-2 hr in saccadic flight [18]. Every second of every

[1] During daylight hours, the sun radiates 680 $\frac{\mathrm{W}}{\mathrm{m}^2}$ of light onto the earth [16]. In bright sunlight, the human pupil shrinks to 3 mm in diameter [17]. 4.8 mW of light enter the eye, *i.e.* 4.8 millijoules per second. The light, yellow on average, has a wavelength of 550 nm, oscillating at 5.5×10^{14} Hz. Each oscillation (each photon) carries with it $5.5 \times 10^{14} \times 2\pi\hbar = 3.6 \times 10^{-19}$ joules. This gives 1.3×10^{16} photons, or thirteen thousand trillion, per second.

minute of our waking lives, our brains make decisions about where to look; we decide which photons to sense in order to perceive the information we require 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.

Action is indelibly tied to perception, and good perception is vital to our survival. This thesis is about creating machines that operate in real time in the real world, using actions to help them perceive. Building systems that operate in the same conditions and timescales as humans complements other fields of cognitive science such as psychology, neuroscience, and philosophy.

On one end, building systems helps uncover hidden assumptions or important omissions in theories of intelligence. On the other, building systems that perform intelligent behaviors shows what is possible, so that no one can claim otherwise. Poverty of the stimulus claims have influenced the scientific understanding of human intelligence, arguing that certain skills must be innate because the environment does not provide a rich enough statistical structure to support learning those skills [19]. Building a machine that can learn what was previously considered to be unlearnable can have important scientific consequences.

In the chapters of this thesis, there are examples of real world, real time active perception domains, each with challenging problems. In each case, we analyze mathematically the nature of these problems, propose solutions, and explore the nature of those solutions. In this introduction, we lay out a theoretical groundwork for the chapters that follow, discuss challenging problems, and highlight contributions of this thesis.

1.1 Perception and Information

Notation:

Unless otherwise stated, capital letters are used for random variables, small letters for specific values taken by random variables. When the context makes it clear, we identify probability functions by their arguments: *e.g.*, $p(a, b)$ is short-

Table 1.1: Table of variables

X_t	State; perceptual interpretations
Y_t	Sensation; *e.g.* retinal or acoustic input
Y'_t	Representation; functional transformation of sensory input
U_t	Action; things we can do
$H_t = Y_{1:t}, U_{1:t}$	History; previous sensorimotor experiences
τ	Horizon, a window of time in the future for planning

hand for the joint probability mass or joint probability density that the random variable A takes the specific value a and the random variable B takes the value b. We use subscripted colons to indicate collections or sequences: *e.g.*, $A_{1:t} \stackrel{\text{def}}{=} \{A_1 \cdots A_t\}$. We work with discrete time stochastic processes, with the parameter $\Delta t \in \mathbb{R}$ representing the sampling period.

Perception as stochastic filtering:

Perception can been seen as a process of Bayesian inference [20]. Perceptions are beliefs about the world that we form anew, every second of every minute: whether a street is safe to cross, whether a fabric is soft or coarse, the name of the person we are talking to. These perceptions are constructed by our brains from sensations: photons bombarding the retina, vibrations of the tympanic membrane, indentations in Pacinian corpuscles. Although perceptions are formed at every moment, they are aided by a trove of sensorimotor history, the experiences of our lives up to this moment. By just seeing a piece of fabric, we know how it feels on our skin: a relic of the times we reached out to touch other cloths.

Probability theory brings the weight of mathematics to the idea of belief. Belief is a probability distribution $p(x_t \,|\, h_t)$ over the outcomes of a random variable X_t given a sensorimotor history H_t. Beliefs are about the current moment, t, and can change with new experiences; thus, we index all random variables by time. Sensorimotor experience H_t is caused by many aspects of the world; X_t is not the

entire world, but some aspect of the world that is relevant to the task at hand. Depending on the situation and the task at hand, X_t could be the safety of the street in front of us, the identity of an interlocutor, any other thing we can perceive, or any combination of such things. Sensorimotor history H_t contains relevant bits of information from everything we have sensed up until this moment, $Y_{1:t}$, and everything we have done, $U_{1:t}$. Bayes's theorem allows us to decompose $p(x_t \mid h_t)$ into two components:

$$\begin{aligned} p(x_t \mid h_t) &= \frac{p(h_t \mid x_t)\, p(x_t)}{p(h_t)} \\ \underbrace{p(x_t \mid h_t)}_{\text{belief}} &\propto \underbrace{p(h_t \mid x_t)}_{\text{likelihood}} \underbrace{p(x_t)}_{\text{prior}} \end{aligned} \tag{1.1}$$

This decomposition says that the believability of a perceptual interpretation X_t is related both to the likelihood function $p(x_t \mid h_t)$, the chances of encountering our sensorimotor experience H_t when the world is in state X_t at time t; and to the prior $p(x_t)$, how consistent that interpretation is with our prior belief [21]. This decomposition is useful in many cases. Consider perceiving the identity of an object X_t given a single image Y_t, *i.e.* $H_t = Y_t$:

1. The physics of optics and light dictate how images H_t are rendered by objects X_t; thus it is sometimes easier to mathematically formalize physics, as in $p(h_t \mid x_t)$, than belief, as in $p(x_t \mid h_t)$.

2. The object X_t renders a single image H_t, but each image H_t may have been rendered by any of several objects X_t. By weighting the possibilities by the prior $p(x_t)$, Bayes rule makes a seemingly ill-posed perceptual inference problem well-posed.

The Bayesian framework has allowed researchers to explain a wide array of perceptual phenomena, such as the influence of contrast on perceived motion direction [22], the recognition of the same object across multiple views [23], the integration of conflicting visual and haptic cues in perceiving the height of object [24], and the grouping of low level object features into the perception of coherent wholes [25].

Organisms are situated in dynamic, changing environments. They can act in order to change their environments, and to sense their environments. Given the

sheer variety of possible sensorimotor experiences, it seems wasteful for perception to require a memory H_t of everything the organism has ever sensed $Y_{1:t}$ and everything it has ever done $U_{1:t}$. Luckily, in many perceptual problems of interest, the sensorimotor history h_{t-1} helps to predict x_t only to the extent that it helps predict x_{t-1}, *i.e.* $p(x_t \mid x_{t-1}, u_t, h_{t-1}) = p(x_t \mid x_{t-1}, u_t)$, which is the Markov assumption (shown in graphical model form in Figure 1.1). Under this assumption, perception takes on a recursive form (for a derivation, see [26]):

$$\overbrace{p(x_t \mid h_t)}^{\text{belief}} \propto \overbrace{p(y_t \mid x_t, u_t)}^{\text{sensor model}} \int \overbrace{p(x_t \mid x_{t-1}, u_t)}^{\text{dynamics model}} \overbrace{p(x_{t-1} \mid h_{t-1})}^{\text{previous belief}} dx_{t-1} \tag{1.2}$$

This recursion obviates the need for an exhaustive memory. The previous perceptual inference $p(x_{t-1} \mid h_{t-1})$ is sufficient, along with the current sensorimotor information, to shape the current perceptual inference $p(x_t \mid h_t)$. This, in turn, is sufficient to shape the next perceptual inference $p(x_{t+1} \mid h_{t+1})$. The distribution $p(x_t \mid h_t)$ is variously named the filtering distribution, or the belief distribution. The process of computing $p(x_t \mid h_t)$ from y_t, u_t, and $p(x_{t-1} \mid h_{t-1})$ is known variously as stochastic filtering, or belief updating.

Active perception as stochastic optimal control

Stochastic filtering is a passive process: for any sensorimotor history, a perception is made. In contrast, perception can be an active process. Given a sensorimotor history, the organism can choose actions. These actions can change both the sensory input and the world around the organism. For example, by changing the direction of her gaze, a mother not only changes what she sees (changing Y_t), but she may trigger a gaze shift in her infant [27]. Since her infant is part of the external world, which is perceived indirectly, the mother's shift in gaze may trigger a change in X_t, depending on whether the gaze of her infant is important to the mother's current goals, and thus part of X_t. So, when making decisions about how to act, it is important to know both the expected sensory consequences and the expected physical consequences of each choice. The theory of stochastic optimal control (SOC) formalizes the question of how to act in order to achieve some goal in the best possible way.

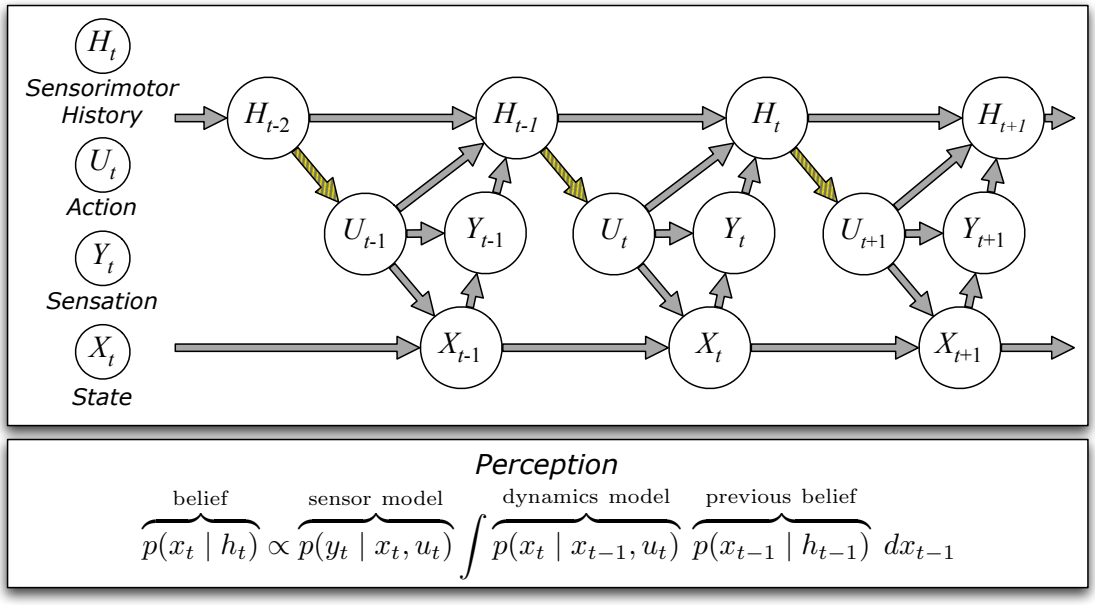


Figure 1.1: Perception as stochastic filtering. Random variables are conditionally independent of all previous variables given their parents. Striped arrows highlight the mapping from sensorimotor history to action, *i.e.* control.

In Figure 1.1, the probability distribution $p(u_t \mid h_{t-1})$ is highlighted. It represents the link between experience and decision making. As a special case, it can be a deterministic function $c_t : H_{t-1} \to U_t$, which encodes a strategy for behavior by mapping the current sensorimotor context into an action. We index the c by t because it may be useful to have different behavior strategies for different times. For example, three minutes before the end of an SAT examination, you may want to begin filling all bubbles marked "c," but this is not a winning scheme for the whole exam. A collection of such behavior strategies $c_{t:\tau}$ is variously called a control law or policy.

The utility of a control law is measured with respect to a goal, which is typically expressed as an accumulation of reward $r_t(H_t)$ over time. For example, if the goal is to fly, then $r_t(H_t)$ should be large when our sensorimotor experiences H_t indicate that we are above the ground with high probability. The reward function can describe anything that is important to the organism right now, such as reaching for an object or getting across a busy street. Sometimes, it is helpful to sacrifice

momentary rewards to better achieve long term goals. If you're in a hurry to cross a street, it's worth your while to take a moment to look for cars before crossing. Even though looking for cars delays crossing, it can be a great boon to long term well-being. Thus in SOC, the value of the current sensorimotor context H_{t-1} under a given control law $c_{t:\tau}$ is determined by the sum of expected future rewards:

$$V_t(h_{t-1} \mid c_{t:\tau}) \stackrel{\text{def}}{=} \sum_{t'=t}^{\tau} \mathrm{E}[r_{t'}(H_{t'}) \mid c_{t:\tau}, h_{t-1}] \text{ for all } h_{t-1} \tag{1.3}$$

where τ is a planning horizon. The aim of SOC is to find the optimal control law $c^*_{t:\tau}$ that maximizes the value function $V_t(h_{t-1} \mid c_{t:\tau})$ for all t, h_{t-1}. The optimal value V^*_t given by the optimal control law is

$$V^*_t(h_{t-1}) = \max_{c_{t:\tau}} V_t(h_{t-1} \mid c_{t:\tau}) \tag{1.4}$$

Finding the optimal control law is difficult because the expectation, $\mathrm{E}[\cdots]$, requires mathematical integration over all future histories: when making any decision, we must consider all possible consequences of the consequences of the consequences of our actions. Bellman discovered that this maximization has a recursive structure [28]. For all t, h_{t-1},

$$\begin{aligned} c^*_t(h_{t-1}) &= \operatorname*{argmax}_{c_t} \left[E[r_t(H_t) \mid c_t, h_{t-1}] + \max_{c_{t+1:\tau}} \sum_{t'=t+1}^{\tau} \mathrm{E}[r_{t'}(H_{t'}) \mid c_{t:\tau}, h_{t-1}] \right] \\ &= \operatorname*{argmax}_{c_t} \left[\mathrm{E}[r_t(H_t) \mid c_t, h_{t-1}] + \mathrm{E}[V^*_{t+1}(H_t) \mid h_{t-1}] \right] \end{aligned} \tag{1.5}$$

leading to the famous dynamic programming algorithm. Even with dynamic programming, this problem is intractable in many practical cases, so the field of SOC deals with many approaches to find controllers that approximately optimize the Bellman equation, Equation (1.5), in practical scenarios. Such approaches include TD-learning [29, 30], point based value iteration [31], policy gradient [32], and many more.

Approaches to control:

SOC models are prospective. They make decisions in the context of expected future sensorimotor outcomes. In our presentation so far, we have blurred

a distinction typically made between finite horizon, receding horizon, infinite horizon, and greedy control. In the finite horizon case, time starts at $t = 1$ and ends at $t = \tau$. In this case, optimal actions can be computed separately for each time point in the episode. As in taking a test like the SAT, where the decisions you make three minutes before the deadline may be different from your decisions at the beginning, so it is with finite horizon control.

In the receding horizon formulation, the deadline keeps moving, *i.e.* $\tau = t + \tau'$ where τ' is a fixed interval in the future over which planning occurs. The advantage of the receding horizon approach is that the same control function $c_t(H_t) = U_t$ can be applied indefinitely, as the horizon keeps receding. The disadvantage is that there is never a sense of urgency from a looming deadline.

The infinite horizon case is given by extending the finite horizon to the limit as $\tau \to \infty$. In such cases, it is important that the optimal value function remain bounded, which is typically achieved by attributing exponentially decreasing rewards $r_{t'}(H_{t'})$ to each time t' in the future. This is also known as discounting future rewards.

In greedy control, $\tau = 1$, meaning the planning horizon is limited to a single step. Greedy approaches, also known as myopic approaches, are still prospective, but are often much simpler than non-myopic ones. In some cases they are provably near optimal [33], but in others they can fail arbitrarily badly, such as crossing a street without pausing to look for cars first.

Even though greedy approaches are much simpler than non-myopic approaches, they can still be quite difficult. Consider a vision based active perception problem: computing the expected sensorimotor consequence of a single action requires integrating over all possible images that could be seen afterward, which could be more than $256^{1,000,000}$ possibilities, *e.g.* in the case of 1-megapixel grayscale images.

An alternative, non-optimal approach to control might be to make decisions retrospectively. In this case, actions for the future are chosen that would have been optimal for the past, in hindsight. For example, you may choose to bring an umbrella today if it rained yesterday, which would have been an optimal strategy

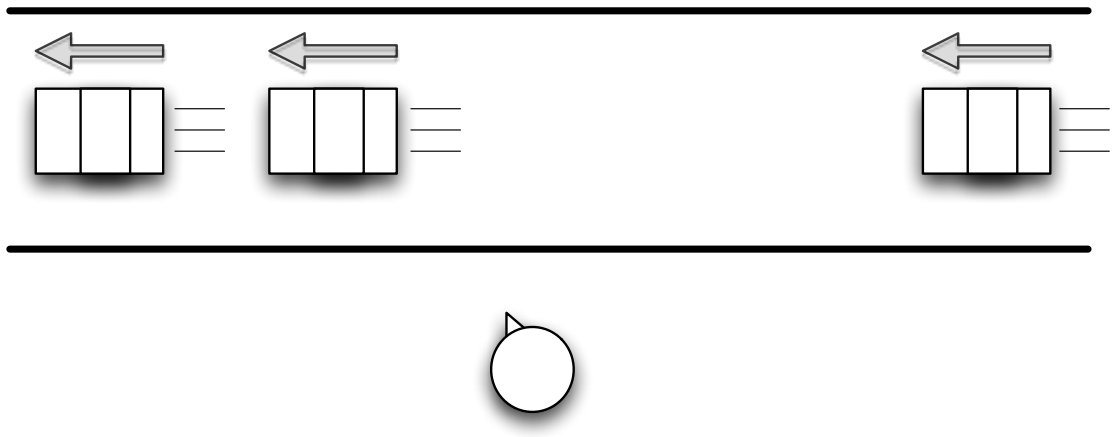


Figure 1.2: When crossing a one way street with cars coming from the right, looking to the right is better for perceiving safety than looking to the left.

yesterday. Or, you may choose to play the winning lottery number from last week, which would have been an optimal strategy last week. There is no guarantee in retrospective approaches that the chosen action will be optimal for any future goal, but we will see some situations in which they may offer a reasonable heuristic for decision making.

Information as a goal for active perception:

Some actions are better than others for gathering information about variables of interest. When crossing a one-way street with cars coming from the right, as in Figure 1.2, it is better to look to the right than to look to the left. Still, it is better to look to the left than to close your eyes. If you choose the wrong action, you may not perceive an oncoming car.

What makes some actions, such as looking to the left, bad for perception, and other actions, such as looking to the right, good for perception?

Much has been understood about neural processing systems by assuming that their goal is to optimize information theoretic quantities (infomax). The specific formulation of the optimality problem has led to several different approaches. Here, we review and contrast these approaches using distinctions made by the definitions given in Figure 1.3. The approaches are illustrated in Figure 1.4.

- Entropy:

$$\mathcal{H}(a) = -\log p(a) \tag{1.6}$$

$$\mathcal{H}(A) = -\int p(a) \log p(a)\, da \tag{1.7}$$

- Conditional Entropy:

$$\mathcal{H}(A \mid b) = -\int p(a \mid b) \log p(a \mid b)\, da \tag{1.8}$$

$$\mathcal{H}(A \mid B) = -\int p(a, b) \log p(a \mid b)\, da\, db \;=\; \int p(b)\mathcal{H}(A \mid b) db \tag{1.9}$$

- Mutual Information:

$$\mathcal{I}(a, b) = \log\left(\frac{p(a, b)}{p(a)p(b)}\right) = \mathcal{H}(a) - \mathcal{H}(a \mid b) \tag{1.10}$$

$$\mathcal{I}(A, b) = \int p(a \mid b)\, \mathcal{I}(a; b)\, da \;=\; \mathcal{H}(A) - \mathcal{H}(A \mid b) \tag{1.11}$$

$$\mathcal{I}(A, B) = \int p(b)\, \mathcal{I}(A, b)\, db \;=\; \mathcal{H}(A) - \mathcal{H}(A \mid B) \tag{1.12}$$

Figure 1.3: Mathematical definitions of information.

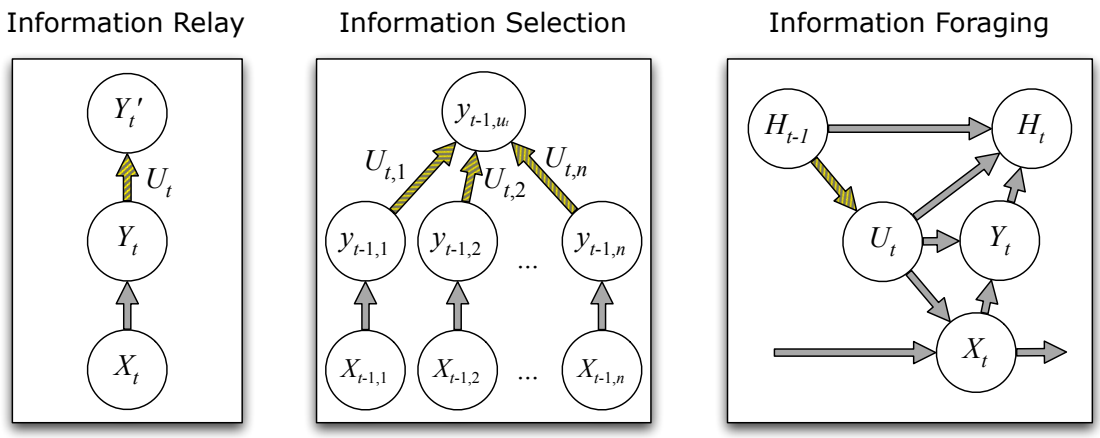
Approach 1: information relay

Barlow articulated the hypothesis that primary sensory neural systems should act as efficient relays for information transmission, producing representations that contain as much information as possible [34]. Higher level neural systems no longer have direct access to the sensory input. They only have access to neural representations, which may be noisy, and lose information. Given that some information will be lost in neural systems, it is important to lose as little information as possible.

Mathematically, neural systems can be modeled as representations Y_t' that are noisy transformations of sensory data:

$$Y_t' = f(Y_t, U_t) + Z_t \tag{1.13}$$

For example, Y_t might represent the retinal representation, Y_t' the representation



$\max_{u_t} \mathcal{I}(Y_t, Y_t' \mid u_t)$
Barlow
Baddeley *et al.*
Bell & Sejnowski
Lewicki
Chapter 2

$\max_{u_t} \mathcal{H}(y_{t-1,u_t})$
Bruce & Tsotsos
Zhang *et al.*
Chapter 3

$r_t(h_t) = \mathcal{I}(X_t, h_t),\ \tau = 1$
Lindley
Nelson *et al.*
Thrun *et al.*
Najemnik & Geisler

$r_t(h_t) = \mathcal{I}(X_t, h_t),\ \tau > 1$
Chapters 4 & 5

Figure 1.4: Several views of infomax active perception.

in primary sensory cortex, U_t the synaptic strengths of the relevant neural systems, and Z_t internal noise. This noise term is simply used to explain why the same retinal representation may not always result in exactly the same cortical representation.

If they take the necessary steps to adapt their synaptic strengths U_t, sensory neurons can be thought of as active participants in perception. From the information relay perspective, sensory neurons should take actions that lead to adjustments in their synaptic strength that maximizes their ability to transmit the information provided by the senses. Thus, they have as their goal,

$$\max_{u_t} \mathcal{I}(Y, Y_t' \mid u_t) \tag{1.14}$$

Bell & Sejnowski showed that a maximum entropy neural code retains the most

information about sensory input [35]:

$$\mathcal{I}(Y, Y'_t \mid u_t) = \mathcal{H}(Y' \mid u_t) - H(Y' \mid Y, u_t) \tag{1.15}$$

$$= \mathcal{H}(Y' \mid u_t) - H(Z_t) \tag{1.16}$$

The noise Z_t is assumed to be independent of the synaptic strength parameter U_t. Thus,

$$\underset{u_t}{\operatorname{argmax}}\ \mathcal{I}(Y, Y'_t \mid u_t) \ = \ \underset{u_t}{\operatorname{argmax}}\ \mathcal{H}(Y'_t \mid u_t) \tag{1.17}$$

leading to the maximum entropy principle. The maximum entropy principle has been used to understand sensory processing systems from the lowest level up to primary sensory cortex. Laughlin showed that blowfly large monopolar cells acted as high capacity information channels for the visual sensory data in the fly's environment [36]. Bell & Sejnowski showed that the structure of V1 simple cell receptive fields matched the transformation $f(Y_t, U_t)$ that maximized the information in the population code [37], and Lewicki found a similar result for cochlear nerve fibers [38].

In Chapter 2, we consider populations of neurons that each, independently, actively adapts its parameters to maximize the entropy of its own output distribution. These actions change the neuron's propensity to fire in a way that is expected to increase the informativeness of signal transmission.

While the goal given in Equation (1.14) tries to keep as much information as possible contained in a sensation Y_t, it will inevitably lose information about some things. The maximum entropy approach is *indiscriminate*, in the sense that it maximizes information about sensory information Y_t, but not necessarily information about any relevant aspect of the world X_t; moreover, maximum entropy approaches are incapable of systematically discarding information contained in the image Y_t that is irrelevant to perceiving X_t. *E.g.*, the representation Y'_t is guaranteed to be maximally informative about the appearance Y_t of a busy street; while we may reasonably hope that this will be a useful representation for discovering whether the street is safe to cross, there is no formal guarantee that this is the case. Instead, the representation may be more useful for measuring the width of the lane lines, or counting the number of birds in the trees.

Moreover, the information relay approach to infomax does not provide a theory for why it is better to look right than left in the example of Figure 1.2. Since the neuron can only control its internal parameters U_t, it neither directly affects the scene X_t, nor the sensation Y_t.

Approach 2: information selection

The information relay point of view has done a remarkable job of explaining early sensory processing in the brain. However, even influential pioneers of the approach recognized its limitations. Barlow writes, "the model of a transmitter, a channel, and a receiver ... is in some ways a poor analogy for the perceptual brain, partly because we must rid ourselves of the idea that there is a homunculus to receive the messages" [39]. Perhaps Barlow is reacting to the indiscriminate nature of maximizing the transmission efficiency of an information relay: by equally representing information about all aspects of sensory information, the information relay approach does not provide a theory for how aspects of the world that are important to the organism can be specifically highlighted, leaving that job to an unspecified homunculus.

An important role of the perceptual system may be to *throw away* information contained in the original sensation Y_t that is irrelevant for perceiving aspects of the world X_t that are important to the organism. Lewi & Weiss found that it was difficult for popular machine learning algorithms, such as cascaded AdaBoost classifiers, to learn to perceive faces in image data using raw pixel values as input, but it was much easier for the algorithm to learn to perceive faces in the same image data if they were represented as edge-orientation histograms (EOH); however, it is impossible to recover the original image Y_t from the EOH, so the EOH loses information about Y_t (image patches), while retaining information about X_t (faces) [40]. Similarly, Shan & Cottrell showed that discarding sign-information from sensory representations was the key to discovering further visual representations that were good for learning about visual categories [41].

The phenomenon of attention has been viewed as a type of information filtering, throwing away extraneous information irrelevant to a current task. Several

theories of attention are based on a second approach to infomax, the information selection approach. In a common formulation, the goal is to optimize the selection of local image features with respect to the information of a preceding event. Bruce & Tsotsos formulated visual salience as a selection process among many local image features [42]. Many local features comprise the observation $Y_t = \{Y_{t,1}, Y_{t,2}, \ldots, Y_{t,n}\}$. A selection operator $U_t \in [1 : n]$ causes a single local feature to be selected for further processing. Since U_t is chosen before Y_t is observed (Figure 1.1), only the preceding sensory information Y_{t-1} is available. The goal of attention is to choose the selection U_t that maximizes the information of the already observed event y_{t-1,u_t}. *I.e.*, given a sensory vector $y_{t-1,1:n}$, the goal of the attention system is to choose for further processing the component y_{t-1,u_t} that had the highest entropy:

$$\max_{u_t} \ \mathcal{H}(y_{t-1,u_t}) \tag{1.18}$$

Zhang *et al.* provide a compelling motivation for channel selection infomax [43]. They posit that the goal of the attentional system is to direct the eyes to regions of the visual field that likely to have contained a specific target of interest. Each region i of the visual field $Y_{t,i}$ is rendered by $X_{t,i}$, where $X_{t,i}$ is either (1) $X_{t,i} = 1$: target of interest, or (2) $X_{t,i} = 0$: not target of interest. Then, the goal of the attention system is to select the image region U_t that maximizes

$$\max_{u_t} \ p(X_{t-1,u_t} = 1 \mid y_{t-1,u_t}) \tag{1.19}$$

Conceptually, this objective can be understood in terms of the following scenario. You are presented with a collection of boxes. Some of them, you are told, have gold coins inside, while the others have something that's not gold. The boxes are translucent, but not transparent. So, you have some sensor information $Y_{t-1,i}$ available from each box i. You can look at all the boxes, but you can only choose one box, U_t, to open, to see if it really has gold inside.

In this scenario, attention is the process that only allows you to open one of the boxes. It would take too long, and be too costly, to spend time carefully examining every aspect of the world. Attention's job is to filter out information

that is probably unneeded, to ensure that you examine aspects of the world that attention hopes will be valuable. While the information relay approach to infomax is indiscriminate, Zhang's information selection approach focuses only on features of the environment relevant to the organism.

Zhang *et al*. further propose that, in free viewing conditions, where there is no clear target object class, the set of possible target objects is very large. In the limit, as this collection of possible objects widens, all possible sensory observations become equally probable, *i.e*.,

$$p(y_{t-1,u_t} \mid X_{t,u_t} = 1) = \kappa \tag{1.20}$$

where κ is a constant independent of y_{t-1,u_t}. Moreover, if all locations have equal prior probability, π, of rendering the target, then

$$p(X_{t-1} = 1 \mid y_{t-1,u_t}) = \pi \ \kappa \ p(y_{t-1,u_t})^{-1} \tag{1.21}$$

In this case, the probability that the location u_t renders a target object is inversely proportational to the marginal probability of $p(y_{t,u_t})$, giving

$$\begin{aligned} \underset{u_t}{\operatorname{argmax}}\, p(X_{t-1,u_t} = 1 \mid y_{t-1,u_t}) &= \underset{u_t}{\operatorname{argmin}}\, p(y_{t-1,u_t}) && (1.22)\\ &= \underset{u_t}{\operatorname{argmin}}\, \log p(y_{t-1,u_t}) && (1.23)\\ &= \underset{u_t}{\operatorname{argmax}}\, \mathcal{H}(y_{t-1,u_t}) && (1.24) \end{aligned}$$

It is interesting to note that

$$p(y_{t-1,u_t}) = \pi \ \kappa + (1 - \pi) \ p(y_{t-1,u_t} \mid X_t = 0) \tag{1.25}$$

indicating that under the model there is a positive linear relationship between the likelihood of the data observed at a location under the background model and the overall probability of the data.

The approaches to information selection in this section are retrospective, because actions U_t are evaluated in terms of already experienced sensory events Y_{t-1} rather than the future distribution of sensorimotor experience $H_{t:\tau}$. Thus, there is no formal guarantee that the action U_t will be optimal with respect to any future goals.

The goal given in Equation (1.19), and its special case, Equation (1.18), are also memoryless, because they explicitly disregard all aspects of sensorimotor history H_{t-1} except for the preceding image Y_{t-1} when choosing U_t. Thus it is the purest form of a reactive system, motivated solely by the present stimulus.

Despite its reactive nature and lack of future guarantees, the goals for active perception given in this section have been useful in explaining a wide array of behavioral phenomena. Zhang *et al*. showed that many perceptual phenomena can be explained as selecting the local image regions with the most instantaneous entropy [43]. In Chapter 3, we modified this approach to compute the entropy of all image features and select among them in real time, and found it empirically useful for highlighting people in real environments.

An information selection approach would account for why it is better to look right than left in the example of Figure 1.2 by defining a category of interest, "cars moving toward me," and direct attention to regions of the scene that seemed to contain objects of that category. But it cannot provide a theory for why "cars moving towards me" is a better category of interest than "cars moving away from me." That determination is left to some other process.

Approach 3: information foraging

A third approach to infomax, often called information foraging [44], was originally articulated by Lindley as "the information provided by an experiment" [14]. Under this view, organisms can be viewed as scientists performing experiments U_t that they expect to produce measurements Y_t. Ideal experiments are ones that are expected to produce measurements that give the most information about some underlying hypothesis X_t, *i.e*. the moment to moment desire of the organism-scientist is to gain information about X_t, and the long term goal is to gather a large amount of information quickly. One way to formalize this goal is to use information gain as a reward function:

$$r_t(h_t) = \mathcal{I}(X_t, h_t) \tag{1.26}$$

$$= \mathcal{H}(X_t) - \mathcal{H}(X_t \mid h_t) \tag{1.27}$$

Note that $\mathcal{H}(X_t)$ is unaffected by h_t, and out of the organism's control. Thus gaining information entails reducing $\mathcal{H}(X_t \mid h_t)$, and so the reward in Equation (1.27) is equivalent to

$$r_t(h_t) = -\mathcal{H}(X_t \mid h_t) \tag{1.28}$$

In Equation (1.28), the organism is rewarded for sensorimotor experiences that reduce the entropy, or uncertainty, of its belief about some state of interest. Given that "the safety of crossing the street in Figure 1.2" is a state of interest, the information foraging approach accounts for why it is better to look right than left. By looking to the left, your uncertainty about oncoming traffic remains high, so it is a bad policy. By looking to the right, your uncertainty about oncoming traffic and the safety of crossing reduces, so it is a good policy.

Looking to the left is like asking the question, "How often do cars pass by on this street?" while looking to the right is like asking the question, "What cars are coming right now?" If you want to perceive whether a street is safe to cross, the latter is a better question than the former. Thus, another interpretation the information foraging approach is that actions U_t are questions that the organism can ask of the environment, and sensations Y_t are the answers given by the environment in response. The goal of active perception is to ask good questions to the environment, and listen to the answers it has to give.

Difficulty of information foraging:

The information foraging view of infomax is prospective. Actions must be evaluated in terms of their expected sensorimotor outcomes. For vision, the space of sensory outcomes is the space of images, $Y_t \in \mathbb{R}^n$, where the number of pixels n can span the range from 100 (small image patches) to 2M+ (HD video). Ideally, we would like to optimize the long term future information gain. Picking the action U_t that is optimal for long term future information gain is especially difficult. Instead of considering all possible sensory outcomes of just one action (*e.g.* integrating over all possible values of an image), the organism needs to consider a sequence of sensory outcomes, and all the actions it is likely to take in response

to those outcomes – consequences of consequences of consequences. Thrun *et al.* specifically recommend against attempting this kind of computation. They write, "given the huge number of *possible* values for the unknown state variables in exploration, any algorithm that integrates over all possible such values will inevitably be inapplicable to high dimensional exploration problems, simply for computational reasons" [1].

The recursive form of the Bellman equation, Equation (1.5), allows dynamic programming to be used to find exact solutions, but only in perception problems where the state, action, and observation spaces are small [45, 46], or are linear Gaussian [47]. Such approaches have been historically useful in some domains. Maximization of expected information gain was proposed by Lindley [14] as a sensible criterion for designing experiments. Stone [47] and Fedorov [48] applied this idea to the efficient estimation of parameters in linear regression and ANOVA models. Bernardo [49] used a Bayesian framework to show that information gain can be used as a utility function in the context of optimal control.

Unfortunately, complicated and high dimensional problems proved difficult for all of these exact solution methods. For this reason information foraging approaches languished for a number of years.

Greedy approaches to information foraging:

There has recently been strong interest in approximate, greedy approaches to information foraging problems. In greedy approaches, we pick the action that is expected to give as much information as possible only in the next observation, *i.e.* the planning horizon $\tau = 1$, and expected values only need to be computed over the single sensory outcome Y_t. In certain cases, greedy approaches to active information gathering are provably near optimal [33]. These tend to be situations in which there are no dynamics. However, in dynamic environments, greedy approaches can be arbitrarily bad. *E.g.*, during eye movements, vision is very poor [18]. In order to get information about something you cannot see, you must first move your eyes, during which time you get no information. Thus the choice to move your eyes is a choice between (1) getting a small amount of information about the things you

currently see, and (2) getting no information for some time, but then getting a large amount of information about the world beyond your current field of view. A greedy approach to information gathering would always choose (1), and the eyes would never move.

Despite their lack of generality, greedy approaches to infomax active perception have been used to great effect in some domains, such as deploying sensors to effectively monitor environmental factors in lakes [50], and in active-learning scenarios to quickly learn how to accurately diagnose health conditions from medical images [51]. Najemnik & Geisler derived a rule for selecting fixations in a visual search task that is optimal for greedy information gain in the limit of a retina with infinite sensors [52].[2] Nelson *et al*. evaluated the information value given by image features chosen by subjects in a concept learning task, where the space of actions and outcomes was small enough that the greedy expected information gain could be computed exactly [53, 54]. Lewi *et al*. found a very efficient approach to find approximations to greedy infomax solutions in 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 [55]. Cakmak *et al*. showed that robot learning improved when robots asked human teachers questions that were expected to give the robots most information, and also that the teaching interactions were more motivating to the human teachers [56].

Infomax approaches in this thesis:

Four chapters in this thesis deal with optimality approaches to active perception. Chapter 2 is an infomax study using the information relay approach. We consider a population of model neurons which each actively tune their parameters to transmit as much sensory information as possible. Chapter 3 is a study using the information selection approach: we adapt the algorithm of Zhang *et al*. to

[2] In this case, the dynamics of eye movement were ignored, and the eyes considered to move instantaneously, thereby escaping the dilemma in the preceding paragraph.

compute visual salience of image data in robotic camera systems faster than real time [43]. In Chapter 4, we study active perception of social contingency from acoustic sensory input. We solve the problem of computing expectations over outcomes of outcomes of outcomes of actions by projecting real acoustic data into a small binary representation that is amenable to dynamic programming based exact solutions. In Chapter 5, we consider the problem of choosing saccade destinations to get information about the location of a search target. We project real images down to a representation $Y' \in [1:10]^{441}$, which is much too large to integrate over. We solve the problem of optimizing the Bellman equation (1.5) implicitly with a reinforcement learning technique called policy gradient [32].

In the remaining chapters, we consider another important and difficult problem for active perception: learning about the hidden causes X_t of sensations Y_t. Theories of active perception require both a sensor model and a dynamics model. Recall Equation (1.2):

$$\overbrace{p(x_t \mid h_t)}^{\text{belief}} \propto \overbrace{p(y_t \mid x_t, u_t)}^{\text{sensor model}} \int \overbrace{p(x_t \mid x_{t-1}, u_t)}^{\text{dynamics model}} \overbrace{p(x_{t-1} \mid h_{t-1})}^{\text{previous belief}} \, dx_{t-1}$$

The sensor model and dynamics model are key components for active perception, and are required in the computation of information. In Chapters 4 & 5, the form of these distributions was constructed by careful analysis of the problem, and the parameters of the distributions were fit by systematic empirical evaluation. In Chapters 6 & 7, we consider how active agents can learn the parameters of these distributions by themselves. In Chapter 6, we focus on the dynamics model, and how an active robot can learn the relationship between the commands it sends to its eyes, and the subsequent sensory consequences. In Chapter 7, we focus on the sensor model, and a how an active robot can learn to perceive visual categories autonomously.

1.2 Themes

This thesis consists of several projects in which we design and implement active perception machines. Aside from this central question, several themes pervade the

chapters of the thesis and bind them together.

Optimality:

Optimality plays an important role in this thesis: finding optimal parameters, discovering optimal inference rules, making optimal decisions in the face of uncertainty. To make a credible claim that something is optimal, we must first establish an evaluation criterion – what are we trying to achieve? – and then fully specify our assumptions in a mathematically rigorous way. Only then can we begin to calculate a solution, or evaluate an approximate solution.

Our use of optimality is not dogmatic, in the sense that it is never the goal to demand the best or nothing. In many cases, approximately optimal solutions to optimality problems suffice. In Chapter 2 & 5 we use gradient-based methods that converge to solutions that are only local maxima of some constrained parameter space. In Chapter 4, we use a receding horizon controller with limited memory and limited lookahead. In Chapter 6, we settle for a maximum conditional *a posteriori* inference rule rather than a maximum *a posteriori* one.

Thus, the role of optimality is more practical and mundane. Todorov highlights three benefits of optimality approaches [57]:

1. Simplicity: Optimality approaches are defined succinctly by an optimality criterion, which is easy to interprate and critically evaluate.

2. Robustness: Alternatives to optimality approaches are typically defined by elaborate solution descriptions, which may be based on hidden and incorrect assumptions, and thus are prone to failure.

3. Believability: Biological systems arise by an optimization process (evolution), and thus typically exhibit similar properties to the solutions found by optimality approaches.

By framing research questions as optimality problems, we mitigate hidden assumptions that lead to wrong results, and maximize our chance of successful experiments. For example, in Chapter 5, we create a foveated object detector that

computes optimal next fixations in order to search a scene for a visual target twice as fast as a previous method that examines the whole image. In Chapter 6, we derive an inference rule to optimally explain the sensory consequences of motor signals. This inference rule allows three robots with different morphologies, in different environments, to learn intentional looking behaviors, even in the presence of extreme experimental perturbations. In Chapter 7, we employ an inference rule meant to optimally explain visual-acoustic experience, which allows a baby robot to learn the appearance of humans from just six minutes of active perception sampled from ninety minutes of experience with the world. It is possible that all of these results could have been obtained using heuristic solution descriptions, without considering the objective we are attempting to optimize. But by framing algorithms in terms of optimal solutions to computational objectives, we improved the chance of success.

An emergent benefit to framing research in terms of optimality was championed vigorously by David Marr: for Marr, the importance of optimality is that it helps to answer the question, "why?" [2]. For example, in Chapter 4, we frame the problem of detecting contingent social interlocutors as an optimal information foraging behavior, given the statistics of social interaction. We find that the behavior of an optimal controller is remarkably similar to the behavior of some human 10-month olds. This does not answer the question "how?", in that it does not tell us how the infant's brain is making moment to moment decisions, but it does give insight into "why?": 10-month-old infants' behavior in social contingency experiments is consistent with the behavior we expect to see if their goal was to maximize the information that they receive.

Time and timing:

Organisms are situated in, and need to understand, worlds that are rapidly evolving. This places important computational constraints on organisms that need to act in, and interact with, their environment. Time and timing play several important roles in this thesis: (1) An emphasis on computationally efficient approaches that are suitable for real-time perception. (2) Dynamics in sensory infer-

ence and motor planning. (3) Attention to the timescales on which sensorimotor learning occurs.

Real-time perception: The history of computer vision is littered with theories that began with some motivating intuition, were implemented in a restricted miniworld, but then failed in real environments, because their initial inspiration failed to account for some important aspect of the rich, changing environments that humans live in [2]. This suggests that in order to test theories of perception, it is important to build systems that implement them. These systems should operate in the environments that humans do, and on the same timescales. Sometimes this requires concessions, such as approximations to optimality criteria. These concessions allow us to build real time systems in order to evaluate theories.

Visual salience is a mechanism for allocating visual processing resources [58]. Simple cues draw attention to regions of the visual field that are likely to contain useful information; those regions must be further processed by some non-salience process. To evaluate whether an account for salience is a good model, we would like to implement it in a system, extract regions deemed salient by the model, and then process these regions to see if they indeed contain useful information. Many published accounts of salience require elaborate computations. Harel *et al*. present a graph-based algorithm that scales as a 4^{th} power with the number of pixels, and Itti & Baldi present an information theoretic approach that requires on the order of a minute to process a single frame [59, 60]. In a changing world, a minute is an eternity; after salience tells you where to look, you must still process and decide what to do with the information at the attended location. Thus, at the time the study in Chapter 3, there was a conspicuous lack of salience algorithms that could be computed in real time in current computers. We developed a real time system based on the salience model of Zhang *et al*. [43]. The real time system enabled us to aim a physical camera to salient regions of a real environment, in real time. We found that in real world environments, the salience algorithm often directs the camera to regions that contain people. This bolsters Zhang *et al*.'s claim that their account of visual salience highlights useful information.

Similarly, in Chapter 5, we compare our method for planning saccade tra-

jectories to one proposed by Najemnik & Geisler [61]. The limiting factor for some experiments in this study was N&G's method for planning saccades, which requires seconds of computation to choose from among 49 next fixation points. Meanwhile, our approach can choose the next fixation target in a fraction of a millisecond, and gives better performance. The efficiency of this approach allowed us to build a computer vision system that discovers and tracks faces very efficiently, providing empirical validation of N&G's original theory.

Dynamics in sensory inference and motor planning: Many approaches to perceptual inference scale poorly when temporal information is added. In contrast, the method presented in Chapter 5 has processing time that *decreases* when temporal information is added. In this case, we have made use of the dynamics of the world to propagate forward in time the information that we collected previously, allowing for dramatic decreases in processing requirements. When temporal information is considered, our approach gets a 10-fold increment in efficiency compared to its static performance.

Timescales on which sensorimotor learning occurs: The poverty of the stimulus argument has been influential in propagating nativist, non-learning approaches to the ontogenetic development of intelligence [19, 62]. Computational approaches are well equipped to investigate the poverty of the stimulus argument more precisely, on a case by case basis.

Morton & Johnson showed that neonates exhibit preferential looking to shape configurations that resemble faces compared to configurations that don't; they argued that infants may be born with innate knowledge of what their species looks like, and expressed skepticism that such preferences could be learned from the small amount of data available to neonates [62]. From a computer vision point of view, it is difficult to create an algorithm that finds faces in real world images; it is typically easier to code an algorithm that learns a solution to finding faces than it is to code the solution itself [63]. In Chapter 7, we build a robot that actively probes her environment to discover what humans look like. From just 6 minutes of experience sampled from 90 minutes of interaction with the world, she not only learns what humans look like, but she shows the same preference for

shape configurations that human neonates do.

We can't claim that neonates are learning using the same algorithm as our robot, but we can make a strong claim that there is enough visual information in the infant's environment to support such learning. By building systems that need to perform tasks that humans do in a similar timeframe, we increase our confidence in the statistical richness of the environment to support rapid learning.

To effectively comment on poverty of the stimulus claims, computational accounts for learning must be sensitive to the time constraints of development. In Chapter 4, we present a reinforcement learning investigation of optimal exploration. We use "vocalizations since birth" as grounded unit of time, and show that reinforcement learning algorithms exist that enable learning of optimal exploration strategies within the timeframe of infant development. In Chapter 6, we show how optimal perceptual inferences made on a short timescale (on the order of seconds) can help a robot to learn on a longer timescale about the appearance of its surroundings (on the order of tens of seconds), which serves as the basis for yet longer timescale learning about the configuration of its body and the sensorimotor consequences of eye movements (on the order of minutes).

Thirteen thousand trillion photons:

This phrase reflects a commitment to working with primary sensory data. The inspiration for much of the work in this thesis comes from studies of information processing on an abstract level. *E.g.*, Kersten & Yuille show how a Bayesian computational theory explains how subjects develop consistent and reliable percepts from underspecified sensory cues [64]. These cues are things like "shadows," "disparity," and "object descriptions." Similarly, Nelson *et al.* show how subjects actively gather information by attending to different cues with varying amounts of predictive power [54]. These are cues like "The plankton's tail is blunt or pointed."

These models use constructed worlds with highly controlled statistical symbolic structure, which allows precise formulation of hypotheses. They give an excellent intuitive understanding of active perception problems that humans face, and have proven very useful as analysis tools for human behavior. We would like

to scale such models to the real world, which has less controlled statistics, and a structure that is harder to extract. In Chapter 4, we consider a robot with a single binary sensor operating on a timescale of 1.25 bits per second. In Chapters 3, and 5–7, we consider image data, operating on a timescale from fifty thousand pixels per second (Chapter 7) to twelve million pixels per second (Chapter 5).

1.3 Contribution to Cognitive Science

The aim of this thesis is to contribute to the understanding of the computational nature of active perception, and to describe the intelligence required to synthesize active perception systems. While it is not our main aim to find algorithms that fit the specific measured parameters of human behavior, we take inspiration from natural intelligence; we attempt to give computational accounts for broad aspects of human behavior; and we show that aspects of human behavior emerge from computational principles.

Inspiration from natural intelligence:

We design and implement several active perception systems that are inspired by human intelligence.

- Inspired by the homeostatic properties exhibited in neurons, we show a novel way to learn sensory transformations similar to those observed in V1. The learning algorithm is more connected to biology than previous information theoretic approaches (Chapter 2).

- An information based model of visual salience inspired by human search asymmetries, combined with the biological constraint that salience processing be very fast, led to the creation of a novel salience implementation that is much faster than other published methods (Chapter 3).

- Inspired by the seemingly active nature of querying in 10-month-old infants as they discover new contingent relationships, we build an active perceptual

system that discovers acoustic contingencies in real time in real life environments, even very noisy ones (Chapter 4).

- Inspired by psychophysical models of foveated vision and the active nature of human eye movements, we build a digital eye that scans static images for faces twice as fast as previous approaches, and in dynamic scenes ten times faster than that (Chapter 5).

- Inspired by the variability of morphologies across individuals and even within individuals throughout their lifetimes, and the biological requirement to do "self-calibration," *i.e.* discover how your body works, we build a method whereby different robots of different morphologies can each discover, on their own, how to move their eyes to fixate desired visual targets (Chapter 6).

- Inspired by the social signals displayed by infants to non-human contingent objects, we show that contingency can serve as a teaching signal to learn a visual model of what humans look like (Chapter 7).

Computational accounts for human behavior:

We give quantitative computational accounts for qualitative phenomena observed in humans. The goal of such accounts is never to make authoritative claims about biology, neuroscience, psychology, or human development. Rather, such studies serve as proofs of concept, demonstrating that certain learning strategies are afforded by the information in an organism's environment. Thus they serve as effective counters to poverty of the stimulus arguments that have historically claimed that observed phenomena must be innate.

- We show that populations of model neurons that tune their parameters to maximize their capacity as information channels yield qualitatively similar receptive fields to those found in V1 (Chapter 2).

- We show that information-gain driven reinforcement learning approaches are capable of explaining the active and intelligent learning behavior exhibited by some human 10-month-old infants (Chapter 4).

- We show that behavior that appears as "forgetting" in human subjects can be interpreted as optimal inference when the world is likely to change in uncertain ways, *e.g.* visual targets can move even when you're not looking at them (Chapter 5). Similar effects have been well illustrated in other domains, *e.g.* by Yu & Cohen [65].

- We show that social contingency could be used as a possible driver of visual learning (Chapter 7).

Similarities to human behavior emerging from optimality principles:

Optimality accounts of intelligence often share much in common with observed biological phenomena. Such coincidences may be the outcome of evolution's own optimizations on similar problems [57].

- We show that a salience algorithm that attends to maximally informative image regions performs equally well at predicting where humans will fixate as algorithms that directly model human fixation data (Chapter 3).

- We show that controllers optimized to gain information about social contingencies exhibit the same turn-taking behaviors observed in ten-month-old infants. Thus we would predict that the human neural reward systems are sensitive to uncertainty reduction. Since this study, Bromberg-Martin and Hikosaka showed that the midbrain dopamine system thought to be responsible for reward-based learning responds strongly to reduction in uncertainty [66] (Chapter 4).

- We show that, in order to do the self-calibration necessary needed to fixate visual targets, it is necessary to predict the specific sensory outcome of a saccade. A mechanism for this prediction would be to remap the current visual field to be in line with its expected location after a saccade. Exactly this remapping has been observed by Duhamel *et al.* in monkey LIP [67] (Chapter 6).

- We show that noisy occulomotor systems attempting to maximize their target fixation accuracy exhibit an undershoot bias (Chapter 6).

- We show that a baby robot that learns about the visual appearance of humans via an acoustic contingency cue shows the same preference to schematic face and non-face stimuli as 40-minute-old neonates [62], and develops the same preference for people around it compared to strangers observed in 1-day-old neonates [68].

Part I

Infomax 1: The Channel View of Information

Chapter 2

Learning Sensory Representations with Intrinsic Plasticity

Abstract

Intrinsic plasticity (IP) refers to a neuron's ability to regulate its firing activity by adapting its intrinsic excitability. Previously, we showed that model neurons combining a model of IP based on information theory with Hebbian synaptic plasticity can adapt their weight vector to discover heavy-tailed directions in the input space. In this chapter, we show how a network of such units can solve a standard non-linear independent component analysis (ICA) problem. We also present a model for the formation of maps of oriented receptive fields in primary visual cortex and compare our results to those from ICA. Together, our results indicate that intrinsic plasticity that tries to locally maximize information transmission at the level of individual neurons may play an important role for the learning of efficient sensory representations in the cortex.

2.1 Introduction

2.1.1 Mechanistic *vs*. optimality models

Computational models of unsupervised learning of sensory representations in the brain abound. Frequently, they fall into one of two categories: *mechanistic models* or *optimality models*. Mechanistic models start from neuroscientific data about the structure of cortical networks and cortical plasticity mechanisms (cell types, connection patterns, plasticity rules, *etc.*) which are distilled into simplified models. These models are trained on actual sensory data or noise patterns and the learned representations can be compared to neurophysiological observations. If the resulting representations are similar to those found in the brain then this provides evidence that the processes in the brain have been accurately captured, but it does not clarify why the brain operates this way or in what sense the brain's solution may be optimal. An example of a model of this kind is by Linsker [69] where V1-style orientation columns are learned from random prenatal visual noise through Hebbian learning. Later, Miller extended this work to learn many of the various map-structures in V1, and used model neurons that were somewhat more plausible [70].

Optimality models focus on the abstract computational goal of the problem. For the case of learning sensory representations they start by asking: what is the *optimal* way to represent sensory stimuli such as natural images, where optimality is usually defined with respect to certain statistical criteria (*e.g.*, sparseness, independence, temporal coherence, *etc.*) and additional constraints. Algorithms are derived to learn the optimal solution to the problem, which can again be compared to neuroscientific data. If the found solution resembles the biological solution, then this provides evidence that the brain may in fact be trying to optimize a similar objective function. Through what mechanisms the brain may achieve this goal is typically not answered, however. Some examples of such an approach will be given below.

Both mechanistic and optimality models have their merits, but for a comprehensive understanding of sensory coding in the cortex we arguably have to

develop models that bridge optimality and mechanistic levels of description [2]. Such models should explain how the physiological mechanisms contribute to optimizing the system's information processing properties in a meaningful way. In the following, we develop a model that can be viewed as a step in this direction.

2.1.2 Information maximization

A central idea in many optimality models of the development of sensory representations is information maximization [34, 71–74]. According to some formulations of this idea, individual neurons should maximize the entropy of their firing rate distribution. If the firing rate is constrained to lie in a fixed interval between zero and the neuron's maximum firing rate, then entropy maximization means that the neuron should use all its firing rate levels equally often. In order to achieve this, it should spread out its responses in dense regions of the input space and compress responses in sparse regions such that it maps the distribution of its inputs to a uniform distribution of its outputs, maximizing entropy. Biological evidence for this idea comes from Laughlin, who showed that blowfly large monopolar cells have been adapted so that their input/output transfer functions nearly optimally represent the contrast statistics of the blowfly's visual environment [36].

Information maximization may not be the only important objective, however, and energy considerations may also play an important role for sensory coding in the brain, *e.g.* [75]. In particular, Baddeley *et al.* found that neurons in different visual cortical areas of cats and monkeys show exponential distributions of their firing rate. They have argued that this maximizes a neuron's information transfer given a fixed energy budget [76]. This is because the exponential distribution has the maximum entropy among all distributions of a positive random variable (the firing rate) with a fixed mean. This and other reasons suggest that *sparse* representations, where individual units are highly active only rarely, may be an important principle of sensory coding [77].

On the modeling side, Olshausen & Field showed that localized, oriented, and bandpass receptive fields similar to those observed in primary visual cortex (V1) arise when optimizing image reconstruction error subject to lifetime sparse-

ness constraints [78]. They imposed a sparse prior on the contribution of each basis function in a generative model with the intuition that among the space of possible sources of an image, each one is present only rarely. In a closely related approach, Bell & Sejnowski showed that the information maximization principle can be applied to the independent component analysis (ICA) problem. They applied their technique to natural images and also found localized, oriented, and bandpass sources [37].

2.1.3 What is the role of intrinsic plasticity for learning sensory representations?

Most work on the learning of sensory representations has focused on synaptic plasticity as the only mechanism for learning efficient codes. But it is becoming increasingly clear that biological neurons also regulate their pattern of firing by adapting their intrinsic excitability through the modification of voltage-gated channels in their membrane. Such *intrinsic plasticity* (IP) seems to be a ubiquitous phenomenon in the brain [79]. For example, Desai *et al.* showed that neurons that had been prevented from spiking for two days increased their response to current injection [80]. Consistent with this finding, it is frequently assumed that IP contributes to the homeostasis of a neuron's mean firing activity. A few computational models do in fact incorporate a mechanism for regulating the mean activity level of a unit by controlling a "threshold" parameter [81–83]. But it is also plausible that IP may help to optimize the encoding and transmission of information in a more sophisticated fashion. Concretely, it has been speculated that IP may be instrumental in achieving approximately exponential firing rate distributions in cortical neurons [84]. More recently, Triesch showed that an IP mechanism that drives a neuron to exhibit an exponential firing rate distribution can synergistically interact with Hebbian learning at the synapses. The two processes lead to the discovery of heavy-tailed directions in the input space [85, 86].

In this chapter, we extend these results to networks of neurons with IP and Hebbian learning. Our specific goal is to explore the potential role of IP for learn-

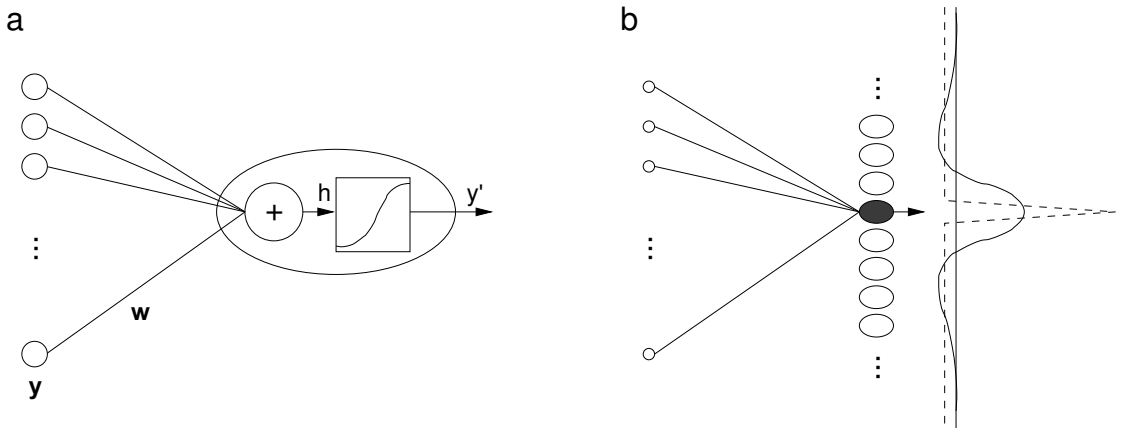


Figure 2.1: **a**: Illustration of an individual unit of the network. The weights **w** are adapted through Hebbian learning, the sigmoidal non-linearity h is adapted through intrinsic plasticity. **b**: Network architecture. The most activated unit (shaded) determines the sign and amount of synaptic learning in neighboring units via a neighborhood function. Two examples of neighborhood functions are shown (not drawn to scale).

ing efficient map-like representations for sensory stimuli. The model we present in the following attempts to bridge the gap between mechanistic and optimality models. On the one hand, it has a clear connection to the idea of information maximization and energy efficient coding [87]. On the other hand, it has a mechanistic formulation that is biologically viable because the learning mechanisms make use of information that is local in time and space. While similar bridges have been attempted before, *e.g.* [82, 88], our model is distinguished by utilizing an IP model derived from information theory as a mechanism for the learning of efficient sensory representations.

2.2 Network model with intrinsic plasticity

In this chapter, we use bold variables, *e.g.* $\mathbf{y}$, to represent vectors, and italics, *e.g.* y_i to represent scalar values. Subscripts index elements, *e.g.* y_i is the i^{th} scalar element of vector $\mathbf{y}$, and $\mathbf{w}_i$ is the i^{th} vector of a collection of vectors. Values

with primes, *e.g.* y'_i denote functional transformations. Specifically, $\mathbf{y}$ represents sensory experience, and $\mathbf{y}'$ is a neural representation of that sensory experience.

We consider a network of units learning to represent a sensory input vector $\mathbf{y}$. The activity y'_i of unit i in the network in response to input $\mathbf{y}$ is given by:

$$y'_i(h_i) \stackrel{\text{def}}{=} [1 + \exp(-a_i h_i - b_i)]^{-1}, \text{ with } h_i \stackrel{\text{def}}{=} \mathbf{y} \cdot \mathbf{w}_i \ , \tag{2.1}$$

where $\mathbf{w}_i$ is the neuron's weight vector, "$\cdot$" denotes the inner or dot product, and a_i and b_i are adjustable parameters of the neuron's transfer function that are controlled by IP (compare Fig. 2.1**a**). In particular, a_i and b_i are adapted in such a way that the unit's output y_i assumes an approximately exponential distribution. To this end, Triesch previously derived a learning rule for a_i and b_i that performs stochastic gradient descent on the Kullback-Leibler divergence between the unit's output distribution and the desired exponential distribution. This leads to the following learning rule [86, 89]:

$$\begin{aligned} a_i &\leftarrow a_i + \eta_{\text{IP}} \left[a_i^{-1} + h_i - (2 + \mu^{-1}) h_i y'_i + \mu^{-1} h_i y_i'^2 \right] \\ b_i &\leftarrow b_i + \eta_{\text{IP}} \left[1 - (2 + \mu^{-1}) y'_i + \mu^{-1} y_i'^2 \right] \ , \end{aligned} \tag{2.2}$$

where "$\leftarrow$" denotes assignment, η_{IP} is a small learning rate and μ is the desired mean activity of all units. Since this learning rule has the effect of making the distribution of y'_i a sparse, approximately exponential distribution, it maximizes the unit's entropy under the constraint of a fixed average activity: the unit transmits information efficiently. Note that this rule is local in space and time, making it physiologically viable.

Plasticity of the weight vectors $\mathbf{w}_i$ is modeled with a Hebbian learning rule. In [85], Triesch considered a single unit learning rule of the form $\Delta\mathbf{w} \propto \mathbf{y}y'$. He showed that the coupling of IP with this form of Hebbian learning allowed the unit to discover heavy-tailed directions in the input, and generalized this result to other Hebbian learning rules in [86]. To extend this approach to a network of model neurons, we introduce a *neighborhood function* $\mathcal{N}$ as illustrated in Fig. 2.1**b**. The value of the neighborhood function for neuron i is determined by its activity y'_i and the activities of all other neurons, *i.e.* $\mathcal{N}(y'_i; \mathbf{y}')$. In particular, we are

considering neighborhood functions that depend on a unit's distance to the most activated unit in the layer, as frequently used in self-organizing maps. Specific forms of $\mathcal{N}$ are introduced below. The general idea is that the neighborhood functions can take on positive and negative values, such that learning is Hebbian for some units and anti-Hebbian for others. This is used to correlate and decorrelate weight updates in specific sets of units, allowing different units to develop different stimulus preferences and facilitating the formation of maps of smoothly varying stimulus preferences. The decorrelation serves the goal of reducing redundancy in the representation, the map formation contributes to wiring length minimization, because units with similar properties will be grouped together. After each stimulus presentation, the weights are updated according to:

$$\Delta \mathbf{w}_i = \mathbf{y} y_i' \mathcal{N}(y_i'; \mathbf{y}'), \quad \mathbf{w}_i \leftarrow \frac{\mathbf{w}_i + \eta_{\text{Hebb}} \Delta \mathbf{w}_i}{\parallel \mathbf{w}_i + \eta_{\text{Hebb}} \Delta \mathbf{w}_i \parallel}, \tag{2.3}$$

where η_{Hebb} is a learning rate and the normalization of the weight vector to unit length mimics competition between synapses on a neuron's dendritic tree [70].

2.3 The "bars" problem

As a first test-bed for studying the learning of sensory representations with networks of units with intrinsic plasticity we consider the "bars" problem. This is a standard non-linear ICA problem introduced by Földiák [83]. Horizontal and vertical bars are presented on a retina of R-by-R pixels. The presence or absence of a bar is independent of that of any other bars. The unsupervised learning problem is to learn filters that correspond to the individual independent components, *i.e.* the bars. The bars problem is non-linear because the pixel at the intersection of two bars is just as bright as any other pixel of the bars, not twice as bright. In our previous work [85, 86], Triesch showed that a single model neuron with IP and Hebbian learning robustly discovers one of the bars when exposed to stimuli from the bars problem. Here, we use a population of units to learn the complete problem. We use a retina of size 10-by-10 pixels and the probability of any of the 20 bars occurring in a given stimulus is 10%. The bar stimuli are unnormalized such

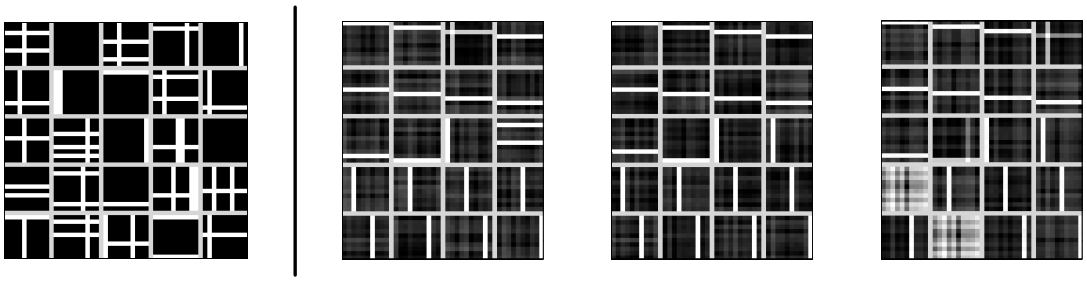


Figure 2.2: *Left*: Example bars stimuli. Stimuli are created by adding bars independently with 0.1 probability. *Right*: Examples of bars learned when β is too low ($\beta = 0$), just right ($\beta = 0.2$), and too high ($\beta = 0.5$) respectively.

that every "on" pixel has value 1.0 and every "off" pixel has value 0. Since we want filters that respond highly when bars are present, and not otherwise, the desired mean firing rate is set to $\mu = 0.1$ which corresponds to 10% of a unit's maximum activation. $\mathcal{N}$ is chosen to enforce a winner-take-all competition between the units, so that the maximally activated neuron updates its weight vector in a standard Hebbian fashion, and all other units update their weight in an anti-Hebbian manner regulated by a decorrelation parameter β:

$$\mathcal{N}_{\text{bars}}(y_i'; \mathbf{y}_i') \stackrel{\text{def}}{=} \begin{cases} 1 & : \quad y_i' = \max(\mathbf{y}_i') \\ -\beta & : \quad \text{else} \end{cases} . \tag{2.4}$$

All units update their intrinsic parameters independently, as described in (2.2).

We examined the learning of bars within the described framework, systematically probing the value of the neighborhood-interaction parameter β, which ranged from 0 to 0.5 in steps of 0.05. Other parameters were: $\eta_{\text{Hebb}} = 0.01$, $\eta_{\text{IP}} = 0.005$, and $\mu = 0.1$. The networks always consisted of 20 units (the number of individual bars). For each value of β we ran 30 independent experiments with 300,000 randomly generated bars stimuli each. Typical examples of bars stimuli and learned representations for different values of β are shown in Figure 2.2. We found that the learning result fell in one of three regimes depending on whether there was too little neighborhood interaction, a good amount of interaction, or too much. Perfect learning results were obtained for β values from 0.1 to 0.2 as

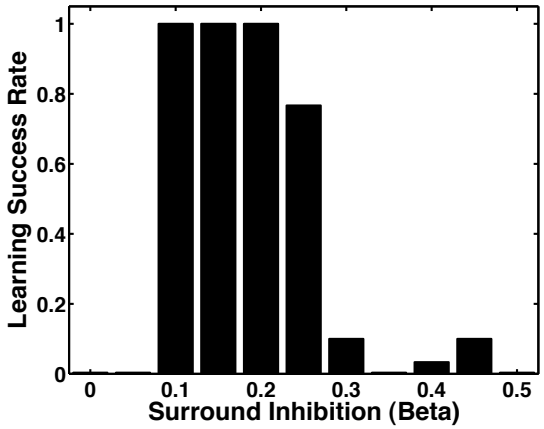


Figure 2.3: Fraction of simulations (out of 30) in which a correct representation was learned for various values of β. When β was 0 or 0.05, a correct representation was never learned. When β was 0.1, 0.15, or 0.2, a correct representation was always learned. When β was 0.3 or greater, correct representations were learned only rarely. For typical examples of representations learned in each regime, refer to Figure 2.2.

illustrated in Fig. 2.3. This means that every unit in the network learned to represent one distinct bar. Learning substantially worsened when β was less than 0.1 or greater than 0.25. When β is too low, all bars are learned, but some are duplicated in the population (some filters learn more than one bar). When β is too high, all bars are learned exactly once, but some filters learn two bars, leaving other filters to learn no bars (see examples in Fig. 2.2).

Varying the learning rates η_{Hebb} and η_{IP} affected learning little, provided both remained above 0. The complete set of filters would *not* be learned without intrinsic plasticity, however. We also studied the influence of μ on the learning result. When μ was 0.05, redundant filters were learned, i.e. multiple units learned to represent the same individual bar while some bars were not represented at all. When it was 0.2, multiple bars were represented within single filters. This suggests that when the true mean of the components is unknown, it may be a better strategy to choose μ too high rather than too low. This way, all true sources will likely be captured because individual filters each learn to represent several sources.

Since its introduction by Földiák, a number of different network architectures for solving the bars problem have been proposed and a number of variations on the problem have also been considered in the literature. The performance of some of the more complex approaches has been tested quite thoroughly, *e.g.* [90]. While a comprehensive review of this literature is beyond the scope of this chapter, it is worth pointing out that our approach shares certain similarities with Földiák's original method [83] and some subsequent approaches. First, our IP mechanism has a similar function as the adaptive threshold regulation in his network. Second, we also utilize a combination of Hebbian and anti-Hebbian weight updates, because the neighborhood function changes the sign of the weight update (positive for most activated unit, negative otherwise). In contrast to Földiák's original method, however, our network does not require adaptable lateral weights between the y'-units to function. Thus, our solution is conceptually particularly simple.

2.4 Modeling the emergence of orientation maps

Receptive fields of simple cells in primary visual cortex (V1) are oriented, localized, and bandpass. In addition, neighboring neurons in V1 will have a similar orientation preference, giving rise to smooth orientation maps. For modelling the emergence of orientation maps, we consider the neurons in our network to be located on a two-dimensional sheet, with neuron i at grid position $(j,k)_i \in \mathbb{N} \times \mathbb{N}$ after the fashion of a self-organizing map (SOM). The most active unit exhibits a center-surround influence on learning in its neighbors according to a difference of Gaussians (DOG) neighborhood function centered around it. Let $d_i^2 \stackrel{\text{def}}{=} (j_i - j_*)^2 + (k_i - k_*)^2$ be the squared distance of neuron i to the most activated unit in the layer at (j_*, k_*). We define:

$$\mathcal{N}_{\text{map}}(y_i'; \mathbf{y}') \stackrel{\text{def}}{=} \frac{1}{2\pi\sigma_c^2} \exp\left(\frac{-d_i^2}{2\sigma_c^2}\right) - \frac{1}{2\pi\sigma_s^2} \exp\left(\frac{-d_i^2}{2\sigma_s^2}\right) , \tag{2.5}$$

where σ_c and σ_s determine the range of the center and surround interaction. In our case, this neighborhood function serves a slightly different role than the Gaussian weighting function usually used in traditional SOMs. The role of $\mathcal{N}$ in our case

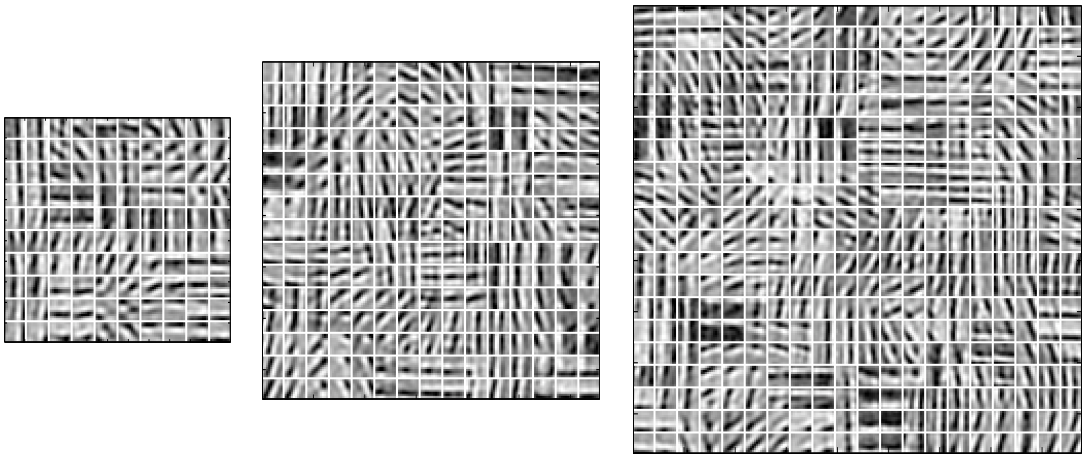


Figure 2.4: Receptive fields learned on various map sizes from natural image patches. We plot the set of resulting weight vectors for networks of three sizes. *Left*: 10-by-10 (100 units, complete), *Middle*: 15-by-15 (225 units, 2.25 times over-complete), *Right*: 20-by-20 (400 units, 4 times over-complete). Parameters were: $\eta_{\text{Hebb}} = 0.05$, $\eta_{\text{IP}} = 0.01$, $\mu = 0.15$, $\sigma_c = 1$, $\sigma_s = 1.5$.

is short range cooperation among units combined with a decorrelation of weight updates for units that are less close. Units that are very far away from the winning unit are prevented from learning altogether. This simple mechanism avoids the development of a large amount of redundancy in the learned representation and it facilitates the formation of maps with smoothly varying orientation preference.

2.4.1 Experiment 1: learning over-complete representations for natural image patches

We trained networks on natural images collected by Van Hateren [91]. We used log-intensity images because these have greater contrast and this transform is performed in the early visual pathway [91]. We convolved the images with a difference of Gaussians (DOG) filter to model the center-surround opponency of neurons in the lateral geniculate nucleus (LGN) [70]. For the DOG filter, we used a center width of 1 pixel and a surround width of 1.2 pixels. 500 image patches of size 10-by-10 pixels were drawn at random from each of 375 images, and were

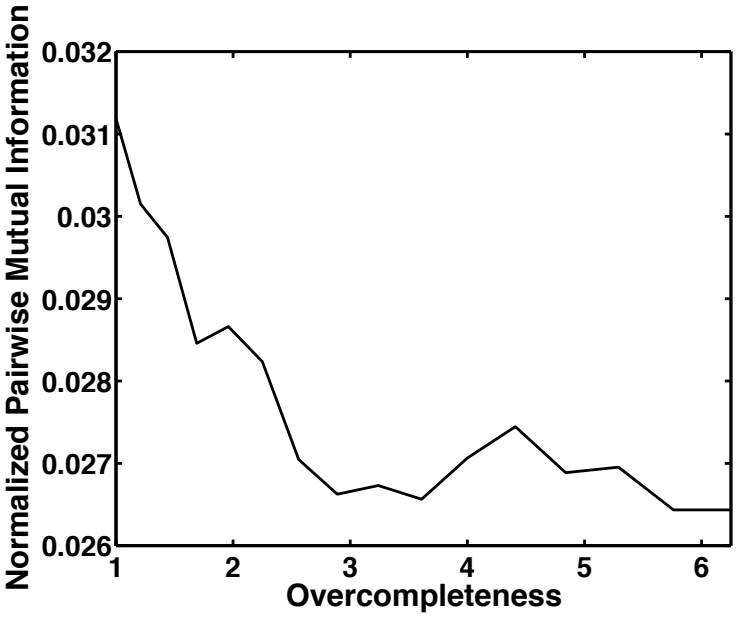


Figure 2.5: Average normalized pairwise mutual information between units in networks with different degrees of over-completeness. The generally low values demonstrate that the network successfully avoids learning many redundant filters. Small degrees of over-completeness actually reduce the average pairwise mutual information measure.

presented once to each neuron in our population (one epoch). The input had positive and negative values simulating populations of ON and OFF cells in the LGN [69]. We used networks of various sizes ranging from 10-by-10 units to 25-by-25 units. Each unit had a 10-by-10 receptive field size, making the populations 1 to 6.25 times over-complete. Parameters were: $\eta_{\text{Hebb}} = 0.05$, $\eta_{\text{IP}} = 0.01$, $\mu = 0.15$, $\sigma_c = 1$, $\sigma_s = 1.5$. Training lasted for 50 epochs each consisting of 3000 image patch presentations for a total of 150,000 natural stimulus presentations.

Typical results of learning are shown in Figure 2.4 for networks of three different sizes. Learning was robust to changes in the parameters over a wide range of values. The learned filters are Gabor-like and exhibit a variety of orientations, frequencies, and locations. Moreover, they exhibit smooth interpolation in local regions of the map. This is reminiscent of the orientation-map structure in V1.

We studied the amount of redundancy in the learned representation by measuring the mutual information between all pairs (i, j) of units in a given network. Here we used a normalized mutual information measure:

$$\mathcal{I}^*(Y_i', Y_j') \stackrel{\text{def}}{=} \frac{2\mathcal{I}(Y_i', Y_j')}{\mathcal{H}(Y_i') + \mathcal{H}(Y_j')} = 1 - \frac{\mathcal{H}(Y_i' \mid Y_j') + \mathcal{H}(Y_j' \mid Y_i')}{\mathcal{H}(Y_i') + \mathcal{H}(Y_j')} \,, \tag{2.6}$$

where $\mathcal{I}(Y_i', Y_j')$ denotes the mutual information between random variables Y_i' and Y_j' and $\mathcal{H}(\cdot)$ denotes the entropy. This measure varies between 0 and 1, with 0 indicating independence and 1 indicating maximal dependence of the filter responses. We calculated the average pairwise normalized mutual information by analyzing the empirical firing histograms with 6 equally spaced bins for networks of different sizes. Fig. 2.5 plots the average normalized mutual information as a function of the amount of over-completeness of the network. The generally small values of below 0.04, i.e. less than 4% of the maximum possible mutual information, indicate that on average a unit's responses are highly correlated to only a small number of other units. The networks successfully avoid learning many redundant filters, which implies that each network's representation of its input can be considered efficient. Interestingly, as over-completeness increases, the values of the average mutual information actually slightly decrease. This decrease in the per-unit redundancy in overcomplete maps implies that as the number of units increases, representation space is covered more evenly and efficiently.

The map-formation mechanism based on the neighborhood function $\mathcal{N}_{\text{map}}$ encourages close neighbors to develop similar weight vectors, making their responses positively correlated, while somewhat more distant units are driven to develop anti-correlated responses. In Fig. 2.6, we plot the average correlation in the responses of pairs of neurons as a function of their separation for a network with 15-by-15 units. As predicted, close neighbors have positively correlated responses while more distant neurons have anti-correlated responses. Very distant neurons are uncorrelated. This pattern mirrors the shape of the neighborhood function $\mathcal{N}_{\text{map}}$. Thus, the pattern of correlations can be influenced by specific choices of $\mathcal{N}_{\text{map}}$. This result also reflects the low levels of redundancy in the learned representation discussed above.

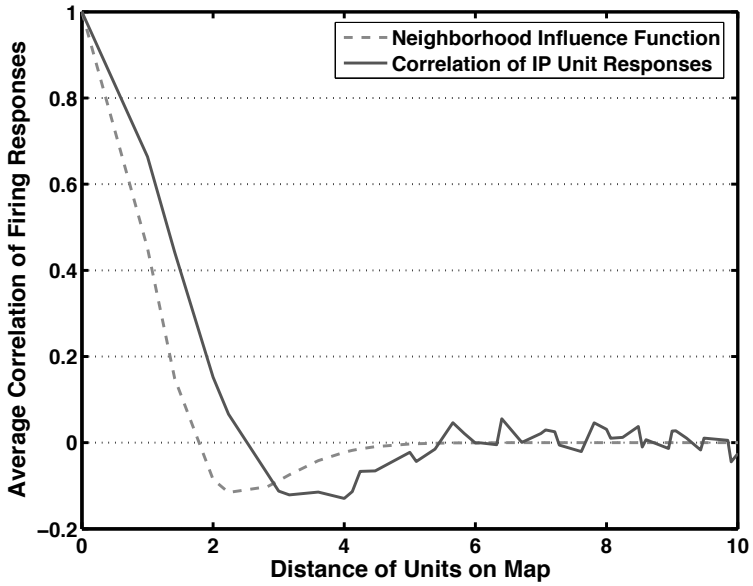


Figure 2.6: Average correlation of units' activities as a function of their spatial separation for a network with 15-by-15 units (2.25 times over-complete representation).

2.4.2 Experiment 2: role of IP in the learning process

In order to better understand the role of IP in the learning process, we systematically varied the strength of IP and observed its impact on the learned filters. Since the networks develop units whose receptive fields are similar to Gabor filters, we assessed network performance by measuring how well the learned filters matched Gabor filters — the standard model of V1 simple cell responses — for different learning rates η_{IP}. To this end, we compared each learned filter to a large number of Gabor filters by computing the dot product between the learned filters and standard Gabor filters. All vectors were normalized to unit length, so a dot product of 1 indicates identical vectors and a dot product of 0 indicates orthogonal vectors. The Gabor filters used for comparison covered odd and even symmetry, 100 center locations, 6 sizes of the Gaussian envelope (ranging from .75 to 4.5), 15 values for the spatial frequency (covering the range from 0.03 cycles per pixel up to 0.45 cycles per pixel) and 8 different orientations (22.5 degree steps). These

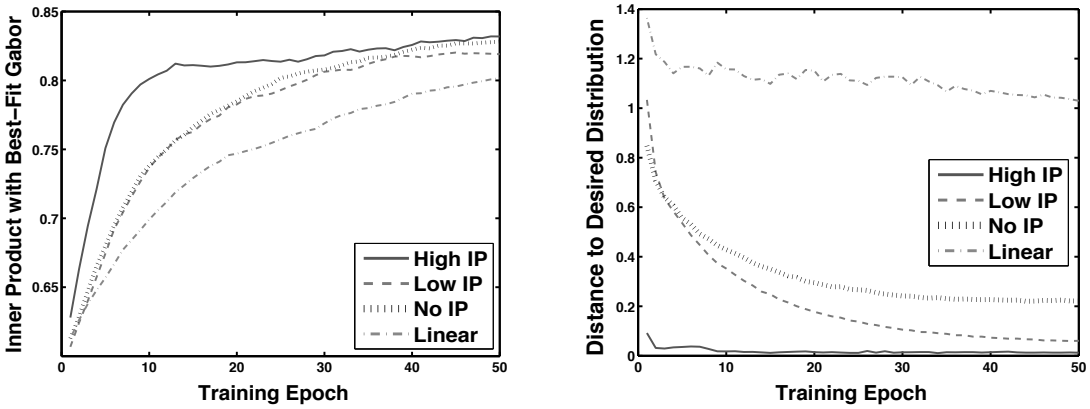


Figure 2.7: Dynamics of learning with and without intrinsic plasticity (IP). The left panel plots the average similarity of learned filters to Gabor filters as a function of the number of learning epochs. Similarity to Gabor filters is calculated as the dot product of a filter with its best-fitting Gabor filter. While IP is not necessary to learn Gabor-like receptive fields, it speeds learning substantially. The right panel shows the average similarity of the marginal distribution of filter responses to that of an exponential distribution with the desired mean. With IP, units quickly assume exponential activity distributions. This effect is not observed in linear units and is less pronounced in units with a fixed sigmoidal non-linearity. Each epoch contains 3000 image patch presentations

values were chosen to fully cover the range of filters learned on 10-by-10 image patches by both our IP model and ICA (see below) [92].

The results are shown in Fig. 2.7. We compared 4 conditions: *High IP* and *Low IP* used the method described above with $\eta_{\text{IP}} = 10^{-2}$ and $\eta_{\text{IP}} = 10^{-5}$, respectively. Condition *No IP* used a fixed, non-adaptive sigmoid non-linearity that was chosen to be $a = 5$ and $b = -2.5$, corresponding to a sigmoid that is roughly linear on the input range 0 to 1. Finally, condition *Linear* used fixed linear units. As shown in Fig. 2.7 (left panel), condition *High IP* was fastest to obtain Gabor-like receptive fields. Interestingly, however, we found that IP is not strictly necessary to learn Gabor-like receptive fields. Even in conditions *No IP* and *Linear*, Gabor-like receptive fields will develop in the network, but at a dramatically slower rate.

This suggests that IP's role in our networks may be primarily to ensure efficient information transmission in individual units and to speed the learning process of the weights, but it does not dramatically alter the resulting weight vectors. The interesting result that somewhat Gabor-like receptive fields even emerge in linear units is caused by the neighborhood function $\mathcal{N}_{\text{map}}$, which forces units to perform anti-Hebbian weight updates whenever the most activated unit is close but not very close to them.

We also measured if and how fast the four different conditions would lead to exponential activity distributions in the units of the network. To this end we measured the marginal activity distributions of individual neurons using a discrete binning with 50 equally spaced bins and compared them to the desired exponential distribution using the L-2 norm. The *High IP* and *Low IP* conditions produce activity distributions that are very close to exponential — the *High IP* condition achieves this much faster, however. In the *No IP* condition (fixed sigmoidal nonlinearity) the units' activity distributions move closer to an exponential shape as their weight vectors are changing, but the units stop short of exhibiting close-to-exponential activity distributions in their firing patterns. In the *Linear* condition, activity distributions of individual neurons remain very far from sparse exponential distributions.

2.4.3 Experiment 3: comparison with ICA

In order to better understand the relation of our model to conventional approaches, we compared the population of learned filters with those resulting from ICA. All simulations were done using Hyvärinen and Hoyer's imageica package (`http://www.cis.hut.fi/projects/ica/imageica/`) [92]. We used the ICA algorithm with 100 filters of 10-by-10 pixels. The training set contained 15,000 image patches and we learned for 300 iterations. No extra pre-processing was performed beyond the whitening procedure that is part of this ICA algorithm. Figure 2.8 displays the learned receptive fields from the ICA algorithm. As expected, we also observe filters that are localized, band-pass, and oriented, and resemble Gabor filters.

Figure 2.8: Set of filters learned by ICA. Each filter has been individually normalized.

Our first analysis aimed to quantify how well receptive fields learned with our network or with ICA matched standard Gabor filters. We found the best fitting Gabor filter for each learned receptive field by an exhaustive search over a set of different Gabor filters covering the complete range of learned filters as described in the previous section. These discrete filters were chosen to fully cover the support of the empirical learned-filter distributions that resulted from both the ICA and IP models. We found that changing the number or range of discrete filters did not significantly alter the shape of the resulting histograms, suggesting continuous underlying filter distributions. Learned filters were compared to their best matching Gabor filters by computing the inner product of the two. On average, filters from our network are more similar to Gabor filters than the filters resulting from ICA. The average dot product to the best matching Gabor filter is 0.8921 for the filters in a 15-by-15 network with IP and only 0.7675 for ICA. A possible reason for this poor fit of ICA filters is that they tend to be quite elongated, while we only consider Gabor filters with rotationally symmetric Gaussian envelopes.

Our second analysis considered the variety of different filters learned by the network with IP or by ICA. Figure 2.9 shows the distribution of various filter

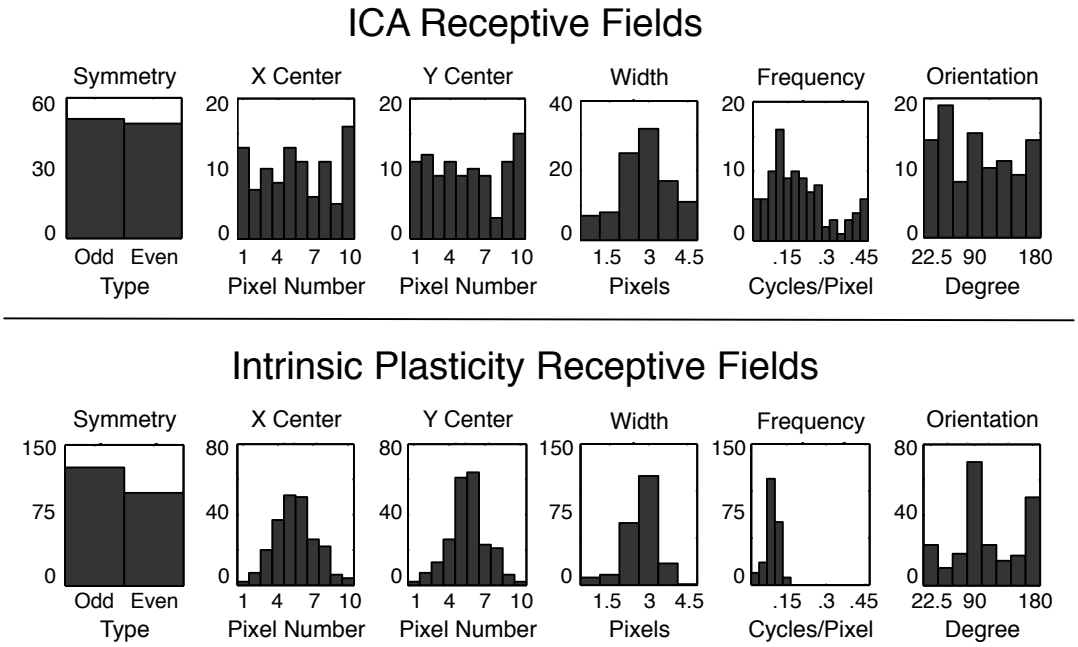


Figure 2.9: Comparison of filters learned by our network with those resulting from ICA.

properties in both cases. Generally, the ICA filters tend to exhibit a greater variety along many different dimensions. *E.g.*, a wider range of spatial frequencies are covered by the ICA filters. A part of the explanation for this behavior is that while ICA tries to achieve independence between all filters, our simple network merely works to decorrelate the responses of filters that are sufficiently far apart in the layer while close-by units are actually encouraged to develop positively correlated responses.

2.5 Discussion

Different forms of plasticity are involved in shaping sensory representations in the brain, and it is important to understand how these different mechanisms interact. In [85, 89], Triesch developed model neurons that maintain sparse lifetime distributions of their individual activities through intrinsic plasticity (IP) and showed that, when IP is combined with various forms of Hebbian learning at the

synapses, a single unit will discover heavy-tailed directions in its input [85, 86]. Here, we constructed networks of such neurons whose learning was coupled using different neighborhood interaction mechanisms: a direct decorrelation method and an approach facilitating the formation of smooth maps of stimulus preferences. In the former case, we solved the "bars" problem, a standard non-linear independent component analysis (ICA) task, and in the latter we found maps of Gabor-like receptive fields as seen in primary visual cortex when learning on natural image patches. We demonstrated that the IP mechanism, while not being strictly necessary for this behavior, significantly speeds up the learning process. Moreover, the learned representations more closely matched the energy-efficient exponential distributions observed in cortical firing, which have both information maximizing and sparse coding properties. When comparing the learned filters in the network to those resulting from ICA, we found that our filters a) provide a closer match to standard Gabor filters and b) are automatically arranged on a smooth map. The learned filters are not as independent as those learned via ICA because significant correlations between neighboring units are introduced, which is biologically plausible, however.

Our simple model is able to learn Gabor-like receptive fields from natural images and arranges the filters into smooth maps. A number of previous models (both mechanistic and optimality ones) have demonstrated similar results. Among them are models based on extensions to the self-organizing map framework [93], BCM-based models [88], topographic ICA [94], extensions to sparse coding approaches [95], and others. What distinguishes our model from these earlier ones is that it utilizes an IP mechanism to obtain energy efficient coding, directly ensuring approximately exponential activity distributions in the networks' units. In addition, we demonstrated that the IP mechanism contributes to rapid learning in the network. Overall, our results suggest that IP may play an important role in the unsupervised learning of sensory representations in the cortex and it underscores the need to carefully study how different forms of neuronal plasticity may interact at the network level.

In the model of visual receptive field development we have used the simple

Hebbian learning rule which was multiplied with a difference of Gaussian function that modulated the sign of Hebbian learning (Hebbian vs. anti-Hebbian) based on a unit's distance to the most activated unit in the map. This implies that units close to the winning unit will strengthen their connections (long term potentiation, LTP) while far away units will weaken their connections (long term depression, LTD). Note that a qualitatively similar effect could be obtained by using a Bienenstock-Cooper-Munro (BCM) learning rule that has LTP and LTD components [88], combined with only an excitatory Gaussian neighborhood function. In future work we would like to explore such alternative learning schemes and also consider the combination with neural fields described by Wilson & Cowan-like dynamics. In addition we would like to construct hierarchical networks to model the development of receptive field properties in higher visual areas.

Acknowledgment

The text of Chapter 2, with some modification, is a reprint of the material as it appears in N.J. Butko and J. Triesch, "Exploring the Role of Intrinsic Plasticity for the Learning of Sensory Representations," *Neurocomputing*, 70(7-9):1130–1138 (2007) [3]. I was the primary author of this publication; the co-author supervised the research that forms the basis for this chapter.

Chapter 3

Visual Salience Model for Robot Cameras

3.1 Abstract

Recent years have seen an explosion of research on the computational modeling of human visual attention in task free conditions, *i.e.*, given an image, predict where humans are likely to look. This area of research could potentially provide general purpose mechanisms for robots to orient their cameras. One difficulty is that most current models of visual salience are computationally very expensive, and therefore not suited to real time implementations needed for robotic applications.

Here we propose a fast approximation to a Bayesian model of visual salience recently proposed in the literature. The approximation can run in real time on current computers at very little computational cost, leaving plenty of CPU cycles for other tasks. We empirically evaluate the salience model in the domain of controlling saccades of a camera in social robotics situations. The goal was to orient a camera as often as possible toward humans. We found that this simple general purpose salience model doubled the success rate of the camera: it captured images of people 70% of the time, when compared to a 35% success rate when the camera was controlled using an open-loop scheme. After 3 saccades (camera

movements), the robot was 96% likely to capture at least one person. The results suggest that visual salience models may provide a useful front end for camera control in robotics applications.

3.2 Introduction

There has recently been a large amount of scientific research to develop computational models of visual salience [42, 43, 58–60, 96–100]. The computational output of these models is a value at each pixel of an image or video sequence (Figure 3.1) that indicates whether that region is likely to be fixated by humans when the task is to simply look at the image or video. Typically these methods are evaluated on how well they predict the actual specific locations that humans have fixated in eye-tracking experiments where the only instruction is "look" or "watch." This area of research is of potential interest to social robotics for two reasons: First, a robot that orients its eyes in a manner similar to humans is likely to give an impression of intelligent behavior and facilitate interaction with humans. Second, such models may orient the robot toward regions of the visual scene that are likely to be relevant.

Unfortunately the currently existing models of visual salience are typically too slow, requiring seconds, if not minutes, to analyze single video frames at very reduced resolution. Here we describe and evaluate a very fast and computationally lightweight adaptation of a recently published model of visual salience. The model can comfortably provide salience maps in about 10 ms per video frame on a modern low-end computer, thus being particularly suitable for robotic applications. We show that the algorithm provides a useful front end for robotics cameras, effectively using peripheral information to orient the camera towards likely regions of interest.

3.3 Previous Models of Visual Salience

Several Bayesian approaches have been developed recently that provide a computational foundation to the notion of visual salience. While at first sight these

Figure 3.1: The purpose of visual salience algorithms is to quantify the importance of attending to each visual location. Salience algorithms are often evaluated on how well they predict humans' eye-fixation data.

models may appear very different from each other, they can be seen as special cases of the same formalism. In particular many of these approaches implicitly or explicitly define the salience of a pixel y as a function of the probability that this pixel renders an object of a category of interest, given the available image. *I.e.*,

$$\begin{aligned} s(y) &\stackrel{\text{def}}{=} \log p(C_y = 1 | f_y) \\ &= \log p(f_y | C_y = 1) + \log p(C_y = 1) \\ &\quad - \log p(f_y) \end{aligned} \tag{3.1}$$

where $s(y)$ is the salience of pixel y and f_y is a feature vector that summarizes the information on image pixels in the neighborhood of y, and C_y is a binary random variable that takes value 1 if pixel y was rendered by an object from the category of interest.

This formulation can be used to compare the choices made by the existing Bayesian approaches. For example, Torralba *et al.* [96] use the $p(C_y = 1)$ term to model class specific location distributions, *i.e.* the density $p(C_y = 1)$ differs for every y depending on the location of y in the image plane. *E.g.*, clouds may be more probable *a priori* toward the top of the image. It can also take on a different value by switching targets, *e.g.* the distribution $p(C_y = 1)$ when searching for clouds is different from $p(C_y = 1)$ when the category of interest is people. They

estimate $p(f_y)$ using a generalized Gaussian fit to the statistics of the specific image being searched.

Bruce & Tsotsos [42] present a model of salience based on the Shannon information of an event, $-\log p(f_y)$. They estimate the density $p(F_y = f_y)$ using a histogram over a small image region, as opposed to the entire image, as in [96]. Their model implicitly assumes that in general purpose tasks the functions $p(f_y|C_y = 1)$ and $p(C_y = 1)$ are approximately constant with respect to y and they can be ignored, since they do not affect the relative salience of different pixel locations.

Harel *et al.* [60] proposed a model of salience based on the use of a dissimilarity metric. Like [42] the context is free-viewing, and the first two terms become irrelevant in ranking pixels. Like [96] the distribution $p(f_y)$ is estimated based on the histogram of the the current image. However, in this case, they use a graphical model that weights inter-pixel distance and feature dissimilarity. Probabilities are estimated by sampling, a process that is $O(n^4)$ with n pixels in the image. While this approach matches human free-viewing data well, it is infeasible for calculating salience maps of moderate size in real time.

Zhang *et al.* [98] follow the model in [96], but estimate $p(f_y)$ using frequency counts from a data set of natural images and videos fit to generalized Gaussian distributions. By using features sensitive to local contrast, they are able to replicate salience effects that in other models require densities to be estimated within each image separately. This makes the model's complexity roughly linear with respect to the number of image pixels, and therefore attractive for real-time implementations, since it does not require recomputing costly frame by frame statistics.

Itti *et al.* [58] proposed a model of visual salience based on the feature integration theory of human attention [101]. Their model computes many features at each pixel by convolving *e.g.* motion, color, and brightness channels with difference of Gaussians (DOG) filters. These are then normalized and half-wave rectified. The different channels are then added together to create a master salience map. Navalpakkam & Itti [97] define visual salience in terms ofsignal to noise ratio. Specifically, the model learns the parameters of a linear combination of low level

features that cause the highest expected signal to noise ratio for discriminating a target from distractors. Itti & Baldi [59] define salience as the number of bits of information that the image features around a pixel give about the process generating those features. Specifically, under their model, salience is related to the number of events generated by a poisson process. A gamma distribution conjugate prior is maintained over the Poisson distribution's parameters. Spatial salience detectors estimate the posterior distribution based on map neighbors, and temporal salience detectors estimate the posterior distribution based on subsequent salience of the same pixel. The model is evaluated in terms of its capacity to fit human saccade data in open ended, free-viewing tasks.

Gao & Vasconcelos [99] define salience as the KL distance between the distribution of a pixel region's filter responses from that of pixels surrounding that region. The distribution of filter responses is estimated as a generalized Gaussian distribution, and a different distribution is fit to each overlapping region of the image.

Kienzle *et al.* [100] used a data-driven approach, using human eye movement data on general purpose tasks to learn features that are highly discriminative of regions that are commonly scanned by humans versus regions with low scanning rates.

3.4 Real-Time Implementation

In this chapter, we propose a simplified version of Zhang *et al.*'s model [43] designed to operate in real time at little computational cost. In [43], Zhang extends the model in [98] to temporally dynamic scenes, and characterizes the video statistics around each pixel using a bank of spatio-temporal filters with separable space-time components, *i.e.*, the joint spatio-temporal impulse response of these filters is the product of a spatial and a temporal impulse response. In [43] the spatial impulse responses are difference of Gaussians (DOG), which model the properties of neurons in the lateral geniculate nucleus (LGN). The surround Gaussian has radius twice the size of the center Gaussian, and each subsequent

scale is twice the size of the previous scale. At the smallest spatial scale, the radius is 1 pixel. The spatial impulse response at scale i is

$$\begin{aligned} g(i) &= \frac{1}{2\pi(2^{i-1})^2}\exp\left(-\frac{h^2+v^2}{2(2^{i-1})^2}\right) \\ &\quad - \frac{1}{2\pi(2^{i})^2}\exp\left(-\frac{h^2+v^2}{2(2^{i})^2}\right) \end{aligned} \tag{3.2}$$

where h and v are the horizontal and vertical distance to the center of the filter. The temporal impulse responses are difference of exponentials (DOE), which can be implemented recursively in a very efficient manner:

$$h(t;\tau) = \hat{h}(t;2\tau) - \hat{h}(t;\tau) \tag{3.3}$$

where $\hat{h}(t;\tau) = \frac{\tau}{1+\tau}\cdot(1+\tau)^t$, $t \in (-\infty, 0]$ is the relative frame number to current frame (0 is the current frame, -1 is last frame, *etc.*) and τ is a temporal scale parameter. The τ of the first scale is a parameter to the model, and it doubles with each successive temporal scale.

The probability distribution of the features $p(f_y)$ is estimated by collecting filter responses over natural videos, fitting a generalized Gaussian distribution for each individual filter, and combining the distribution across temporal and spatial scales assuming conditional independence. For the real-time implementation explored in this chapter, we simplified Zhang's model in the following ways:

1. We used only image intensity channels, not color channels.

2. The DOG filters were approximated by difference of box (DOB) filters (See Figure 3.2).[1]

3. The filter impulse response distribution was modeled as a Laplacian distribution with unit variance, a special case of the generalized Gaussian distribution.[2]

[1] DOB are types of box-filters, a computationally efficient class of filters that have been used with much success recently in visual object classification [63]

[2] In the generalized Gaussian case we have $-\log p(f_y) = \sum_i |f_{y,i}/\sigma_i|^{\theta_i}$. This becomes $-\log p(f_y) = \sum_i |f_{y,i}|$ under our Laplacian with $\sigma_i = 1$ approximation.

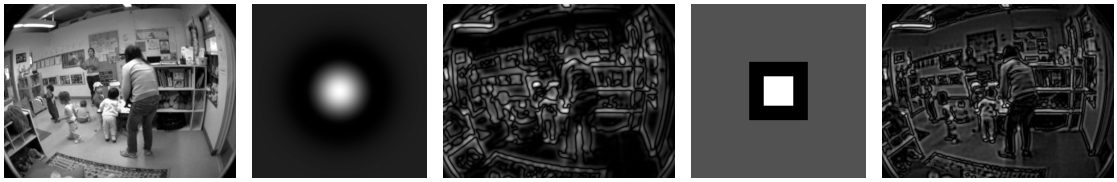


Figure 3.2: Difference of Gaussians filter, and the Difference of Boxes approximation. The filters are typical of those used in this chapter, with the $r_{center} = 1/2\ r_{surround}$. The filters are respectively applied to the original image (left). Absolute filter responses are shown.

As in Zhang's original model, we assume an open-ended visual search task, *i.e.* we don't have prior knowledge about where in an image generally interesting objects will appear, or what they will look like. Under these conditions the location prior $p(C_y = 1)$ and the object appearance model $p(f_y|C_y = 1)$ are constant with respect to y and thus can be ignored.

The approach is pseudocoded in Algorithms 1 & 2. In Algorithm 2, all arithmetic operations are vector operations.

The computational complexity was linear with respect to n, the number of pixels, as well as NS and NT, the number of spatial scales and temporal scales. Tables 3.1 & 3.2 show the time needed to compute salience on a frame varying each of these three complexity dimensions. The computations were performed on a Mac Mini with a 1.87 GHz Intel Core Duo processor. Box filter operations were performed with Apple's vImageBoxConvolve_Planar8 function. Vector algebra operations were performed using the BLAS library. The time was measured in absolute (wall) time, but since the processor was dual core and the process single-threaded, the process-specific times were nearly identical. In practice our implementation is orders of magnitude faster than those reported in the literature. For example, the popular salience model of Itti & Baldi [59] requires ≈ 1 minute for each 30×40 pixel video frame, while the model proposed here takes 11 milliseconds for each 120×160 pixel video frame.

In order to ensure that the simplifications in our approach still maintain the important properties of other visual salience algorithms, we compared its per-

Algorithm 1 Initialize Salience

$NS \Leftarrow 5$ {Parameter: # of Spatial Scales}
$NT \Leftarrow 5$ {Parameter: # of Temporal Scales}
$\text{Min}\sigma \Leftarrow 1$ {Parameter: Smallest Box Filter Radius $\in [1, \infty)$}
$\text{Min}\tau \Leftarrow 1$ {Parameter: Smallest Time Parameter $\in (0, \infty)$}
$\sigma[1] \Leftarrow \text{Min}\sigma$
$\tau[1] \Leftarrow \text{Min}\tau$
for $i = 1$ to NS **do**
    $\sigma[i+1] \Leftarrow 2\sigma[i]$
end for
for $j = 1$ to NT **do**
    $\tau[j+1] \Leftarrow 2\tau[j]$
end for
for all $Exp[i, j]$ **do**
    $Exp[i, j] \Leftarrow \vec{0}$ {Exp has $(NS+1, NT+1)$ vectors the size of the salience map.}
end for

formance to the model of Itti & Baldi [59]. The task was to predict human eye fixation on videos in a free viewing task; the data were those originally used in [59]. The performance of our algorithm (0.633 AROC) was very similar to that of Itti & Baldi (0.647 AROC). This is also comparable with Zhang's original algorithm, and so very little performance is sacrificed making the three approximations above.

Table 3.1: Processing time needed to compute salience map as a function of image size (5 spatial / 5 temporal scales).

	80×60	160×120	320×240	640×480
Time	2.93 ms	10.82 ms	44.96 ms	214.82 ms

Algorithm 2 Calculate Salience $s(y)$

Require: $NS, NT, \sigma, \tau, Exp$ initialized in Algorithm 1. Exp is updated in this Algorithm.

$SalienceMap \Leftarrow \vec{0}$
$Im \Leftarrow$ get downsampled frame from camera
$BoxFilt[1] \Leftarrow$ Filter Im with box-filter, width=$2\sigma[1]+1$
for $i = 1$ to NS **do**
  $BoxFilt[i+1] \Leftarrow$ Filter Im with box-filter, width=$2\sigma[i+1]+1$
  $DOB[i] \Leftarrow BoxFilt[i] - BoxFilt[i+1]$
  $Exp[i,1] \Leftarrow \frac{\tau[1]}{1+\tau[1]} DOB[1] + \frac{1}{1+\tau[1]} Exp[i,1]$
  for $j = 1$ to NT **do**
    $Exp[i,j+1] \Leftarrow \frac{\tau[j+1]}{1+\tau[j+1]} DOB[i] + \frac{1}{1+\tau[j+1]} Exp[i,j+1]$
    $DOE[i,j] \Leftarrow Exp[i,j+1] - Exp[i,j]$
    $SalienceMap \Leftarrow SalienceMap + \text{abs}(DOE[i,j])$
  end for
end for
return $SalienceMap$

3.5 Field Study

As part of the RUBI project [102, 103], our laboratory has conducted field studies with social robots immersed at the Early Childhood Education Center at UCSD. The goal of these studies is to explore the possibilities of social robots to assist teachers in early childhood education (Figure 3.3). One critical aspect of these robots is to be able to find and orient towards humans. Previously, members of the Machine Perception Laboratory developed powerful algorithms for detecting the presence of humans using video [104]. These tend to be computationally expensive and thus best suited for scanning low resolution images, or a small, isolated region of a large scene. As such, we were interested in investigating whether a lightweight salience model could be used on peripheral regions to help orient the fovea towards the most promising regions of the visual scene.

A 2 degree of freedom (pan and tilt) robot camera was constructed using

Table 3.2: Processing time needed to compute salience map over various spatiotemporal scales (160×120 pixels).

Space\Time	1 Scale	2 Scales	3 Scales	4 Scales	5 Scales
1 Scale	1.32 ms	1.64 ms	1.95 ms	2.26 ms	2.82 ms
2 Scales	2.04 ms	2.71 ms	3.36 ms	3.93 ms	4.62 ms
3 Scales	2.81 ms	3.81 ms	4.72 ms	5.90 ms	7.06 ms
4 Scales	3.35 ms	4.65 ms	5.77 ms	7.58 ms	8.95 ms
5 Scales	3.88 ms	5.32 ms	6.77 ms	9.29 ms	10.82 ms

Figure 3.3: Three robot members of the RUBI project. **Left:** QRIO is a humanoid robot prototype on loan from Sony corporation. **Center:** RUBI-1, the first prototype developed at UCSD. **Right:** RUBI-3 (Asobo) the third prototype developed at UCSD. It teaches children autonomously for weeks at a time

an iSight IEEE1394 640x480 camera with a fisheye lens (160° FOV), 2 Hitech HS-322HD servo motors, and a Phidgets servo control card operated by a Mac Mini (1.87 GHz Intel Core Duo). The robot camera was placed in Room 1 of the UCSD's Early Childhood Education Center (ECEC), where the RUBI project is taking place. The camera was located on a bookshelf above the reach of the children (18–24 months old). The system collected data continuously for 9 hours during one day's operation of ECEC, from 7:30am–4:30pm.

Images were processed in real-time. They were received from the camera at 640×480 resolution at approximately 15 FPS (i.e. every 66 msec). For the purpose

Figure 3.4: Experimental Setup: A simple robotic camera (left) collected very wide angle – 160° – images at 640×480 resolution (center) and downscaled them to 160×120 resolution for the purpose of computing a salience map (top right). The camera then rotated – pan/tilt – so that the maximum salience pixel was now in the center of gaze. After movement, a 160×120 snapshot of the center of gaze at full resolution was saved as a foveal representation (bottom right). This fovea was coded offline for the presence of people.

of computing salience, they were downsampled to a 160×120 pixel resolution. A salience map was then computed in six-times-faster-than-real-time for all the pixels (≈ 11 msec, see Table 3.2), using a bank of 5 spatial filters and 5 temporal filters. The DOB spatial filters had odd center widths $\{3, 5, 9, 17, 33\}$ so that they would be defined about a central pixel. The above diameters correspond to radii about the center of $\{1, 2, 4, 8, 16\}$ respectively. The corresponding surround widths were $\{5, 9, 17, 33, 65\}$. The τ temporal parameters were $\{1, 2, 4, 8, 16\}$.

Experimental Camera – Salience Track At the start of each experiment, the camera was moved to a central location.

Starting 30 frames after any camera movement, on each successive frame, if the maximum salience pixel exceeded threshold and its location was more than 10 degrees in either the pan or tilt direction from the current fixation point, the servos would reposition the camera so that the maximum salience pixel in the salience map was now at approximately the center of the image plane.

Salience Tracking Condition

Playback Condition

Figure 3.5: Center of attention (fovea) in salience tracking condition and playback condition. In each case, 18 images were chosen randomly from the whole set, and so the sample is representative. In the salience condition, at least 14 of the randomly chosen images have people. In the playback condition, people are clearly visible in only 6 of the randomly chosen images.

15 frames after a movement was initiated (to allow for the movement's completion), an image of the camera's view was saved. Additionally, a foveal view containing the center 160×120 pixels of the high resolution 640×480 image was saved, simulating the foveal region over which high level but computationally expensive perceptual primitives could operate (*e.g.*, person detection, expression recognition).

Control Camera – Playback An additional control condition was implemented. In this condition, the camera played back, in open-loop, the exact same movements as in the preceding salience-directed movement condition. This served as a control with the same motion statistics as the salience condition, but in which the movements were not caused directly by current events in the world. In addition to preserving the motion statistics, the playback framework served to tie together in the two conditions the implicit prior on the "location of the class of generally interesting objects," or $p(C_y = 1)$ in Equation 3.1. Thus the only difference be-

tween the two conditions was that one was caused by features that were unlikely in natural statistics, *i.e.* ones for which $-\log p(f_y)$ was high.

Each condition ran sequentially for 3 minutes at a time. A pair of conditions salience and playback would take about 6 minutes. There was an additional 3 minute break between cycles. In all, 64 cycles were completed and 4964 images were collected.

3.6 Analysis of results

After the experiment, a subset was chosen randomly and uniformly from all 4964 collected images. The foveal regions of each image in this subset was coded by 4 coders. Two of the coders were investigators in this study, and two were naïve third parties. The coders were instructed to label the number of people they could see in each 160×120 foveal region. The coding was done in a double-blind fashion: the images were ordered randomly across labels and time collected. All coders, including the authors, were given no extra information to indicate which images came from which condition. All coders labeled 1050 images (510 salience condition, 540 playback condition) in the same order.

The average Pearson correlation between the four coders across the 1050 labels was 0.8723. We marked a foveal snapshot as "containing a person" if two or more coders agreed that there was at least one person in the snapshot.

3.6.1 Results

It should be noted that the control condition in our experiment was designed to be particularly difficult, much harder than random search. For example, in the control condition, the camera oriented toward regions of space that had been salient in the experimental condition. These regions tended to have people in the experimental condition and thus were still likely to have people at control time. In spite of this, the experimental camera (salience tracking) performed much better than the control camera (playback). In the salience tracking condition, 68.04% of foveal images contained people. In the playback condition, only 34.81% of foveal

images contained people. Thus, by orienting toward salient events in the image plane, the camera attended to people twice as often as just looking in the places where people are likely to appear. A random sample of images from both conditions is shown in Figure 3.5.

Note that with a detection rate of 68% per saccade, after 3 saccades, we are 96.8% likely[3] to have seen at least one person. A post processing algorithm operating over these saccades would review $(3*160\times120)$ pixels, representing more than an 81% reduction in search time, relative to searching the entire 640×480 image plane.

Most importantly, the salience algorithm is fast and efficient. Salience was calculated in less than 11 ms for each 67 ms frame grab, leaving over 83% of CPU cycles to be dedicated to other tasks important to the function of the robot, including sophisticated visual post-processing.

An additional benefit is derived from salience's resilience to distorted images: it works well on the entire image plane of a very wide angle camera. However, object identification algorithms are often brittle to the warping caused at the edges of the wide angle lens. By using salience on a very wide field of view, we can identify from large regions of the real world areas of interest and then point the center of the lens toward them. Objects in the central region are undistorted, and may be discovered easily by machine perception algorithms.

Although we did not investigate it systematically, the salience algorithm also appears to be robust to lighting conditions. For example, during nap time, the lights of the classroom were turned off, but the robot continued to orient toward teachers walking around the room.

---

[3] Assuming people are always present. This figure is an underestimate and the true rate will be higher given presence of people because this average performance figure includes even times when there are no people to be seen, such as nap time or when children are playing outside.
$96.8\% = 1 - (1 - .68)^3$

3.7 Conclusions

We presented a fast visual salience algorithm that approximates very well current models of early human visual attention. From a Bayesian point of view the algorithm is designed to find regions of an image plane most likely to be useful in unconstrained conditions, *i.e.*, situations where there is a very large number of potential tasks of interest. The proposed approach matches human eye fixation data almost as well as current state of the art models of early visual attention, yet it is orders of magnitude faster. It can operate in real time in a low end modern computer, leaving plenty of CPU for other operations. This makes the approach ideal for robotic applications.

We presented empirical results from a field study using a robotic camera in daily life conditions. To our knowledge this is the first example of a practical use of current models of early human visual attention to a real time robotics task. The results suggested that models of visual salience may provide a promising approach for efficient camera orientation in social robotics applications.

Acknowledgment

The text of Chapter 3, with some modification, is a reprint of the material as it appears in N.J. Butko, L. Zhang, G.W. Cottrell, and J.R. Movellan, "Visual Saliency for Robot Cameras," *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, 2398–2403 (2008) [4]. I was the primary author in this publication.

Part II

Infomax 2: Information Foraging

Chapter 4

Detecting Contingencies: An Infomax Approach

Abstract

The ability to detect social contingencies plays an important role in the social and emotional development of infants. Analyzing this problem from a computational perspective may provide important clues for understanding social development, as well as for the synthesis of social behavior in robots. In this chapter, we show that the turn-taking behaviors observed in infants during contingency detection situations are tuned to optimally gather information as to whether a person is responsive to them. We show that simple reinforcement learning mechanisms can explain how infants acquire these efficient contingency detection schemas. The key is to use the reduction of uncertainty (information gain) as a reward signal. The result is an interesting form of learning in which the learner rewards itself for conducting actions that help reduce its own sense of uncertainty. This chapter illustrates the possibilities of an emerging area of computer science and engineering that focuses on the computational understanding of human behavior and on its synthesis in robots. We believe that the theory of stochastic optimal control will play a key role providing a formal mathematical foundation for this newly emerging discipline.

4.1 Introduction

Peter picks up the phone: “Hello, this is Peter,” he says. A voice responds, “もしもし、ブッコです。” Surprised, Peter repeats, “Hello, this is Peter.” The voice responds, “日本語を話しますか?” Peter says, “I think you are calling the wrong number. Who are you trying to reach?” The voice responds, “分かりません。御免なさい。” Peter did not understand a single word, but he had the distinct impression that there was a person trying to communicate with him at the other end of the line. It did not feel at all like a pre-recorded message.

Infants face situations like this very early in their lives. They do not understand human language, but they still need to identify what entities are responsive to them and when they are so. Developmental psychologists refer to this ability to identify responsive entities as “contingency detection”, “contingency analysis”, “contingency perception”, and “contingency learning”.

There is a large body of evidence suggesting that the ability to detect contingencies plays a crucial role in the social and emotional development of infants [11, 105–108]. For example, it has been hypothesized that infants use contingency, not appearance, as the main cue to detect conspecifics. The appearance of human beings becomes special to infants because they can generate contingencies. This point of view traces back to an experiment conducted by John Watson in 1972. In this experiment, 2-month-old infants learned to move their heads to activate a mobile located above their cribs [11]. Each infant in the experimental group was presented with a mobile that rotated in response to the motion of her head. For the infants in the control group, the mobile moved in a pre-recorded, non-contingent manner. After four daily 10-minute sessions, and an average of 200 total responses, there was evidence that the infants in the experimental group had learned that they could control the mobile. At the same time, these infants displayed a number of powerful social responses towards the mobile, including vigorous cooing and smiling. Essentially, the mobile began functioning as a “social stimulus”. Watson hypothesized that contingency was being used by these infants as a cue to define and identify caregivers.

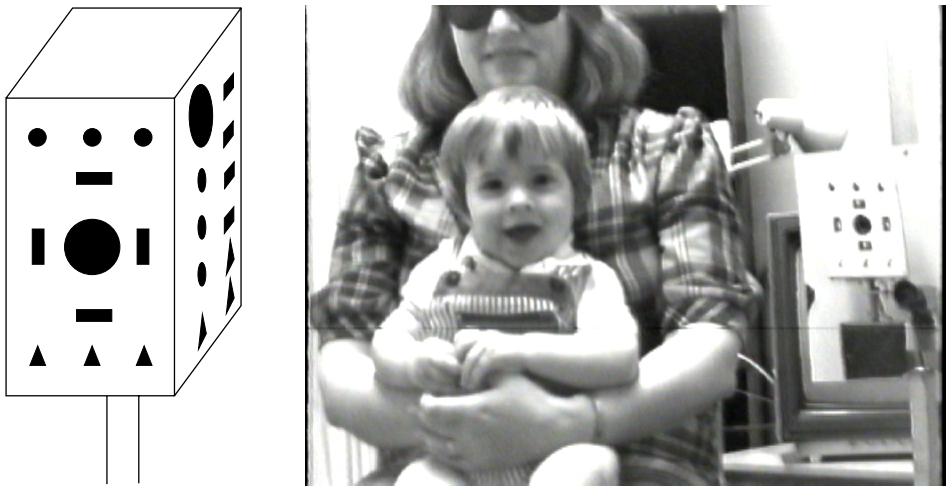


Figure 4.1: Left: schematic of the robot head used my Movellan & Watson. Right: Baby-9. The image of the robot is seen reflected on a mirror positioned behind the baby.

Watson and Movellan [12, 109] conducted a similar experiment with 10-month-old infants. Infants were seated in front of a robot that did not look particularly human. The "head" of the robot was a rectangular prism whose sides contained geometric patterns (see left side of Figure 4.1). The robot could make sounds and turn its head to the right or left. Infants were randomly assigned to an experimental group or a matched control group. In the experimental group, the robot produced sounds in response to the infants' vocalizations. In the control group, the robot reproduced the same responses that had been recorded in the matched experimental session. In this way, infants in the control group experienced exactly the same robot activity, except that it was pre-recorded and not responsive to them. After a few minutes of exposure to the robot, many infants in the experimental group were treating the robot as if it was a social agent: they produced 5 times more vocalizations than the infants in the control group, and they followed the "line of regard" of the robot when it rotated [12, 109]. Similar results were later replicated with 12-month-old children [110].

Particularly striking was the quality of interactions that were observed in some infants in the experimental group: (1) Their vocalizations toward the robot

appeared to be like questions. Each vocalization was followed by 5 to 7 seconds of silence, during which the infants seemed to be actively waiting for an answer from the robot. (2) After a few such vocalizations and less than a minute into the experiment, most observers report that these infants know that the robot is responding to them.

The video of one such baby, hereafter named "Baby-9," will be the focus of this document. This video is available at `http://mplab.ucsd.edu/baby9` and is an essential companion to this thesis. The reader is recommended to watch this video to better understand the focus of this chapter, as well as later chapters. Most people that watch the video report that Baby-9 has clearly detected the responsiveness of the robot. Many of these additionally indicate that Baby-9 is actively querying the robot, as if questioning whether or not it is responsive.

Challenge Problems: Understanding the pattern of behavior that Baby-9 exhibited poses theoretical challenges with important consequences for the scientific study of social development in infants:

1. What does it mean to "ask questions" for an organism like Baby-9 that does not have language?

2. Was it smart for Baby-9 to schedule his vocalizations in the way that he did?

3. Was it smart for him to decide within a few responses and less than a minute into the experiment that the robot was responsive?

4. What mechanisms can explain the transition from the relatively slow learning that Watson observed in 2-month-old infants to the very fast and active learning that was observed in 10-month-old infants like Baby-9?

In this chapter, we explore a computational approach to these theoretical questions based on the framework of stochastic optimal control. Originally developed by engineers to control complex systems like airplanes and industrial robots, stochastic optimal control is giving behavioral scientists a unifying theory to describe diverse human skills such as reaching, walking, eye movements, and concept learning [6, 53, 111–114]. We propose that the same framework can be

used to understand the development of social interaction. In particular, the behavior observed in Baby-9's video can be seen as a sensorimotor schema optimized for gathering information as to whether or not a social contingency is present.

In the chapter we show how social skills, like the ones observed in Baby-9, could be acquired using standard reinforcement learning mechanisms. The key for this to happen is to use information as an intrinsic reward. This opens the possibility that the same mechanisms that are used to learn how to reach, walk, and look, could also be used to acquire social skills, including the development of symbolic communication.

A long term goal of this work is to illustrate how stochastic optimal control may be used to provide a computational basis for the study of human development. The approach provides a modern alternative to behaviorist approaches that were popular in the first half of the 20^{th} century, and to cognitive/mentalist approaches that dominated in the second half. We aim for the approach illustrated in this document to provide a computational basis to help bridge the study of the brain, the study of development, and the synthesis of intelligent behavior in robots.

4.2 Stochastic Optimal Control

Due to the inherent variability of situations that organisms encounter through their lives, biological motion can seldom rely on a predetermined sequence of actions. Instead, the behavior of organisms is more like a dance with the environment, in which sensory information is continuously polled to generate actions that are tuned to the current state of the world. Influential developmental psychologists, such as Piaget, have long argued that these sensorimotor schema provide the primordial conditions out of which high-level cognitive processes develop.

Control theory is a rigorous mathematical formalism for analyzing the sensorimotor dance between complex systems and the environment. Its focus is solving the problem of how to map sensory information into motor commands to generate intelligent behavior in real time. To give the reader a better intuition for the control theory formalism, we present a simple example. The point of this example is

to illustrate the different elements of the control theory formalism. Refer to this chapter's appendix, Section 4.8.1 for information on mathematical notation and conventions.

Simple Control Theory Example – Reaching: Consider a robot who is trying to reach for an object as quickly as possible, while using as little energy as possible. To analyze this scenario in the language of control theory, we must specify the relevant states x_t that the robot can encounter, actions u_t that the robot can take to affect the state, observations y_t that the robot can use to get feedback about its progress, and the goal ρ that the robot is trying to achieve. In each of these, the momentary nature of the dance with the environment is captured by the subscript t, denoting that each element can and does constantly change.

For this problem, the relevant state x_t consists of the current angles between each of the robot's joints. The robot affects these angles by applying voltages u_t to each of its motors. The relationship between voltages and changing joint angles is captured in the world dynamics, also known as system dynamics, given by the electro-mechanic equations of motion. This is defined by a probability distribution $p(x_{t+1} \mid x_t, u_t)$ that specifies probable next states x_{t+1} given current states x_t and actions u_t. By expressing this relationship as a probability distribution, the robot can express the natural variability in the voltages it sends, as well as unpredictable external perturbations, such as people grabbing its arm.

The robot gets feedback y_t about its progress from sensors, such as encoders that measure the angle at each joint. The sensor model $p(y_{t+1} \mid x_{t+1})$ describes the encoders' readings given particular joint configurations.

The joint angles x_t determine the position p_t of the robot hand in 3D Euclidean space. The robot's goal of touching a target at a position p^* can be specified using a reward function that measures the Euclidean distance between the current position of hand and the desired posture. In addition, we could penalize actions that consume too much energy. For example, the reward could take the following form:

$$r_t = -\|p_t - p^*\|^2 - k\|u_t\|^2 \tag{4.1}$$

where $k \geq 0$ is a constant that penalizes for using too much energy.

Given such a problem specification, the theory of stochastic optimal control provides algorithms to find optimal "control laws". These are also known as "policies", or simply "controllers." Controllers are the technical equivalent of the sensorimotor schemas that Piaget discussed. Formally, a control policy c is a collection of functions $c = (c_1, c_2, c_3...)$ indexed by time t. Each function c_t maps the history of data H_t available to the robot to an action U_t to be taken by the robot:

$$U_t = c_t(H_t) \tag{4.2}$$

The information history H_t consists of everything the robot has seen and done prior to taking an action at time t. This includes the entire history of actions $U_1 \cdots U_{t-1}$ and the entire history of sensor values $Y_1, \cdots, Y_t$, *i.e.*,

$$H_t = (U_1, \cdots, U_{t-1}, Y_1, \cdots Y_t) \tag{4.3}$$

Stochastic optimal control is essentially a computational theory of intentional, goal oriented behavior. The goals are specified using a reward variable R_t that represents the desirability of states and actions at particular points in time. The overall goal of the controller is typically expressed as a weighted sum of the expected accumulation of future rewards:

$$\rho(c) = \sum_{t=1}^{\tau} \alpha_t E[R_t \mid c] \tag{4.4}$$

where τ is the temporal horizon, or terminal time. Controllers are evaluated in terms of the expected reward gathered before the terminal time. Depending on the situation, this terminal time can be finite, or infinite. The α_t terms are non-negative constants that modulate the relative importance of rewards at different points in time. Stochastic optimal control considers the problem of finding control policies c that optimize the goal function $\rho(c)$.

Stochastic optimal control has been traditionally applied to optimization of physical goals (*e.g.*, maintaining a motor's velocity under variable loads, regulating a room's temperature, and making smart weapons). In this document we show how the same approach also illuminates the development of social behavior from a computational point of view.

4.3 Formalizing the Contingency Detection Problem

In order to analyze the contingency problem within the stochastic optimal control framework, we must formalize it with the same elements as the motor control problem described above: states, actions, observations, system dynamics, sensor models, and goal. Our formalization was inspired by John Watson's contingency detection model [106], in which background noise and responsive caregivers are modeled as Poisson processes. While Watson focused on the inference problem, *i.e.*, the development of algorithms to infer the presence or absence of contingency given a history of sensorimotor experiences h_t, we focus on the control problem, *i.e.*, how to schedule behaviors in real-time to ensure that sensorimotor experiences h_t are as informative as possible in a limited period of time. We will investigate

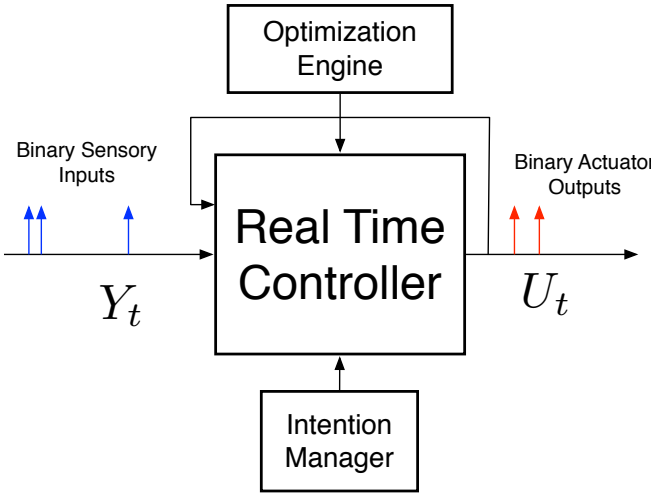


Figure 4.2: A bare-bones social robot

the problem of detecting social contingency from the point of view of a bare-bones baby robot (see Figure 4.2). This idealized baby robot has a single binary sensor and a single binary actuator. The sensor tells the robot whether a sound is present, and the actuator produces vocalizations. There will be two players: (1) a *social agent* that plays the role of the caregiver, and (2) a *baby robot* that plays

the role of the infant. The agent and robot are in situated an environment with random background activity. When the social agent is present, she responds to the sounds produced by the baby robot, introducing a contingency between the robot's actuator and the robot's sensor. Our goal is to find an optimal control policy for the baby robot to detect, as efficiently and accurately as possible, whether or not such a contingency is present and, by extension, whether the social agent is present. While at first sight this may appear to be a simple problem, the following complications need to be considered:

- **Self-feedback:** When the robot makes a sound, the sensor will register the sound with some delay, creating spurious contingencies.
- **Variability in background conditions:** If the baby robot is in a noisy room, the sensor will be frequently active. If it is in a quiet room, the sensor will be seldom active. The baby robot needs to consider the level of background activity when deciding whether or not a social agent is present.
- **Variability in social agents' responsiveness:** Social beings are highly unpredictable, with different individuals having different levels of responsiveness. The baby robot needs to consider the potential levels of activity of the agent when deciding whether or not an agent is present.

These considerations point to three causal factors that activate the baby robot's sensor: (1) self-feedback, (2) background activity independent of the robot, and (3) responsive social agents. The baby robot may find itself in one of two possible situations, or *contingency clusters*, which we identify with the following names: "responsive agent absent," and "responsive agent present" (see Figure 4.3). When the robot makes a sound and no responsive agent is present, the robot's auditory sensor will activate for a period of time due to self-feedback. Afterward, the sound sensor becomes active at random times due to background activity. In addition to the self-feedback and background periods, there is a critical period of time during which social agents will respond to the robot's sounds, but only if a responsive agent is present (see Figure 4.3).

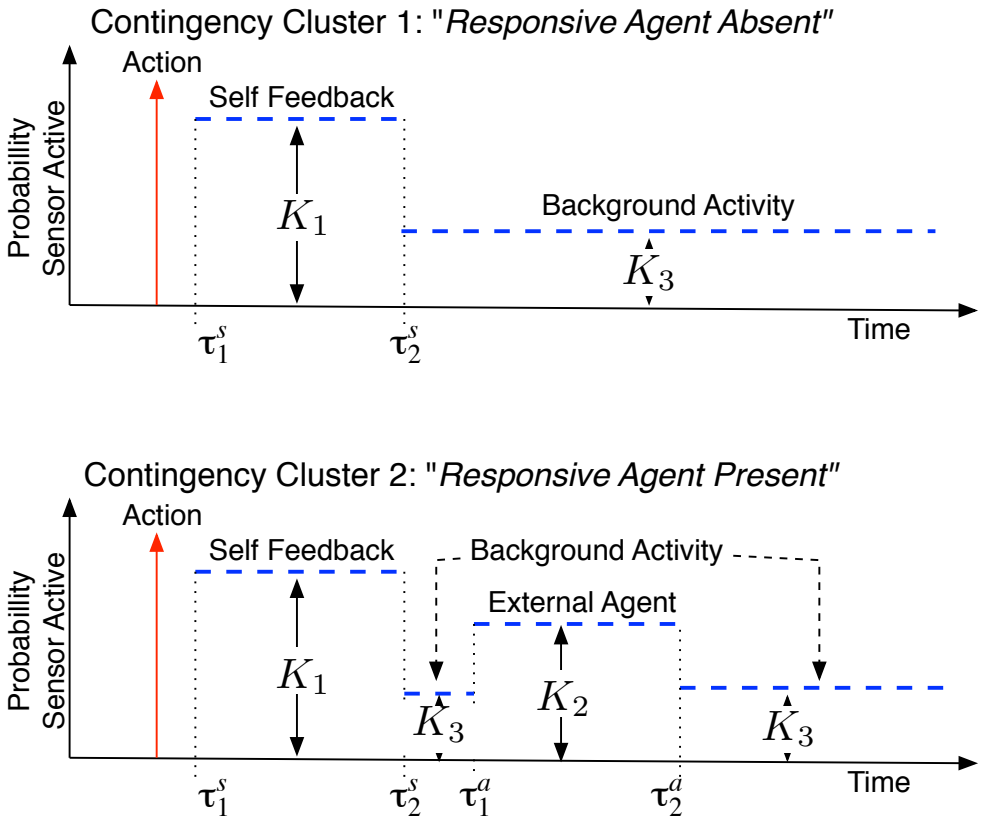


Figure 4.3: Illustration of two contingency clusters produced by the model. The variable S indicates which of the two clusters is active in the current situation.

4.3.1 State, Action, Observation, System Dynamics, and Sensor Model:

The state X_t of the baby robot's environment contains five relevant variables: S, Z_t, K_1, K_2, K_3. The first variable S encodes whether a responsive agent is present ($S = 1$) or absent ($S = 0$). The second variable Z_t is a timer that encodes the amount of time since the baby robot's last vocalization. This timer determines which of three periods the baby robot is in: (1) a self-feedback period, which occurs immediately after a sound is made, (2) a critical period, during which social agents are more likely to respond to the last baby robot vocalization, and (3) a background period, unlikely to contain responses to the last vocalization. These three time periods are defined by the parameters $0 \leq \tau_1^s \leq \tau_2^s < \tau_1^a \leq \tau_2^a$.

If the timer Z_t is between τ_1^s and τ_2^s, then the robot is in its self-feedback period. If the timer takes a value between τ_1^a and τ_2^a, then the system is in the critical period, during which social agents are likely to respond to the last vocalization. If Z_t is larger than τ_2^a, then the observed sounds are unlikely to be related to the last baby robot vocalization (See Figure 4.3). The last three state variables K_1, K_2, K_3 are real-valued numbers that represent the expected rates of sensor activity during self, agent, and background periods. The state variables S, K_1, K_2, K_3 are assumed to be static. The timer variable Z_t increases by one on each time step until the baby robot vocalizes, at which point it resets to 1.

The action U_t represents the activation of the robot's sound actuator (*e.g.*, a loudspeaker). At each moment, the baby robot can choose to vocalize ($U_t = 1$), meaning that it will activate the loudspeaker at time t. Otherwise it can choose to not vocalize, *i.e.*, deactivate the loudspeaker ($U_t = 0$). By choosing the "vocalize" action, the baby robot is implicitly choosing to reset the timer Z_t that governs the unfolding of natural, social turn-taking behavior. By choosing the "don't vocalize" action, the baby robot is choosing to let the timer run its course.

We let Y_t represent the activation of the baby robot's sound sensor (*e.g.*, a microphone). $Y_t = 1$ indicates that the sound level is larger than a fixed threshold θ, otherwise $Y_t = 0$. At each time step the sensor activates in a probabilistic manner. The probability that it becomes active is determined by K_1, K_2, K_3. If the timer Z_t is such that the system is in the self-feedback period, then the probability of activation is K_1. If Z_t is such that the system is in the critical period of agent response, then the probability of activation is K_2. Otherwise the system is in the background period and the probability of activation is K_3. If an agent is present and responding ($S = 1$) then K_2 and K_3 will be different. If an agent is not present ($S = 0$), then the agent and background activity rates are the same, *i.e.*, $K_2 = K_3$

Under this model, the problem of detecting that a responsive agent is in the room is equivalent to the problem of detecting whether the background time K_3 and agent time K_2 rates of sensory activation are different.

4.3.2 Inference Process

The baby robot is assumed to follow an optimal probabilistic inference process. The specifics of this process are explained in the appendix of this chapter. For now, it suffices to say that at every point in time t, this process correctly determines the probability $p(S \mid h_t)$ that a social agent is responding given the history of vocalizations and sounds h_t. If this probability is close to 0.5, the robot is uncertain about the presence or absence of a contingency. If $p(S = 1 \mid h_t) \approx 1$, the robot is quite certain that a responsive agent is present. If $p(S = 1 \mid h_t) \approx 0$, the robot is quite certain a responsive agent is not present. A common measure of the level of uncertainty about a random variable is the entropy, in bits, of the probability distribution of that variable, *i.e.*,

$$\mathcal{H}(S \mid h_t) = -\sum_{s=0}^{1} p(s \mid h_t) \log_2 p(s \mid h_t) \tag{4.5}$$

For example, if $p(S = 1 \mid h_t) = 0.5$ then the entropy is 1 bit (high uncertainty). If $p(S = 1 \mid h_t) = 0.99$ or $p(S = 1 \mid h_t) = 0.01$ then the entropy is 0.08 bits (low uncertainty).

4.3.3 Goal: Information Maximization

The goal of the baby robot is to gather as much information as possible and as quickly as possible about S, *i.e.*, about the presence or absence of a social contingency. We call control policies that are optimized for the goal of information gathering "information maximization controllers" (infomax controllers for short).

Suppose by time t, the robot has access to the history h_t of sensor data and actions performed up to that time. A natural way to define an infomax controller is to let the reward at time t be equal to the amount of information that h_t provided about S, *i.e.*,

$$r_t = \mathcal{I}(S, h_t) \tag{4.6}$$

where $\mathcal{I}$ is the mutual information operator, an information theoretic quantity that corresponds to the intuitive notion of "information about" (see this chapter's

appendix). Mutual information encodes the amount of information that the history of observed data h_t provides about the state S. This information can be expressed as a difference of entropies,

$$\mathcal{I}(S, h_t) = \mathcal{H}(S) - \mathcal{H}(S \mid h_t) \tag{4.7}$$

where $\mathcal{H}(S)$ is the initial uncertainty (entropy) about S, *i.e.*, how uncertain the baby robot is about whether or not a social agent is present and responding *before it has done or heard anything.* This initial uncertainty is a constant independent of the available data and thus independent of the controller. $\mathcal{H}(S \mid h_t)$ is the uncertainty about S given the available data h_t. This value depends on the data history h_t, and therefore on the controller. Since $\mathcal{H}(S)$ is independent of the controller, we can ignore it and simply use the following reward function:

$$R_t = -\mathcal{H}(S \mid h_t) \tag{4.8}$$

This reward function promotes controllers that choose vocalizations that lead the baby robot to have high confidence (low entropy) about S. From a pure infomax standpoint, Baby-9 didn't necessarily care that the social agent was responding to him. Instead, he cared about knowing whether or not it was responding to him. He would be just as happy after discovering that the unresponsive outcome was the correct one, just so long as he was confident in that discovery.

This brings us to the first challenge problem:

- What does it mean to "ask questions" for an organism like Baby-9 that does not have language?

From an infomax point of view, questions are behaviors that are expected to provide information about variables of interest. We hypothesize that Baby-9 was asking about the state variable S: "Is that thing out there responding to me?" To say that Baby-9 was asking questions about the state S means that his vocalizations helped him resolve his uncertainty about S. To say that he was asking *good* questions about S means that his vocalizations helped him resolve his uncertainty about S as quickly as possible. In order to analyze whether Baby-9 was doing something smart, *i.e.* asking good questions, we must first find the optimal controller, and then compare Baby-9's behavior with that of the optimal controller.

4.4 Optimal Infomax Controller for Detecting Social Contingencies

4.4.1 Model Parameters

The model described above has the following parameters:

- Δ_t: The sampling period used to discretize time.
- $\tau_1^s \leq \tau_2^s < \tau_1^a \leq \tau_2^a$: Latency parameters that determine the self-feedback period, the period for agent likely responses, and the background period.
- θ: The threshold used to binarize the output of a sound sensor.
- π: The probability that an agent is present, prior to collecting any data.
- The time horizon τ over which the controller optimizes the information reward.

In order to set these parameters to reasonable values we conducted a study with four people that played the role of caregivers. They were presented with a humanoid robot that made sounds at randomly selected intervals. The participants were asked to treat the robot as if it were a baby, and to respond verbally to the sounds it made. The ages of the participants were 4, 6, 24, and 35 years. Each participant interacted independently with the robot for a five minute period. During this time, the robot vocalized at random intervals and the participants responded to it in the way that was most natural to them.

There were a total of 150 trials, during which the vocalizations of the robot and participants were digitized. Each trial started with a vocalization of the robot and ended 4 seconds later. The sound intensity threshold θ was chosen automatically by applying a k-means clustering procedure to the digitized sound data.

Figure 4.4 shows the probability of activation of the binarized sound sensor as a function of time over 150 trials. The first peak in activity of the sound sensor is due to self-feedback, *i.e.*, the sensor is recording its own sound. This peak occurs at 360 ms, indicating a delay between the time at which the program told

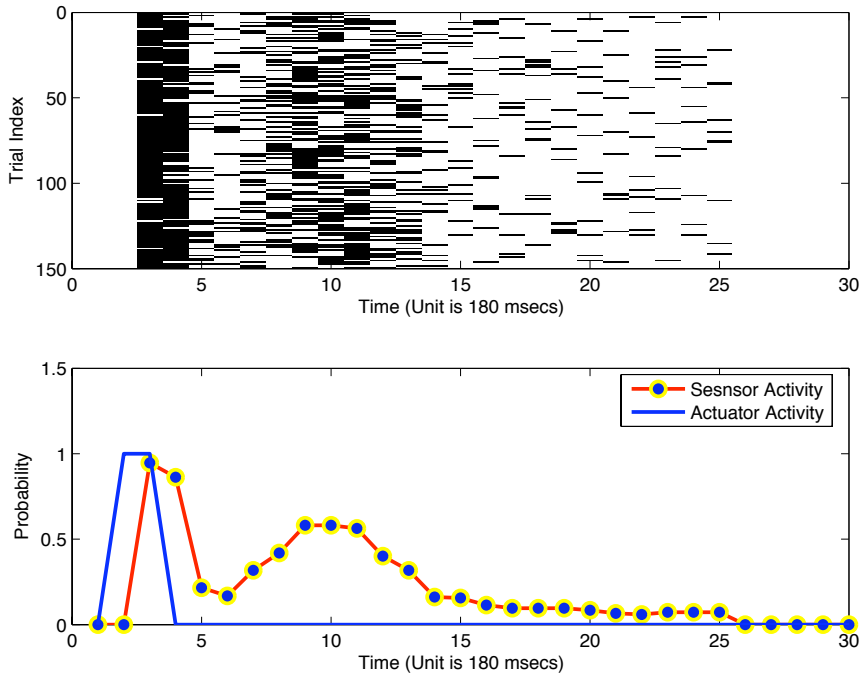


Figure 4.4: Top: Raster plot of 150 trials. On each trial a robot made a sound and subjects were asked to talk back to the character and let it know that they were listening. Dark indicates that the audio sensor was active. Bottom: Probability of the audio sensor being active as a function of time. The probabilities are estimated by averaging across the 150 trials in the raster plot.

the robot to make a sound, and the time at which the sound was detected by the sound sensor. By about 1300 ms after the end of the robot's vocalization, there is a second, smaller peak of activity in the sensor, which is now caused by the vocalizations of the human participants.

We chose Δt large enough to make self-feedback delays negligible, thus fixing $\tau_1^s, \tau_2^s = 0$, but small enough to capture the behaviors of interest. We found $\Delta t = 800$ ms to be a good compromise. The limits for the agent activation intervals were set to $\tau_1^a = 1, \tau_2^a = 3$, *i.e.*, 800 ms and 2400 ms respectively. The prior probability that an agent is present was set to $\pi = 0.01$, thereby requiring a significant amount of data to become convinced that an agent is present. The

reason for choosing this conservative value for π is explained below. The time horizon parameter τ was set to 40 time steps, *i.e.*, 32 seconds. This value was chosen because, at the time, it was the longest horizon for which we could compute an optimal controller in a reasonable amount of time. As we explain later, we found that after approximately 12 time steps (approximately 10 seconds) the controllers stabilize. This indicates that it does not pay to use horizons longer than 10 seconds in situations governed by the statistics of social interaction.

4.4.2 Computation and Analysis of the Optimal Controller

Infomax control is a specific instance of a general class of control problems known as partially observable Markov decision processes (POMDPs). In infomax control, information gain acts as a reward signal. The utility function optimized by the controller is the long term gathering of information about states of the world that are not directly observable. While finding exact solutions to infomax control problems is generally difficult [1], in this particular case there is a recursive statistic A_t that summarizes the observable data history without any loss of information. This allowed us to find an optimal controller using standard dynamic programming algorithms [45] (see appendix III of this chapter).

The solution found using dynamic programming was a large lookup table that mapped each possible statistic a_t of the sensorimotor history h_t into a binary action u_t. Such a lookup table is provably optimal for every possible state, but it doesn't give us much intuition about which features of the sensorimotor history were important for making the optimal decision. In order to gain a better understanding of how the controller solved the problem, we developed a simple model that was evaluated on its ability to predict what the optimal controller would do next. We focused on the behavior of the controller for time steps $18 \leq t \leq 24$, because these are times that are not too close to the beginning and end of the controller's window of interest. We found that the following control policy matched the action of the optimal controller with 98.5% accuracy over all possible data

history conditions:

$$c_t(h_t) = \begin{cases} 1 & \text{if } Z_t > \tau_2^a \text{ and } \frac{\mathrm{Var}(K_2 \mid h_t, S_t=1)}{n_{2,t}} > 9\, \frac{\mathrm{Var}(K_3 \mid h_t, S_t=1)}{n_{3,t}} \\ 0 & \text{else} \end{cases} \tag{4.9}$$

where Z_t is the time since the last vocalization of the robot, and $\mathrm{Var}(K_2|h_t, S_t = 1)$ is the current uncertainty (variance) about K_2, the sensor activation rate during the critical period in which social agents respond to the robot's vocalizations. $\mathrm{Var}(K_3 \mid h_t, S_t = 1)$ is the current uncertainty (variance) about K_3, the sensor activation rate during background noise periods, *i.e.*, periods under which social agents are unlikely to respond to the last vocalization of the robot. The denominators dividing the variances indicate the total number of time steps collected up to date for the agent period ($n_{2,t}$) vs. the background period ($n_{3,t}$). Dividing the variance by the number of observations accounts for how much the variance can be expected to reduce further with new observations.

Thus, the optimal controller always waits at least τ_2^a seconds, the longest period of time under which agents are likely to respond, before making a new vocalization. In addition, it does not vocalize unless it is significantly more uncertain about the rate of sensor activation during the critical period of social response than about the rate of activation during background periods. The effect is to homeostatically keep the uncertainty about the agent interval and the uncertainty about the background interval at a fixed ratio. If the agent rate is too uncertain, then the controller chooses to vocalize, thereby earning an opportunity to learn more about the rate of the agent intervals. If the background rate is too uncertain, then the controller chooses to remain silent, thereby gaining information about background intervals.

Notably, for a vocalization to occur, the uncertainty about the sensor activation rate K_3 during the agent period has to be at least 9 times larger than the uncertainty about the rate during the background period K_2. This may be due to the fact that vocalizations are more costly, in terms of information return, than silent periods. If the baby robot chooses to vocalize at time t, it gains no information during the times $[t+\tau_1^s, t+\tau_2^s]$ since self-feedback observations are not

informative about S. In addition, during times $[t + \tau_1^a, t + \tau_2^a]$ the controller instructs the robot not to act and thus during those periods the robot can only gain information about K_2, not K_3. By contrast if the robot chooses to remain silent at time t, no time will be wasted due to self-feedback. Moreover the robot can still choose to act or not to act in the future without constraints. This helps explain why uncertainty about the agent activity rate K_2 needs to be much larger than the uncertainty about the background activity rate, K_3, before an action occurs.

Note that "greedy" one-step controllers [53, 111] that seek as much information reward as possible immediately, at the expense of future expected rewards, would fail on this task. The reason is that when the baby robot chooses to vocalize, its self-vocalization prohibits it from getting any information about K_2 or K_3 temporarily, while it would still get a small amount of information about K_3 by choosing to remain silent. Thus a greedy controller ends up deciding to never vocalize. Looking into the future allows the baby robot to conclude that vocalizing periodically provides a better long term information return than always choosing silence.

4.4.3 Comparison with the Behavior of Baby-9

We compared the behavior of the optimal infomax controller described above to the behavior observed in the video of Baby-9. This video lasts 43 seconds, during which Baby-9 produced 7 vocalizations. The first vocalization occurred 5.58 seconds into the experiment. The intervals, in seconds, between the beginning of two consecutive infant vocalizations were as follows: $\{4.22, 10.32, 5.32, 6.14, 5.44, 3.56\}$. Most observers report that Baby-9 clearly has detected that there is a responsive agent in the room by the end of the 43 seconds.

We ran the optimal controller with a receding time horizon of 24 time steps (19.2 seconds), *i.e.*, at each point in time the controller behaved so as to maximize the expected information to be gained over a period of 19.2 seconds into the future. As in the Baby-9 experiment, every time the baby robot's controller made a sound, it was given a response, simulating a social agent. Figure 4.5 shows the result of the simulation.

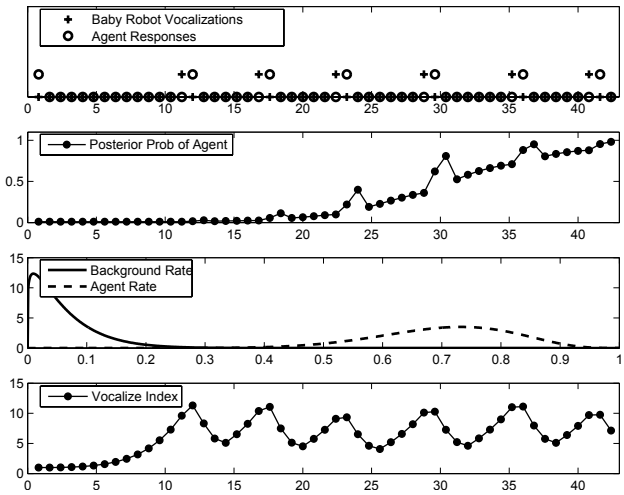


Figure 4.5: The horizontal axis represents time in seconds. From top to bottom: (1) Responses of the infomax controller (which simulates a baby). Note that the social agent responded every time the baby robot vocalized, but otherwise the environment was silent. (2) Posterior probability for the presence of a responsive agent as a function of time. (3) Posterior distribution for the agent and background rates after 43 seconds. (4) Ratio of the uncertainty about the agent's response rate vs the uncertainty about the background's response rate.

The top graph shows the vocalizations of the optimal controller, which serves as a model of Baby-9. The infomax controller exhibited turn-taking behaviors that were very similar to the ones observed in Baby-9: the infomax controller makes a sound and follows it by a period of silence as if waiting for the outcome of a question.

This turn-taking behavior was not built into the system. Instead, it emerged from the requirement to maximize information gain given the time delays and levels of uncertainty typical in social interactions.

The controller produced six vocalizations over a period of 43 seconds. The average interval between vocalizations was 5.92 seconds which is remarkably close to the average of 5.83 seconds of silence between vocalizations for Baby-9. There

seems to be a tendency both in the model and in Baby-9 for the early silence intervals to be longer than the later ones.

This provides an answer to the second challenge problem in the introduction to this chapter:

- Was it smart for Baby-9 to schedule his vocalizations in the way that he did?

Baby-9's behavior was smart in the sense that he asked good questions: questions that helped to quickly resolve his uncertainty about whether the rectangular prism in front of him was actually a contingent social agent. In fact, Baby-9's pattern of vocalizations and silences was very close to optimal. This also explains the sense of intentionality that most people intuitively perceive when they watch the video of Baby-9. The behavior of Baby-9 makes a great deal of sense if one were to assume that his goal is to discover whether social agents are responsive to him.

The second graph from the top in Figure 4.5 shows the system's beliefs about the presence of a responsive agent. These beliefs are updated in real time using standard Bayesian inference (see this chapter's appendix). In our simulation, we chose a conservative prior probability $\pi = 0.01$ for the presence of social contingency to force the controller to gather a significant amount of data before deciding that there is a social contingency present. Note that in spite of this conservative prior, by the end of the 43 seconds, the posterior probability that there is a responsive agent is very close to 1. The third graph shows the posterior probability distributions about the agent and background response rates by the end of the 43-second period. Note these two distributions are very different, consistent with the idea that there is indeed a responsive agent present.

This provides an answer to the third challenge problem in the introduction:

- Was it smart for him to decide within a few responses and less than a minute into the experiment that the robot was responsive?

Given the statistics of social interaction, it was indeed very smart for Baby-9 to decide within a few responses and less than a minute into the experiment that a social contingency was present.

Finally, the last graph in Figure 4.5 shows the ratio between the uncertainty about K_2, the sensor rate during agent periods, and the uncertainty about K_3, the sensor rate during background periods. Note that when this ratio reaches the value of 9, the optimal controller vocalizes.

4.5 Learning to Detect Contingencies

In the previous section, we used standard dynamic programming algorithms to find an optimal infomax controller. We found that this model appeared to describe well the turn taking behaviors observed in some 10-month-old infants when they are trying to detect the presence of social contingency. This begs the question: how did these infants acquire a policy for finding social contingency that is so close to optimal?

One possibility is that children are born with these policies. The differences in contingency detection efficiency found between 2-month-old infants and 10-month-old infants may be due to the maturation of brain structures. Just like teeth mature to allow more efficient chewing, some brain structures may be specially programmed by evolution to mechanistically mature into a machine for more efficient detection of contingency.

Another possibility is that children are born with something akin to a dynamic programming algorithm that allows them to find the optimal controller. The advantage of dynamic programming is that it finds controllers guaranteed to be optimal. However the dynamic programming hypothesis has several drawbacks: (1) it requires detailed and precise knowledge of the system dynamics and observation model; (2) it is very time and memory intensive; (3) it is not easily implementable on neural-like hardware; (4) it provides no mechanisms to benefit from experience interacting with the world.

An alternative to both pre-programmed controllers and dynamic programming is reinforcement learning (RL). RL is an area of machine learning and control in which the goal is to learn control policies that approximate the solutions given by dynamic programming without requiring detailed and precise knowledge of the

system dynamics [30]. RL is easily implementable in neural-like hardware and provides a natural set of mechanisms to make good use of experience and interaction in the world. Unfortunately, RL itself has drawbacks. While dynamic programming requires a good deal of time, memory, and computational effort, RL requires many trial and error experiences to learn efficient policies. In a sense, the difficulty of the computation is offloaded to the world around the robot, and to interaction with its environment. The amount of experience required in some cases is so great that RL cannot be considered a plausible model of learning in a developmentally reasonable time frame.

4.5.1 Infomax RL Results

In this section, we consider whether the optimal contingency detection strategies observed in some 10-month-old infants could be explained as the manifestation of an RL process driven by an information based reward system (infomax RL). To demonstrate the computational plausibility of the infomax RL hypothesis, it suffices to show that at least one RL algorithm can learn within a developmentally plausible period of time. We chose this time frame to be 60,000 vocalizations, which was meant to be a conservative ballpark estimate, based on 200 vocalizations per day of the infant's first 10 months of life.

We implemented infomax RL using temporal difference (TD) learning, a popular RL algorithm that has been shown to have correspondences in the pattern of dopamine release from neurons in the basal ganglia [29] (see appendix, Section 4.8.4).

Empirically, we found that the number of vocalizations needed for the TD learning algorithm to converge grew as a fifth-power of the temporal horizon τ. Convergence within 60,000 vocalizations was only achievable with horizons of 12 time steps (10 seconds) or less. A horizon of 16 time steps required 230,000 vocalizations, and a horizon of 20 time steps required 700,000 vocalizations, which is much higher than our estimate of a reasonable developmental time-frame.

We then investigated the question of how 10-second controllers compare to optimal controllers with longer time horizons. Given the statistics of social

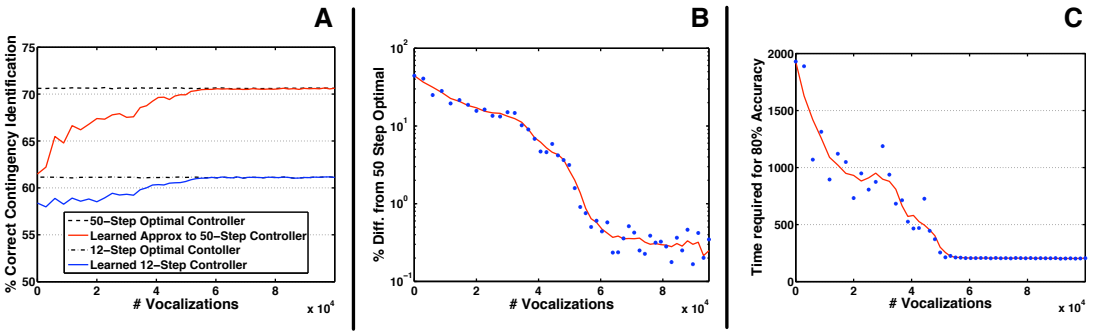


Figure 4.6: **A:** Performance of infomax TD Learning in the finite horizon (12-step), and receding-horizon (50-step) case, based on the total number of vocalizations made since birth. **B:** When a receding horizon controller with 6.5 seconds of memory and a 3.5 second deadline is used to approximate an optimal controller with a perfect memory and much longer deadline, the final information gathering performance is nearly identical. **C:** The number of time steps spent acting, exploring, and listening to the world that are required to achieve 80% social agent identification accuracy.

interaction, does it pay off to use time horizons longer than 10 seconds?

Fifty new simulations were performed, with a time horizon of 12 time steps. As expected, on average, infomax RL converged after less than 60,000 vocalizations. We then used dynamic programming to compute optimal 12-step and 50-step controllers in order to serve as evaluation standards for the controller learned from experience. The performance of the optimal 12-step controllers found using dynamic programming (an exact method) was identical to the 12-step controllers found using infomax RL (an approximate method), indicating that infomax RL converged to an optimal solution.

To compare the learned controller to the 50-step optimal controller, we adopted a receding horizon approach: the 12-step learned controller was artificially limited to eight time steps of memory (about 6.5 seconds), and then chose the action that would help it gather as much information as possible in the next four time steps (about 3.5 seconds). This limited memory controller, which had learned

from experience and information reward over a simulated ten month time frame, was almost as good as the performance of the optimal 50-step controller: after 60,000 vocalizations, the average performance was better than 99.5% of the optimal performance (see Figure 4.6 A & B).

This provides an answer to our fourth and final challenge problem:

- What mechanisms can explain the transition from the relatively slow learning that Watson observed in 2-month-old infants to the very fast and active learning that was observed in 10-month-old infants like Baby-9?

Simple reinforcement learning algorithms, in which uncertainty reduction is used as a reward signal, are a plausible mechanism to explain how infants improve on their capacity to detect contingencies. In 10 months of simulated experience, infomax RL agents perform 99.5% as well as the best possible controller.

4.6 Real-Time Robot Implementation

Once computed, the optimal infomax policy can be applied to sensor data in real time, trivially, on any modern computer. To test how well this policy would work in real life, we implemented it on RobovieM, a humanoid robot developed at ATR's Intelligent Robotics and Communication Laboratories. While the robot was not strictly necessary to test the real-time controller, it greatly helped improve the quality of the interactions developed between humans and machines, thereby providing a more realistic method for testing the controller.

For the binary sensor, we chose to average acoustic energy over 500 ms windows and binarize it using the threshold θ that was found by applying a k-means algorithm to the acoustic portion of the natural interaction data that were collected previously. The actuator was a small loudspeaker producing a 200 ms robotic sound. The self-feedback delay parameters of the controller were chosen by measuring the time delay between issuing a command to produce a sound and receiving feedback from the audio sensor. The agent delay parameters were the same as in the simulation of Baby-9.

The robot was programmed to change its posture based on the controller's belief about the presence/absence of a responsive agent: a posture that indicated a high level of attention when the controller believed that an agent was present, and a posture that indicated boredom when it believed that an agent was not present.

Overall, the infomax controller was remarkably effective in a wide range of environments, and it required very little computational and sensory resources. In standard office environments, with relatively high levels of noise, the controller reliably detects within 3 or 4 vocalizations whether or not a responsive agent is present. We have demonstrated this system at both scientific talks and poster sessions. Demonstrations at talks, which generally have relatively low noise levels, work very well. During poster sessions, the rooms are typically very noisy, but it only takes a few more vocalizations for the controller to gather enough information to make make reliable decisions. The level of performance is remarkable considering the difficulty of these adverse conditions, and the simplicity of the sensors being used.

4.7 Conclusions.

There is evidence that the ability to detect social contingencies plays an important role in the social and emotional development of infants [11, 105–108]. Analyzing this problem from a computational perspective provided important clues for understanding social development in infants and for the synthesis of social behavior in robots. We framed our analysis of contingency detection within the theory of stochastic optimal control. In particular, we formulated contingency detection as a control problem in which the goal is to gather information as efficiently as possible about the presence or absence of contingencies.

A popular model of the social contingency detection problem describes social agents and background noise as Poisson processes [106]. We showed that under this model, the optimal information gathering policy exhibits turn-taking behaviors very similar to the ones found in some 10-month-old infants: vocalizations followed by periods of silence of about 6 seconds. The results suggest that some

10-month-old infants have an exquisite understanding of the statistics of social interaction and have acquired efficient policies to operate in this world. Even though these infants lack a language, they are already asking questions: They schedule their vocalizations in a manner that maximizes the expected information return given the temporal statistics of social interaction.

One of our goals was to explore to what extent social development can be bootstrapped from simple perceptual and learning primitives so that it can be synthesized in robots. For example, our approach does not require high level conceptual primitives, such as the concept of people or the idea that people have minds. In our model, the terms "responsive agent present" and "responsive agent absent" are just mnemonic labels for contingency clusters that may not correspond to categories easily describable with words. Indeed, in John Watson's original experiment [11], 2-month-old infants seemed to group together responsive caregivers and contingent mobiles.

We showed that simple temporal difference reinforcement learning mechanisms could explain how infants acquire the efficient social contingency detection strategies observed in some 10-month olds. The key is to use the reduction of uncertainty (information gain) as a reward signal. The result is an interesting form of learning in which the learner rewards itself for conducting actions that help reduce its own sense of uncertainty. Traditional models of classical and operant learning emphasize the role of external reward stimuli, like food or water. The brain is probably set up to recognize these stimuli and to encode them as rewarding because it is advantageous to do so. Infomax control suggests that it may also be similarly advantageous for organisms to recognize uncertainty and to encode the reduction of uncertainty as rewarding. There is some evidence that the brain may indeed reward reduction in uncertainty with the same mechanisms that it rewards food or water. It has been found that dopamine-releasing neurons located in the substantia nigra pars compacta and ventral tegmental area play an important role in reward based learning [29, 115, 116]. Initially the activity of these neurons was studied for basic forms of reward, such as food and water. However, in recent years it has been found that the same neurons that signal the expected amount of

physical rewards, like food or water, also signal expected information gain. Thus it appears, that information gain may indeed have a special status as an intrinsic motivational reward in the brain [66].

The long range goal of this work is to illustrate the possibilities of a science of behavior and development that is anchored on rigorous computational analysis. As proposed by David Marr [2, 117], the goal of computational approaches is to help understand the problems faced by the brain, as well as the solutions it finds, when operating in everyday life. This approach offers a modern alternative to the behaviorist and the mentalist/cognitive approaches that dominated psychology in the 20^{th} century.

Computational analysis has proven to be a very useful tool for the study of the brain. Our hope is to illustrate that it may also prove useful to understand social development, and to synthesize it in robots. It is remarkable that, after all these years, neither the behaviorist nor the cognitive/mentalist traditions in psychology have significantly contributed to the synthesis of intelligent behavior. We believe that stochastic optimal control may provide a formal mathematical foundation for an emerging area of computer science and engineering that focuses on the computational understanding of human behavior, and on its synthesis in robots.

4.8 Appendices

4.8.1 Appendix I: Definitions and Conventions

Unless otherwise stated, capital letters are used for random variables, small letters for specific values taken by random variables, and Greek letters for fixed parameters. When the context makes it clear, we identify probability functions by their arguments: *e.g.*, $p(x, y)$ is shorthand for the joint probability mass or joint probability density that the random variable X takes the specific value x and the random variable Y takes the value y. We use subscripted colons to indicate sequences: *e.g.*, $X_{1:t} \stackrel{\text{def}}{=} \{X_1 \cdots X_t\}$. We work with discrete time stochastic processes, with the parameter $\Delta t \in \mathbb{R}$ representing the sampling period. We use E for expected values and Var for variance. The symbol $\sim$ indicates the distribution of random variables. For example $X \sim \text{Poisson}(\lambda)$ indicates that X has a Poisson distribution with parameter λ. We use $\delta(\cdot, \cdot)$ for the Kronecker delta function, which takes value 1 if its two arguments are equal, otherwise it takes value 0.

- Beta Variables:

$$X \sim \text{Beta}(\beta_1, \beta_2) \tag{4.10}$$

$$p(x) = \text{Beta}(x, \beta_1, \beta_2) = \frac{\Gamma(\beta_1 + \beta_2)}{\Gamma(\beta_1)\Gamma(\beta_2)} \, (x)^{\beta_1 - 1}(1 - x)^{\beta_2 - 1} \tag{4.11}$$

$$\text{E}(X) = \frac{\beta_1}{\beta_1 + \beta_2} \tag{4.12}$$

$$\text{Var}(X) = \frac{\beta_1 \beta_2}{(\beta_1 + \beta_2)^2(\beta_1 + \beta_2 + 1)} \tag{4.13}$$

  where Γ is the Gamma function

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt \tag{4.14}$$

- Entropy:

$$\mathcal{H}(Y) = -\int p(y) \log p(y) \, dy \tag{4.15}$$

- Conditional Entropy:

$$\mathcal{H}(Y \mid x) = -\int p(y|x) \log p(y \mid x)\, dy \tag{4.16}$$

$$\mathcal{H}(Y \mid X) = -\int p(x, y) \log p(y \mid x)\, dxdy \tag{4.17}$$

$$= \int p(x)\mathcal{H}(Y \mid x)dx \tag{4.18}$$

- Mutual Information: The information about the random variable Y provided by the specific value x from the random variable X is defined as follows:

$$\mathcal{I}(Y, x) = \mathcal{H}(Y) - \mathcal{H}(Y \mid x) \tag{4.19}$$

The average information about the random variable Y provided by the random variable X is defined as follows

$$\mathcal{I}(Y, X) = \sum_x p(x)\, \mathcal{I}(Y, x) = \mathcal{H}(Y) - \mathcal{H}(Y \mid X) \tag{4.20}$$

4.8.2 Appendix II: Summary of the Contingency Detection Model

Parameters: The extended version of the model has 15 parameters:

$$\begin{aligned}
&\Delta t \in \mathbb{R}. \quad \text{Sampling period in seconds.}\\
&\pi \in [0, 1]. \quad \text{Prior probability.}\\
&0 \leq \tau_1^s \leq \tau_2^s. \quad \text{Delay parameters for self-feedback loop.}\\
&\tau_2^s < \tau_1^a \leq \tau_2^a. \quad \text{Delay parameters for social agents.}\\
&(\beta_{i,1}, \beta_{i,2}), i = 1, 2, 3. \quad \text{Parameters for Beta Prior distribution.}\\
&\theta. \quad \text{Threshold for binarizing auditory signal.}\\
&\tau. \quad \text{Time horizon.}
\end{aligned} \tag{4.21}$$

For the simulations presented in this chapter, we worked with a simplified model with 5 parameters: $\Delta t, \tau_1^a, \tau_2^a, \theta, \pi$. We choose Δt large enough to make delays in the onset of self-feedback to be negligible, thus fixing $\tau_1^s, \tau_2^s = 0$, but small

enough to capture the behaviors of interest. We found $\Delta t = 800$ ms to be a good compromise. The values of τ_1^a were set based on a pilot study described in the main body of this chapter: $\tau_1^a = 1, \tau_2^a = 3$, *i.e.*, 800 and 2400 ms respectively. In the simplified model, we treat the agent and background response rates as random variables with uninformative priors, thus fixing the β parameters to 1. We chose $\pi = 0.01$ thus making the prior probability for the presence of agents small, requiring large likelihood ratios to become convinced that an agent is present. The sound threshold θ was chosen using a k-means maximum entropy procedure on the statistics of the available sound. We chose the largest temporal horizon $\tau = 40$ for which we could compute an optimal controller using traditional dynamic programming approaches. Later investigation showed that longer time horizons do not significantly change the optimal policy.

Static Random Variables:

$$S \sim \text{Bernoulli}(\pi). \quad \text{Presence/Absence of Responsive Agent} \tag{4.22}$$

$$K_1 \sim \text{Beta}(\beta_{1,1}, \beta_{1,2}). \quad \text{Sensor activity rate during self period.} \tag{4.23}$$

$$K_2 \sim \text{Beta}(\beta_{2,1}, \beta_{2,2}). \quad \text{Sensor activity rate during agent period} \tag{4.24}$$

$$K_3. \quad \text{Sensor activity Rate during background period} \tag{4.25}$$

$$K_3 \sim \text{Beta}(\beta_{3,1}, \beta_{3,2}), \quad \text{if } S = 1 \tag{4.26}$$

$$K_3 = K_2, \quad \text{if } S = 0 \tag{4.27}$$

Stochastic Processes:

The following processes are defined for $t = 1, 2, \cdots$

$$\text{Timer:} \quad Z_t \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } U_{t-1} = 1 \\ Z_{t-1} + 1 & \text{if } U_{t-1} = 0 \text{ and } Z_{t-1} \leq \tau_2^a \\ Z_t & \text{else} \end{cases}$$

$$\text{Indicator of Self Period:} \quad I_{1,t} = \begin{cases} 1 & \text{if } Z_t \in [\tau_1^s, \tau_2^s] \\ 0 & \text{else} \end{cases}$$

$$\text{Indicator of Agent Period:} \quad I_{2,t} = \begin{cases} 1 & \text{if } Z_t \in [\tau_1^a, \tau_2^a] \\ 0 & \text{else} \end{cases}$$

$$\text{Indicator of Background Period:} \quad I_{3,t} = (1 - I_{1,t})(1 - I_{2,t})$$

$$\text{Self Driver:} \quad D_{1,t} \sim \text{Poison}(K_1)$$

$$\text{Agent Driver:} \quad D_{2,t} \sim \text{Poison}(K_2)$$

$$\text{Background Driver:} \quad D_{3,t} \sim \text{Poison}(K_3)$$

$$\text{Robot Sensor:} \quad Y_t = I_t \cdot D_t$$

$$\text{Robot Controller:} \quad C = (C_1, \cdots, C_\tau)$$

$$\text{Robot Actuator:} \quad U_t = C_t(Y_{1:t}, U_{1:t-1})$$

$$\text{Sensor Activity Counters:} \quad P_{i,t} = \sum_{s=1}^{t} I_{i,s} Y_s \text{ for } i = 1, 2, 3$$

$$\text{Sensor Inactivity Counters:} \quad Q_{i,t} = \sum_{s=1}^{t} I_{i,s}(1 - Y_s) \text{ for } i = 1, 2, 3$$

4.8.3 Appendix III: Detailed Model Description

The model presented in this section was inspired on John Watson [106] formulation of the social contingency detection problem: Background and responsive caregivers are modeled as Poisson processes. Caregivers respond within a fixed window of time from the last response from the baby. Watson focused on the inference problem, *i.e.*, how to make decisions given the available data. Here we

focus on the control problem, how to schedule behaviors in real-time to optimally gather data.

Self-Feedback Processes

We let the robot sensor respond to its own actuators, *e.g.*, the robot can hear its own vocalizations, and allow for delays and uncertainty in this self-feedback loop. In particular we let the distribution of self-feedback delays be uniform with parameters $\tau_1^s \leq \tau_2^s$. The indicator variable for self-feedback period is thus defined as follows:

$$I_{1,t} = \begin{cases} 1 & \text{if } Z_t \in [\tau_1^s, \tau_2^s] \\ 0 & \text{else} \end{cases} \tag{4.28}$$

During Self periods, the activation of the sensor is driven by the discrete time Poisson process $\{D_{1,t}\}$ that has rate K_1, *i.e.*,

$$p(D_{1,t} = 1) = K_1 \tag{4.29}$$

Social Agent Process

The parameters $0 \leq \tau_1^a \leq \tau_2^a$ bound the reaction times of social agents *i.e.*, it takes agents anything from τ_1^a to τ_2^a time steps to respond to an action from the robot. "*Agent periods*", which are designated by the indicator process $\{I_{2,t}\}$ are periods of time for which responses of agents to previous robot actions are likely if an agent were to be present. The indicator variable for an agent period is as follows

$$I_{2,t} = \begin{cases} 1 & \text{if } Z_t \in [\tau_1^a, \tau_2^a] \\ 0 & \text{else} \end{cases} \tag{4.30}$$

During agent periods, the robot's sensor is driven by the Poisson process $\{D_{2,t}\}$ which has rate K_2, *i.e.*,

$$p(D_{2,t} = 1) = K_2 \tag{4.31}$$

The distribution of K_2 depends on whether or not a responsive agent is present. If an agent is present, *i.e.* $S = 1$, we let K_2 be independent of K_1 and K_3 and endow

it with a prior Beta distribution with parameters $\beta_{2,1}, \beta_{2,2}$ reflecting the variability in response rates typical of social agents. If an agent is not present, *i.e.*, $S = 0$, then the response rate during agent periods is the same as the response rate during background periods, *i.e.*, $K_2 = K_3$.

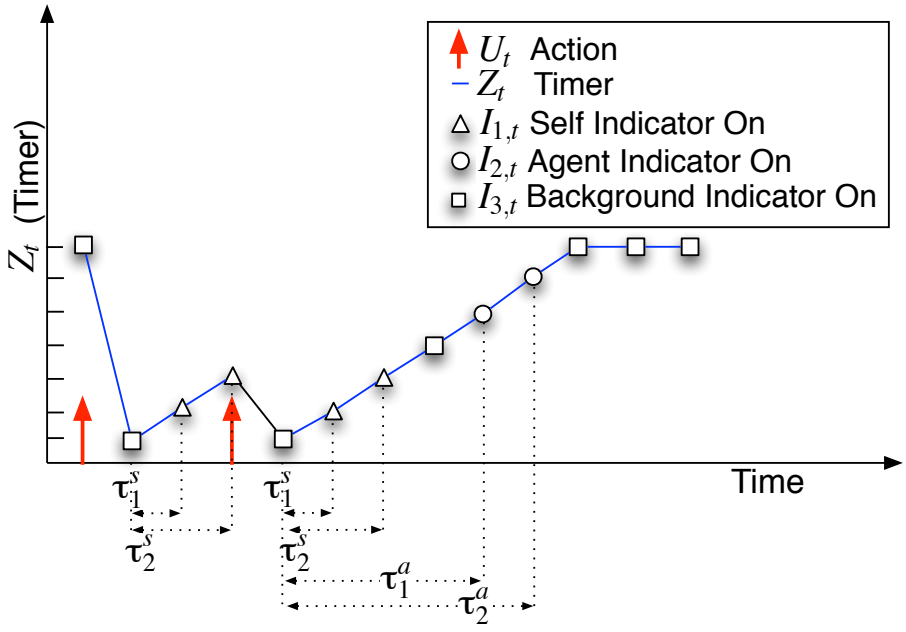


Figure 4.7: Graphical representation of the dynamics of the timer and the indicator variables.

Background Process

The background is modeled as a Poisson process $\{D_{3,t}\}$ with rate K_3, *i.e.*,

$$p(D_{3,t} = 1) = K_3 \tag{4.32}$$

The background drives the sensor's activity that is not due to self-feedback and is not due to social agent responses. Note this can include, among other things, the actions from external social agents who are not responding to the robot (*e.g.*, two social agents may be talking to each other thus activating the robot's sound sensor). We let the background rate K_3 have a prior Beta distribution with parameters

$\beta_{3,1}, \beta_{3,2}$ reflecting the variability of background activity from situation to situation. If $\beta_{3,1} = \beta_{3,2} = 1$ the distribution is uninformative, *i.e.*, all responsiveness rates are equally possible a priori:

$$K_3 \sim \text{Beta}(\beta_{3,1}, \beta_{3,2}) \tag{4.33}$$

The background indicator keeps track of periods for which self-feedback or responsive actions from a social agent may not happen, *i.e.*,

$$I_{3,t} = (1 - I_{1,t})(1 - I_{2,t}) \tag{4.34}$$

Sensor Model

The activity of the sensor is a switched Poisson process: during self-feedback periods it is driven by the Poisson process $\{D_{1,t}\}$, during agent periods it is driven by $\{D_{2,t}\}$ and during background periods it is driven by $\{D_{3,t}\}$, *i.e.*,

$$Y_t = I_t \cdot D_t = \sum_{i=1}^{3} I_{i,t}\ D_{i,t} \tag{4.35}$$

Auxiliary Processes

We will use the processes $\{P_t, Q_t\}$ to register the sensor activity, and lack-thereof, up to time t during self, agent and background periods. In particular for $t = 1, 2, \cdots,$

$$P_{i,t} = \sum_{s=1}^{t} I_{i,s} Y_s, \text{ for } i = 1, 2, 3 \tag{4.36}$$

$$Q_{i,t} = \sum_{s=1}^{t} I_{i,s}(1 - Y_s) \text{ for } i = 1, 2, 3 \tag{4.37}$$

Constraints

Figure 4.8 displays the Markovian constraints in the joint distribution of the different variables involved in the model. An arrow from variable X to variable Y indicates that X is a "parent" of Y. The probability of a random variable is

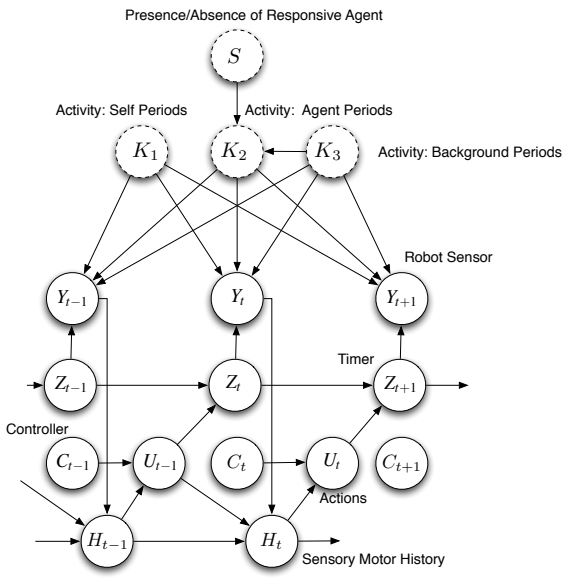


Figure 4.8: Graphical representation of the model. Arrows represent dependency relationships between variables. Dotted figures indicate unobservable variables, continuous figures indicate observable variables. The controller C_t maps all the observed information up to time t into the action U_t. The effect of the action depends on the presence or absence of a responsive agent S and on the timing of the action as determined by Z_t. The goal is to maximize the information return about the actual value of S

conditionally independent of all the other variables given the parent variables. Dotted figures indicate unobservable variables, continuous figures indicate observable variables.

Optimal Inference

Let $(y_{1:t}, u_{1:t}, p_t, q_t, z_t)$ be an arbitrary sample from $(Y_{1:t}, U_{1:t}, P_t, Q_t, Z_t)$. Then

$$p(y_{1:t} \mid k, u_{1:t}, s) = \prod_{i=1}^{3} (k_i)^{p_{i,t}} (1 - k_i)^{q_{i,t}} \tag{4.38}$$

Note the rate variables K_1, K_2, K_3 are independent under the prior distribution. Moreover, if $S = 1$, then they affect the sensor at non intersecting periods of

time. It follows that the rate variables are also independent under the posterior distribution. In particular

$$p(k \mid y_{1:t}, u_{1:t}, S = 1) = \prod_{i=1}^{3} \text{Beta}(k_i; \beta_{i,1} + p_{i,t}, \beta_{i,2} + q_{i,t}) \tag{4.39}$$

If the null hypothesis is correct, *i.e.*, $S = 0$, then $K_2 = K_3$, *i.e.*, the probability distribution of sensor activity during the "agent periods" is the same as during background periods. Moreover, the set of times for which the sensor's activity depends on K_2, K_3 does not intersect with the set of times for which it depends on K_1. Thus K_1 will be independent of K_2, K_3 under the posterior distribution:

$$\begin{aligned} p(k \mid y_{1:t}, u_{1:t}, S = 0) =& \text{Beta}(k_1; \beta_{1,1} + p_{1,t}, \beta_{1,2} + q_{1,t}) \\ & \text{Beta}(k_2; \beta_{2,1} + p_{2,t} + p_{3,t}, \beta_{2,2} + q_{2,t} + q_{3,t})\, \delta(k_2, k_3) \end{aligned} \tag{4.40}$$

Note for an arbitrary k such that $p(k \mid y_{:t}, u_{1:t}, s) > 0$ we have that

$$\begin{aligned} p(y_{1:t} \mid u_{1:t}, s) &= p(y_{1:t} \mid k, u_{1:t}, s)\, \frac{p(k \mid u_{1:t}, s)}{p(k \mid y_{1:t}, u_{1:t}, s)} \\ &= p(y_{1:t} \mid k, u_{1:t}, s)\, \frac{p(k)}{p(k \mid y_{1:t}, u_{1:t}, s)} \end{aligned} \tag{4.41}$$

Thus

$$p(y_{1:t} \mid u_{1:t}, S = 1) = \prod_{i=1}^{3} \left((k_i)^{p_{i,t}} (1 - k_i)^{q_{i,t}} \frac{\text{Beta}(k_i; \beta_{i,1}, \beta_{i,2})}{\text{Beta}(k_i; \beta_{i,1} + p_{1,t}, \beta_{i,2} + q_{i,t})} \right) \tag{4.42}$$

$$= \prod_{i=1}^{3} \frac{\Gamma(\beta_{i,1} + \beta_{i,2})}{\Gamma(\beta_{i,1})\Gamma(\beta_{i,2})} \frac{\Gamma(\beta_{i,1} + p_{i,t})\, \Gamma(\beta_{i,2} + q_{i,t})}{\Gamma(\beta_{i,1} + \beta_{i,2} + p_{i,t} + q_{i,t})} \tag{4.43}$$

and

$$\begin{aligned} p(y_{1:t} \mid u_{1:t}, S = 0) = & \frac{\text{Beta}(k_1; \beta_{1,1}, \beta_{1,2})}{\text{Beta}(k_1; \beta_{1,1} + p_{1,t}, \beta_{1,2} + q_{1,t})} \\ & \frac{\text{Beta}(k_3; \beta_{3,1}, \beta_{3,2})}{\text{Beta}(k_3; \beta_{3,1} + p_{2,t} + p_{3,t}, \beta_{3,2} + q_{2,t} + q_{3,t})} \quad \prod_{i=1}^{3} (k_i)^{p_{i,t}} (1 - k_i)^{q_{i,t}} \end{aligned} \tag{4.44}$$

$$\begin{aligned} &= \frac{\Gamma(\beta_{1,1} + \beta_{1,2})}{\Gamma(\beta_{1,1})\Gamma(\beta_{1,2})} \frac{\Gamma(\beta_{1,1} + p_{1,t})\, \Gamma(\beta_{1,2} + q_{1,t})}{\Gamma(\beta_{1,1} + \beta_{1,2} + p_{1,t} + q_{1,t})} \frac{\Gamma(\beta_{3,1} + \beta_{3,2})}{\Gamma(\beta_{3,1})\Gamma(\beta_{3,2})} \\ & \frac{\Gamma(\beta_{3,1} + p_{2,t} + p_{3,t})\, \Gamma(\beta_{3,2} + q_{2,t} + q_{3,t})}{\Gamma(\beta_{3,1} + \beta_{3,2} + p_{2,t} + p_{3,t} + q_{2,t} + q_{3,t})} \end{aligned} \tag{4.45}$$

where we used the fact that $k_2 = k_3$ with probability one under $S = 0$. Thus the likelihood ratio between the two hypothesis is as follows:

$$L_t(p_t, q_t) = \frac{p(y_{1:t} \mid u_{1:t}, S = 1)}{p(y_{1:t} \mid u_{1:t}, S = 0)} = \frac{\Gamma(\beta_{2,1} + \beta_{2,2})}{\Gamma(\beta_{2,1})\Gamma(\beta_{2,2})} \frac{\Gamma(\beta_{2,1} + p_{2,t})\,\Gamma(\beta_{2,2} + q_{2,t})}{\Gamma(\beta_{2,1} + \beta_{2,2} + p_{2,t} + q_{2,t})}$$
$$\frac{\Gamma(\beta_{3,1} + p_{3,t})\,\Gamma(\beta_{3,2} + q_{3,t})}{\Gamma(\beta_{3,1} + \beta_{3,2} + p_{3,t} + q_{3,t})} \frac{\Gamma(\beta_{3,1} + \beta_{3,2} + p_{2,t} + p_{3,t} + q_{2,t} + q_{3,t})}{\Gamma(\beta_{3,1} + p_{2,t} + p_{3,t})\,\Gamma(\beta_{3,2} + q_{2,t} + q_{3,t})} \tag{4.46}$$

The posterior odds, which is the product of the prior odds and the likelihood ratio,

$$\frac{p(S = 1 \mid y_{1:t}, u_{1:t})}{p(S = 0 \mid y_{1:t}, u_{1:t})} = L(p_t, q_t) \frac{\pi}{1 - \pi} \tag{4.47}$$

contains all the information available to the robot about the presence of a responsive agent.

Infomax Control

The goal in infomax control is to find controllers that provide as much information as possible about a random variable of interest S. Suppose we have a fixed controller c under which we have observed the history of sensorimotor data $h_t = (y_{1:t}, u_{1:t-1})$. The information about the random variable S provided by the observed sequence is as follows:

$$\mathcal{I}(S, h_t) = \mathcal{H}(S) - \mathcal{H}(S \mid h_t) \tag{4.48}$$

The prior uncertainty $\mathcal{H}(S)$ does not depend on the observations, and thus it will be the same regardless of the controller c. Thus, if our goal is to gain information about S, then we can use as reward function the negative of the entropy of S given the observed sequence h_t, *i.e.*,

$$r_t \stackrel{\text{def}}{=} \mathcal{H}(S \mid h_t) \tag{4.49}$$

The value of a controller is expressed as a weighted sum of the expected accumulation of future rewards, up to a terminal time τ:

$$\rho(c) = \sum_{t=1}^{\tau} \alpha_t E[R_t \mid c] = \sum_{t=1}^{\tau} \alpha_t \mathcal{H}(S \mid Y_{1:t}, U_{1:t-1}) \tag{4.50}$$

where the $\alpha_t \geq 0$ are fixed numbers representing the relative value of information return at different points in time.

The controller c_t maps the information history $h_t = (y_{1:t}, u_{1:t-1})$ that is available prior to taking the action into the action taken at that time, *i.e.*,

$$u_t = c_t(h_t) \tag{4.51}$$

The information history is Markovian and the reward is a function of the information history. Therefore, infomax control is a Markov Decision process with respect to the information history. Unfortunately, the number of possible observable sequences grows exponentially as a function of time, making it very difficult to use standard optimal control algorithms for horizons beyond a few time steps. In particular each action and each observation is binary, , *i.e.*, for any given time t there are 2^{2t} separate state histories that must be learned. Fortunately the observation history can be summarized by a statistic A_t consisting of 5 integers: The number of time steps since the last vocalization, the number of active and the number of inactive observations during the periods of agent and background states, *i.e.*,

$$A_t \stackrel{\text{def}}{=} (Z_t, P_{2,t}, P_{3,t}, Q_{2,t}, Q_{3,t}) \tag{4.52}$$

The statistic A_t has the following properties

1. It is a recursive function

$$A_{t+1} = f_t(A_t, U_t, Y_{t+1}) \tag{4.53}$$

2. The predictive distribution of Y_{t+1} is conditionally independent of H_t given A_t, U_t, *i.e.*,

$$p(y_{t+1} \mid h_t, u_t) = p(y_{t+1} \mid a_t, u_t) \tag{4.54}$$

3. The expected reward is conditionally independent of the observed sequence given the statistic of the sequence,

$$\mathrm{E}[R_k \mid h_t, u_t] = \mathrm{E}[R_k \mid a_t, u_t] \tag{4.55}$$

Given these properties, infomax control can be expressed as a Markov decision process where the state is given by the statistic A_t. This allows for solving the Bellman equations using standard dynamic programming and reinforcement learning approaches.

4.8.4 Appendix IV: Infomax TD Learning

We used the following finite horizon version of value based TD(0) learning. For each state a_t of the A_t statistic, and for each time $t = 1, \cdots, \tau$, we initialize the value estimates $V_t(a_t)$ to zero, which is an optimistic value. Each learning trial starts at time $t = 1$ and ends at the terminal time τ. At time $t = 1$ we draw s, k_1, k_2, and k_3 from their prior distributions, and initialize a_1 to $\{Z = z_1, P_2 = Q_2 = P_3 = Q_3 = 0\}$, where z_1 is drawn from the uniform probability distribution over the range $1 : \tau_2^a + 1$. Then for $t = 2, \cdots, \tau$, we choose with probability $(1 - \epsilon)$ the action $\hat{u}_t$ that maximizes the expected value:

$$\hat{u}_t = \underset{u_t}{\operatorname{argmax}} \sum_{y_{t+1}} p(y_{t+1} \mid a_t, u_t) V_{t+1}(a_{t+1}) \tag{4.56}$$

where

$$a_{t+1} = f_{t+1}(a_t, u_t, y_{t+1}) \tag{4.57}$$

With probability ϵ we choose the other action. After each trial, we perform backups to the value estimates $V_t(a_t)$ of each visited state a_t, in reverse order, according to the following equation:

$$V_t(a_t) = r_t + \sum_{y_{t+1}} p(y_{t+1} \mid a_t, u_t) V_{t+1}(a_{t+1}) \tag{4.58}$$

where

$$r_t = -\mathcal{H}(S \mid a_t) \tag{4.59}$$

For the terminal time τ we simply let

$$V_\tau(a_\tau) = r_\tau = -\mathcal{H}(S \mid a_\tau) \tag{4.60}$$

The update equations are repeated for multiple trials. As the number of trials increases, the estimate of the value function $V_t(a_t)$ converges to its true value. At evaluation time, setting ϵ to 0 gives the optimal policy.

Acknowledgment

The text of Chapter 4, with some modification, is a reprint of the material as it appears in N.J. Butko and J.R. Movellan, "Detecting Contingencies: An Infomax Approach," *Neural Networks* 23(8–9):973–984 (2010) [5]. Both authors shared in writing the paper. My main research role in this project was the implementation of the reinforcement learning developmental model.

Chapter 5

Infomax Control of Eye Movements

5.1 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. 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 chapter, we propose future avenues of research by which autonomous physical agents may use developmental experience

to subjectively characterize the uncertainties they face.

5.2 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 17^{th} 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 hr in saccadic flight, during which useful vision is very poor [18]. 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 chapter, 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

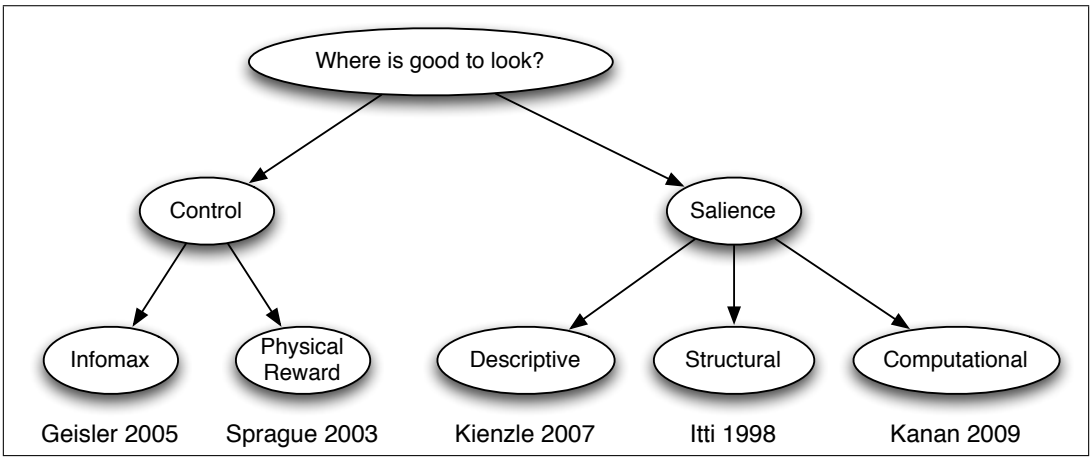


Figure 5.1: Taxonomy of Eye Movement Models with example references, which are not exhaustive. More references and discussion can be found in the text.

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?

5.2.1 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 (Figure 5.1). 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 that pixels will be looked at. These probabilities are learned from examples of images and the locations where people look at in those images [100]. Structural models appeal to neural mechanisms [58] or mental mechanisms [101, 118] to explain why some image regions are more salient than others. Computational models explain observed behaviors as solutions to some problem or objective [2]. For example some recent visual salience algorithms, *e.g.* [42,43,60,96] propose that when people view images they are implicitly trying to maximize the chance of looking directly at a visual search target [4]. 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 [59, 119]. We review the relationship between surprise based information models of eye movements and infomax control models of eye movements shortly (Section 5.2.3).

In this chapter 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 chapter.

(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 doesn't 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 chapter, 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 don't want to just look in the same place always [61]. This task can be something physical, *e.g.* "pick up and throw away garbage" [120] and "track a moving cursor with an unreliable joystick" [121], or it could be purely exploratory, gathering information as quickly as possible, which we call infomax (Figure 5.1) [61]. 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 [122].

(4) Finally, a common distinction made in the literature is "top down" *vs.* "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 [123]. 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 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 [124, 125]. Beliefs and expectations modulate primary visual cortex [126]. Moreover psychological laws that were sup-

posed 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 chapter we abandon the top down *vs.* bottom up terminology.

5.2.2 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 $X_{1:t} = (X_1, \cdots, X_t)$.

5.2.3 The Value of Information

Consider the problem of crossing a one-way street like the one shown in Figure 5.2. 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: $X_t = 0$, indicates that it is unsafe to cross at the current time t, and $X_t = 1$ means that crossing is safe. $H_t = (Y_{1:t-1}, U_{1:t-1})$ represents the history of actions and observations up to time t. $p(X_t = 1|h_t)$ is our belief, based on the history of observations h_t as to whether or not it is safe to cross. We can choose to take one of three actions: $U_t = l$ means that we look left, $U_t = r$ means that we look right (where the cars are coming from) and $U_t = c$ means that we cross. Our beliefs about X_t are shaped by the observations provided by our visual system after taking action u_t. For simplicity, assume the system tells us whether or not a car is present

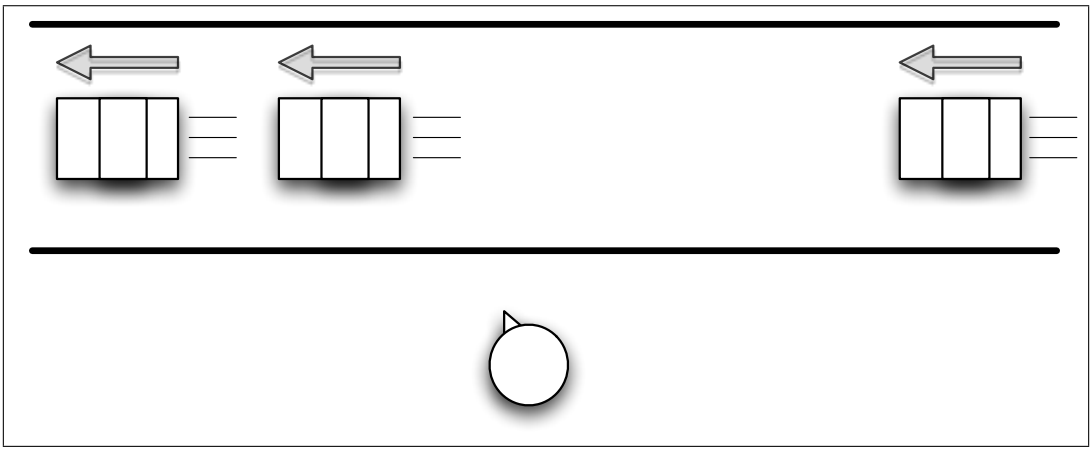


Figure 5.2: We get more information about whether it's safe to cross this one-way street by looking to the right than by looking to the left.

in the field of view: $Y_t = 0$ if no car is visible and $Y_t = 1$ 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 won't 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 (Figure 5.2). 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 y_t and action u_t provide *about* a variable of interest X_t is the reduction

of uncertainty about X_t due to that observation and action:

$$\mathcal{I}(X_t, y_t, u_t|h_t) = \mathcal{H}(X_t|h_t) - \mathcal{H}(X_t|y_t, u_t, h_t) \tag{5.1}$$

$$= \int p(x_t|y_t, u_t, h_t) \log p(x_t|y_t, u_t, h_t) dx_t$$

$$- \int p(x_t|h_t) \log p(x_t|h_t) dx_t \tag{5.2}$$

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

$$\mathcal{I}(X_t, Y_t, u_t|h_t) = \int p(y_t|h_t, u_t)\ \mathcal{I}(X_t, y_t, u_t|h_t) dy_t$$

$$= \int p(x_t|y_t, u_t, h_t) \log p(x_t|y_t, u_t, h_t) dx_t dy_t$$

$$- \int p(x_t|h_t) \log p(x_t|h_t) dx_t \tag{5.3}$$

$$= \mathcal{H}(X_t|h_t) - \mathcal{H}(X_t|Y_t, u_t, h_t) \tag{5.4}$$

where the Shannon entropy $\mathcal{H}$ is a measure of uncertainty. Note that $\mathcal{H}(X_t|h_t)$ is constant with respect to u_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 X_t . In our example $\mathcal{H}(X_t|Y_t, U_t = r, h_t)$ will be smaller than $\mathcal{H}(X_t|Y_t, U_t = l, h_t)$ and thus we choose to look right. Equation (5.3) reveals the details that are needed to be able to quantify information: $p(x_t|y_t, u_t, h_t)$, where $p(x_t|y_t, u_t, h_t) \propto p(y_t|x_t, u_t) \int p(x_t|x_{t-1}, u_{t-1}) p(x_{t-1}|h_t) dx_{t-1}$ by the Markov property and Bayes' rule. Thus, in general, in order to compute the information reward, we need both a model of system dynamics $p(x_t|x_{t-1}, u_{t-1})$, and a model of the observation function $p(y_t|x_t, u_t)$.

Some authors in the visual salience literature [119] 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 y_t is defined as the KL divergence between the prior distribution of a state and the posterior distribution of the state given the observation:

$$\text{Surprise}(X_t, y_t|h_t) \stackrel{\text{def}}{=} \int p(x_t|h_t, y_t) \log \frac{p(x_t|h_t, y_t)}{p(x_t|h_t)} dx_t \tag{5.5}$$

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

$$\text{Surprise}(X_t, Y_t|h_t, u_t) = \int p(y_t|h_t, u_t)\ \text{Surprise}(X_t, y_t|h_t)dy_t \tag{5.6}$$

$$= \int p(x_t|h_t)p(y_t|h_t, u_t)\ \log \frac{p(x_t|h_t, y_t, u_t)}{p(x_t|h_t)} dx_t dy_t \tag{5.7}$$

$$= \mathcal{I}(X_t, Y_t, u_t|h_t) \tag{5.8}$$

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 [119] 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 (5.5). Control models consider (sometimes implicitly) the consequence of future actions and observations, as in Equation (5.6), 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.

5.2.4 Infomax in other domains

Maximization of expected information gain was proposed by Lindley [14] as a sensible criterion for designing experiments. Stone [47] and Fedorov [48] applied this idea to the efficient estimation of parameters in linear regression and ANOVA models. Bernardo [49] 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 non-linear 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 [55].

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 [127]. Nelson *et al.* [111] showed that information maximization could be used to model how humans ask questions in active concept learning tasks.

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 [56].

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 [50], and in active-learning scenarios to quickly learn how to accurately diagnose health conditions from medical images [51].

In the preceding chapter, we showed that 10-month-old infants schedule vocalizations so as to optimally detect contingent social interaction. We also showed that information gain could be used as a reward for reinforcement learning algorithms and explain the developmental trajectories observed in infants. However, solving the problem exactly was computationally expensive, and only fairly limited controllers could be learned in a developmentally plausible time frame.

In this chapter, we attempt to extend the infomax approach that we adopted for a baby robot with a binary sensor and binary actuator to a sensory motor system that is on the order of complexity of human vision. *I.e.*, we attempt to answer the questions, "How can photons be translated into information about something?" and "How can eye movements be scheduled to gather information as quickly as possible?"

5.3 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 [128]. This game is analogous to a situation in which an observer moves her eyes in order to search a 2D image plane of bounded size for a target that is not moving.

5.3.1 POMDP Problem Formulation

A POMDP is defined by the following elements [26] (with their correspondences in the Where's Waldo? control model):

- X_t is random variable that represents the state of the world at time t. In this chapter, 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, $X_t = i$ means that Waldo is at location i, at time t.

- U_t is random variable that represents the action taken by the agent at time t. In the Waldo example, $U_t = k$ means that the agent fixated location k at time t.

- Y_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.

- $p(x_{t+1}|x_{1:t}, u_{1:t}, y_{1:t}) = p(x_{t+1}|x_t, u_t)$: 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 $p(x_{t+1}|x_t, u_t) = 1$ if $x_{t+1} = x_t$, 0 otherwise.

- $p(y_t|x_{1:t}, u_{1:t}) = p(y_t|x_t, u_t)$: Markovian observation model – How objects appear at different points in the fovea or periphery. Red & 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.

5.3.2 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

$$B_t^i \stackrel{\text{def}}{=} p(X_t = i|h_t, u_t, y_t) \tag{5.9}$$

It is easy to show that the belief state vector at time t is a function of u_t, y_t and B_{t-1}. Specifically, given a history $h_t = \{u_{1:t-1}, y_{1:t-1}\}$ of actions and observations, and a new action u_t and observation y_t, then

$$\begin{aligned} B_t^i &= p(X_t = i|h_t, u_t, y_t) \propto p(X_t = i, y_t|h_t, u_t) && (5.10)\\ &= p(y_t|X_t = i, u_t)p(X_t = i|h_t, u_t) \\ &= p(y_t|X_t = i, u_t) \sum_{j=1}^{N} p(X_t = i|X_{t-1} = j, u_t)\ B_{t-1}^j && (5.11) \end{aligned}$$

Waldo never moves, so this becomes

$$B_t^i = \frac{p(y_t|X_t = i, u_t)B_{t-1}^i}{\sum_{k=1}^{N} p(y_t|X_t = k, u_t)B_{t-1}^k} \tag{5.12}$$

Thus the previous belief state B_{t-1} encodes all the relevant information from h_t, the history of the 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, [61] argues that subjects forget what they've seen because it's simply too costly to remember many observations. But Equation (5.11) tells us that there is practically no cost to memory: to remember everything *that's 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's almost completely irrelevant where he was or wasn't 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 the agent still remembers all that's relevant about everything it has 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 [61] 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, which is an environment where objects move.

5.3.3 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 be a decreasing function of the entropy of the state belief

$$\begin{aligned} R_t &= -w_t \mathcal{H}(X_t|u_t, y_t, h_t) \\ &= w_t \sum_{i=1}^{N} B_t^i \log B_t^i \end{aligned} \tag{5.13}$$

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, $U_{t+1} = \pi_t(B_t)$. 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

$$V_t^{\pi}(b_t) = \sum_{s=t}^{T} E[R_t|b_t, \pi] \tag{5.14}$$

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

$$\pi_t^*(b_t) = \underset{\pi}{\operatorname{argmax}}\, V_t^{\pi}(b_t) \tag{5.15}$$

At first sight, the infomax reward function appears peculiar in that it is based on our own beliefs, *e.g.* it doesn't matter to the agent where Waldo is; the agent only cares about being sure of where he is." This is in fact a typical of POMDP problems, not just infomax problems. Kaelbling *et al.* observe that 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 it 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 [26].

5.3.4 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 $p(y_t|x_t, u_t)$ specifying how the world may look like, and the system's dynamics model in the form of a family of distributions $p(x_t|x_{t-1}, u_{t-1})$ specifying how the world may change in the future. In [120], these probability distributions were constructed by creating simulated worlds. Since the researchers constructed the worlds, they knew precisely the uncertainties in those worlds. In the preceding chapter, we measured the statistics of human-robot social interaction to fit the model parameters. In [61], 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.

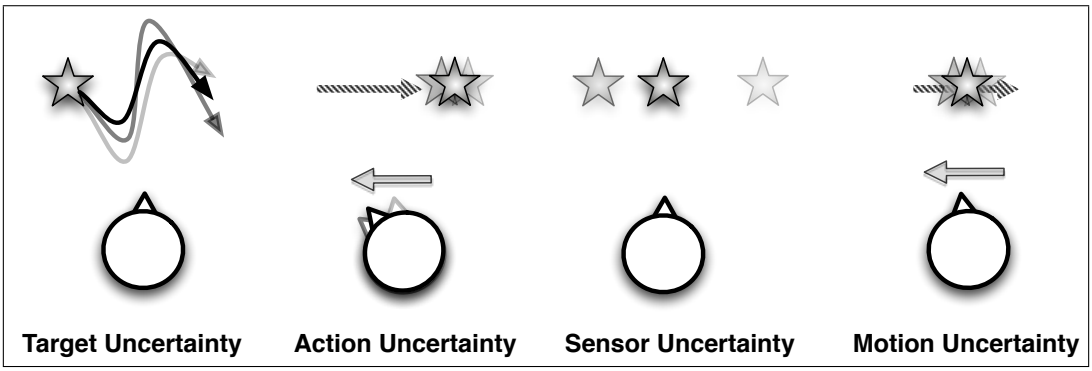


Figure 5.3: Different factors introduce uncertainty in visual search targets localization. A few examples of these many factors are: how targets will move, the reliability of our own muscles, loss of reliability at visual eccentricity, and motion blur or distortion.

Without specifying these probability models and their associated uncertainty, we cannot compute the belief update in Equation (5.11), or the information reward in Equation (5.13). The following are examples of sources of uncertainty that may be considered in modeling eye movement (Figure 5.3):

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

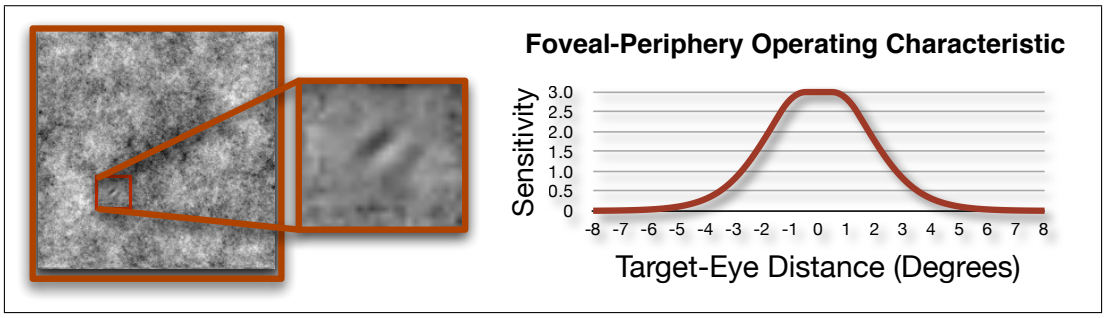


Figure 5.4: **Left:** A wavelet is "hidden" in a pink noise background. **Right:** Najemnik & Geisler measured subjects' ability to detect these targets as a function of how far away they were looking.

5.4 A Control Model of Visual Search

In this section, we present a psychophysical model of visual search developed by Najemnik & Geisler (N&G) [61], reformulate it from the point of view of stochastic optimal control. In later sections, we 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 non-moving targets.

In N&G's model, the task is to find a target stimulus (a Gabor wavelet) in a correlated Gaussian noise background (Figure 5.4). 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, by using psychophysical experiments 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 Figure 5.4.

In terms of the sources of uncertainty described in Figure 5.3, N&G's model can be summarized as follows:

- **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.
- **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 shown in Figure 5.5. At each timepoint t, a noisy observation $y_t \in \mathbb{R}^N$ is sampled from a visual array. The visual array contains N potential target locations, and the element y_t^j of the observation vector gives some information about the presence or absence of the visual target at visual array location j. This noisy observation is illustrated in the "Signal+Noise" column of Figure 5.5. In locations without a target, the observation is drawn from a baseline Gaussian distribution, which has zero-mean and standard deviation one[1]:

$$y_t^j \sim \mathcal{N}(0,1), \text{ when } x_t \neq j \tag{5.16}$$

These zero-mean locations are shown as darker regions in the "Signal" column of Figure 5.5. 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" column) and converges towards zero as the eccentricity increases. *I.e.*, let i be the location of the target, k be the location of fixation, then:

$$y_t^i \sim \mathcal{N}(d_{i,k}, 1), \text{ when } x_t = i \tag{5.17}$$

The mean signal $d_{i,k}$ is the discriminability of a target at location i given that the fovea is centered at location k. We call this the foveal-peripheral operating characteristic (FPOC) of location i given that the retina is centered at k. In humans the FPOC $d_{i,k}$ decreases with increased distance of location i from the

[1] Throughout this chapter, we use the notation $y \sim \mathcal{N}(\mu, \sigma^2)$ to denote that the random variable Y has its value y drawn from the Gaussian probability distribution with mean μ and variance σ^2. We use the notation $\mathcal{N}_y(\mu, \sigma^2)$ to denote the value of the Gaussian probability density function for that distribution, evaluated at y.

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" column of Figure 5.5.

Under the model the individual observations Y_t^j are conditionally independent given the external scene,[2] and so the likelihood of an entire vector of observations $y_t = (y_t^1, \cdots, y_t^N)$ given that the target is at location i and the eye is focusing on location k is as follows:

$$\begin{aligned} p(y_t|X_t = i, U_t = k) &= \prod_{j=1}^{N} p(y_t^j|X_t = i, U_t = k) \\ &= \mathcal{N}_{y_t^i}(d_{i,k}, 1) \prod_{j \neq i} \mathcal{N}_{y_t^j}(0, 1) \\ &= \frac{\mathcal{N}_{y_t^i}(d_{i,k}, 1)}{\mathcal{N}_{y_t^i}(0, 1)} \prod_{j=1}^{N} \mathcal{N}_{y_t^j}(0, 1) \\ &= \frac{\exp((y_t^i - d_{i,k})^2/2)}{\exp((y_t^i)^2/2)} \, Z \\ &= \exp(\alpha_{i,k} d_{i,k}) \, K; \qquad \alpha_{i,k} \stackrel{\text{def}}{=} (y_t^i - d_{i,k}/2) \end{aligned} \tag{5.18}$$

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" column of Figure 5.5). Combining this with Equation 5.12 yields the proportional belief update ("Belief" column of Figure 5.5)

$$b_{t+1}^i \quad \propto \quad \exp(\alpha_{i,k} d_{i,k}) b_t^i \tag{5.19}$$

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

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

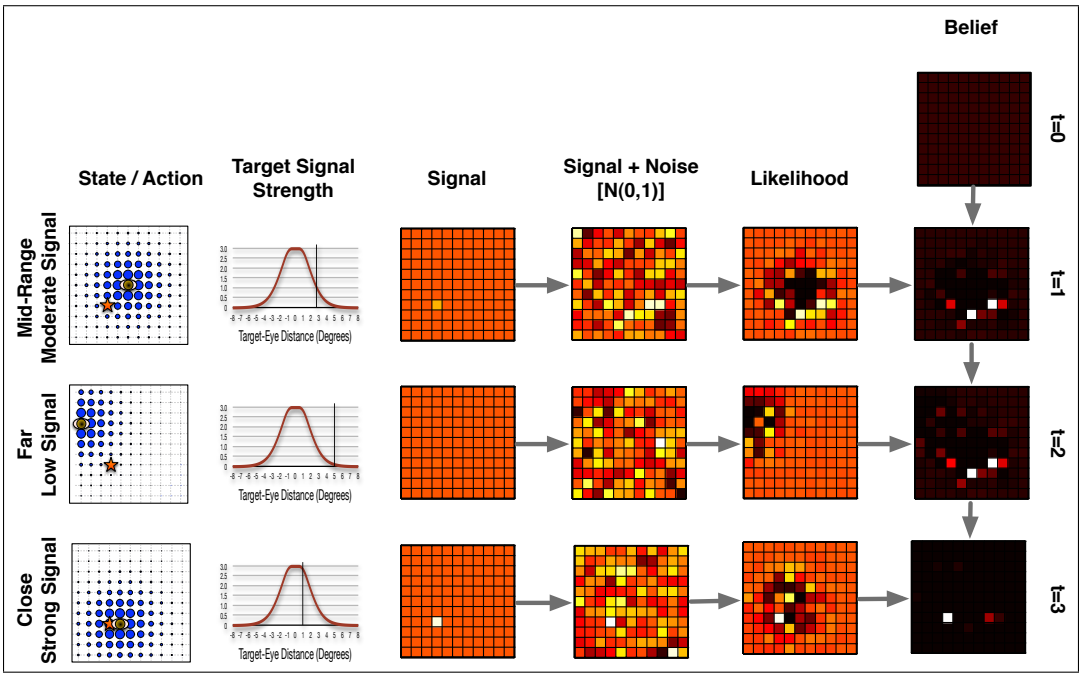


Figure 5.5: The I-POMDP model of Eye Movement: A target is located at a visual location previously unknown to the subject. After making several fixations, the subject comes to know with high confidence the location of the visual target. See text for further description.

grows linearly. The belief about location of the search target could be updated with simple neural circuitry and strictly local update rules.

5.5 Learning Where to Look

N&G [61] 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 the 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 $\pi(B_t) \rightarrow U_{t+1}$ that approximates the optimal policy defined in Equation (5.15). Algorithms for learning exactly optimal policies in POMDPs exist, but are only feasible with few states, actions, and observations [26]. Point-based approximation methods can learn approximately optimal policies for POMDPs with many states and actions, but require a small observation space [31]. The I-POMDP model has an $\mathbb{R}^N$ 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 non-belief-linear reward functions like Equation (5.13).

5.5.1 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 Equation (5.14) with respect to θ [32].

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 [32]:

1. Choose an initial value for θ.

2. Set $t = t_0$; Get an initial belief state b_t. Set $z = \vec{0}$ ($z \in \mathbb{R}^N$).

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 $z \leftarrow z + \beta \frac{\nabla_\theta p(b_{t+1}|b_t,\theta)}{p(b_{t+1}|b_t,\theta)}$
- Set $\theta \leftarrow \theta + \gamma_t r_{t+1} z$
- 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, $\frac{\nabla_\theta p(b_{t+1}|b_t,\theta)}{p(b_{t+1}|b_t,\theta)}$. In this chapter's appendix, Sections 5.8.1 & 5.8.2, we show how this can be done for logistic policies of the type described below.

5.5.2 Policy Gradient with Logistic Policies

We parameterize the policy as a logistic function. Let the parameter $\theta \in \mathbb{R}^{N \times M}$ 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:

$$p(U_{t+1} = k|b_t, \theta) \quad = \quad \frac{\exp(\theta^k \cdot \phi(b_t))}{\sum_{i=1}^{n} \exp(\theta^i \cdot \phi(b_t))} \tag{5.20}$$

where $\phi(\cdot)$ is a feature function that takes the belief vector as input and outputs another vector. Logistic policies can 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 $\phi(b_t) \stackrel{\text{def}}{=} b_t$, *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, while allowing an intuitive examination of the learned policy. For example, a policy of greedily fixating at the most probable target location would be represented as $\theta = \omega I$ for $\omega \to \infty$. A policy of searching randomly would be represented as $\theta = 0$.

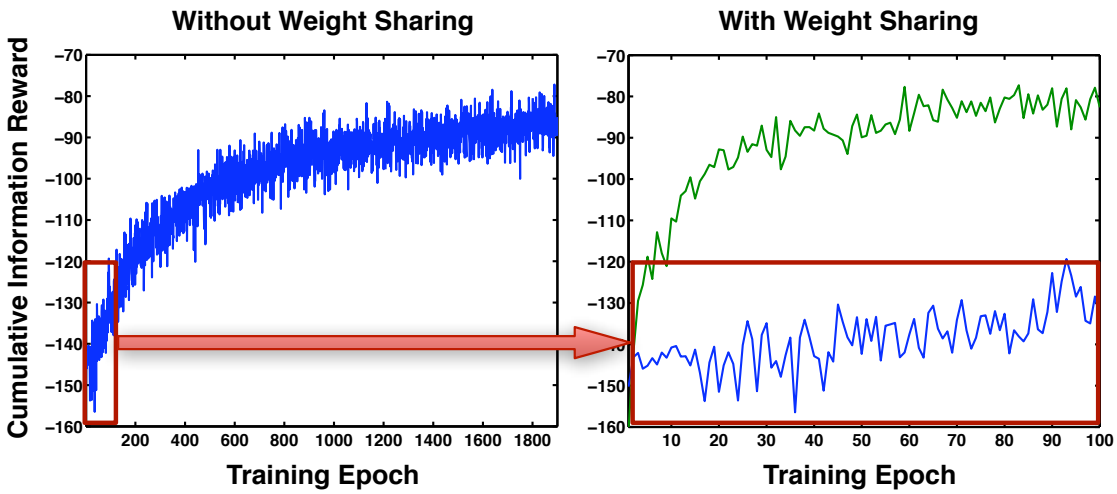


Figure 5.6: **Left:** Policy gradient enables learning even when there are 14,641 parameters. **Right:** Learning is 20 times faster when we use weight sharing to exploit invariances, reducing the number of parameters to 61. The original learning curve is duplicated in blue in "With Weight Sharing" to highlight this timescale difference.

5.5.3 Convolutional Policies

The policy model in Equation (5.20) can have many parameters. For an 11×11 visual array, there are 14,641 parameters. Figure 5.6 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.[3] This approach results in a convolutional policy which is defined by a rotationally symmetric, two-dimensional kernel. Under convolutional policies of this type, the belief map is treated as an image, which is convolved with a kernel defined by the elements of θ. The output is then normalized to give a probability distribution over next fixations U_{t+1}.

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

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

converges much faster (Figure 5.6). For the remainder of this chapter, 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.

5.5.4 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

- **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 the salience literature [123].
- **Visual System Properties:** In addition to using an FPOC from psychophysical data [61], 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)". *I.e.*, 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; In a 7×7 visual array, if the location of the target had higher belief than all 48 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.

5.5.5 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. **%-Correct Greedy:** Choose the action that yields the highest expected %-correct after the observation, *i.e.* that maximizes $R_{t+1} = \max_i E[B^i_{t+1}]$ (proposed in [61]). 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 [4, 58].

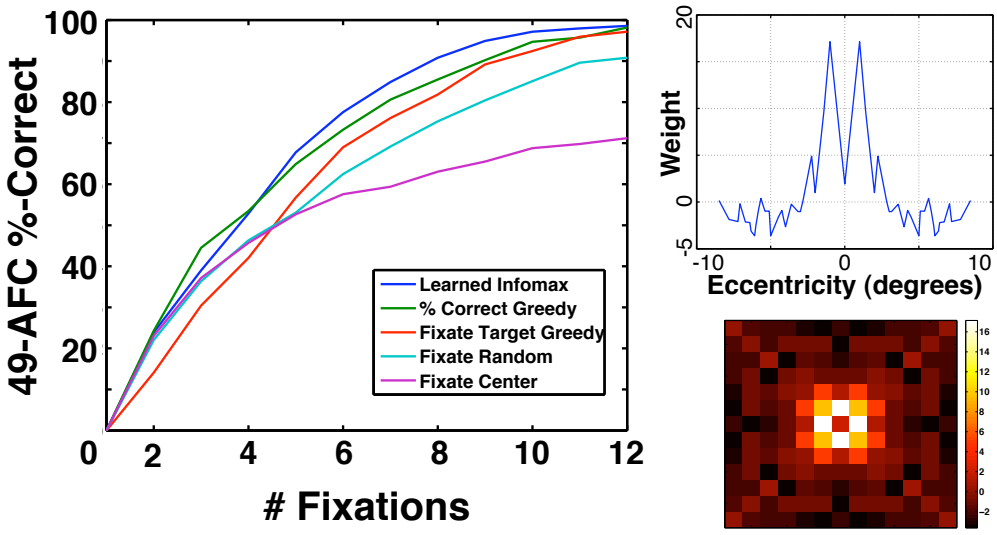
We also evaluated the performance of two policies that we expected to perform poorly:

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 Figure 5.5 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 11x11 or 7x7, depending on the experiment. For the latter, there were 4,900 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 5.1). The performance of all policies is shown in Figure 5.7a.



(a) Comparison of Policies (b) Learned Policy

Figure 5.7: **(a)** The Learned Policy performs better than 4 alternative policies described in Section 5.5.5. Policy "%-Correct Greedy", proposed in Najemnik & Geisler, outperforms the learned policy in only the first 4 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.

The policy that achieves this high performance is visualized in Figure 5.7b. 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

Table 5.1: # Fixations to reach 90% Correct (49-AFC)

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

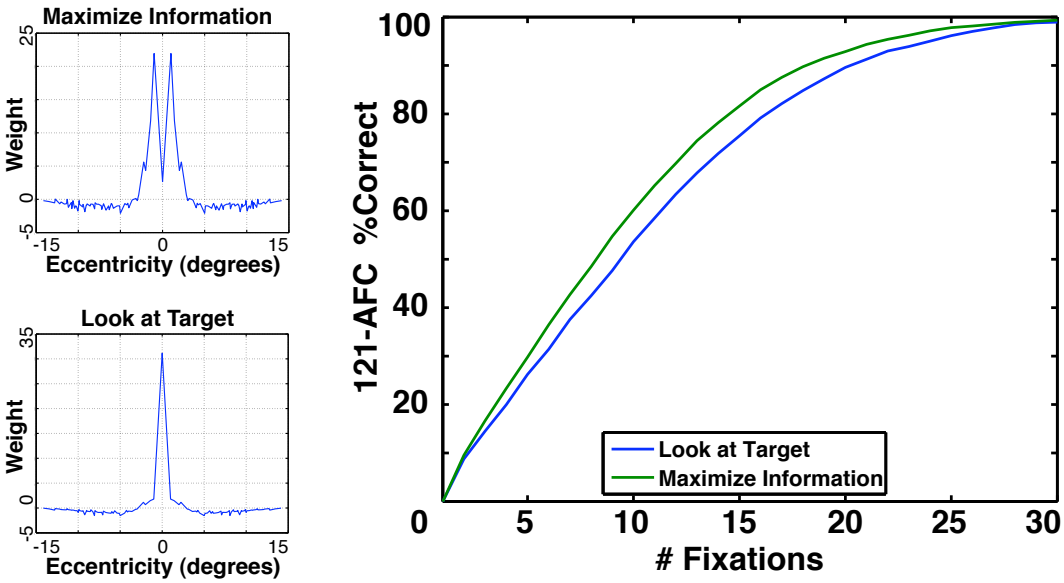
plausible hypotheses turn out to be wrong.

5.5.6 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 [26], 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 $U_t = k$.

We trained Infomax and Fixate Target controllers on an 11×11 visual array I-POMDP. The learned control laws are visualized in Figure 5.8a. 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. Figure 5.8b indicates that this is a reasonable but suboptimal strategy. Controllers optimized to Fixate Target require 20 fixations to reach 90% accuracy



(a) Learned Policy

(b) Comparison of Objective Functions

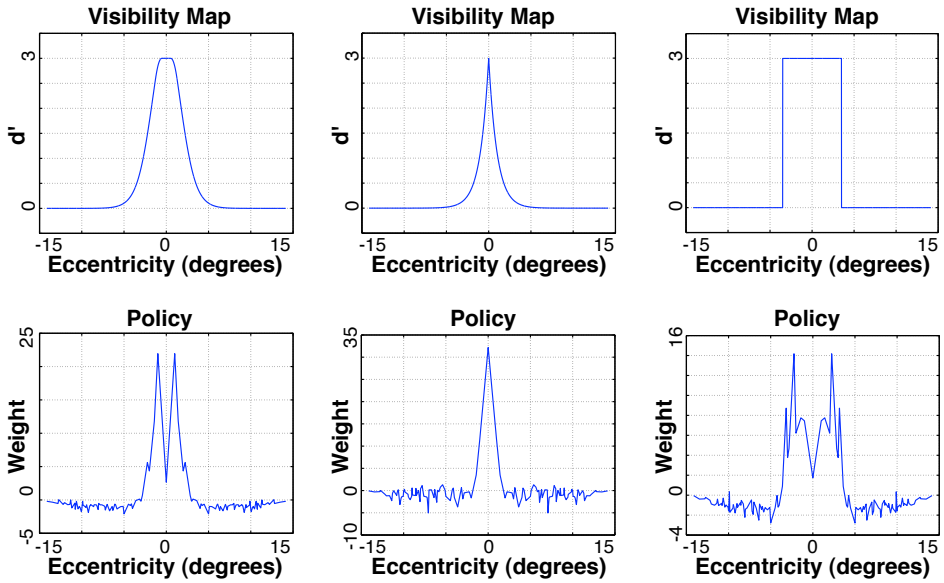
Figure 5.8: Performance loss from directly fixating the target; the visual array is 11×11. **(a)** Learned "receptive fields." *Top:* The Infomax policy closely resembles the policy in Figure 5.7b which was trained on a smaller visual array. *Bottom:* A different policy is learned when the goal is to look directly at the target. **(b)** Maximizing information performs noticeably better than trying to look directly at the target.

on a 121-AFC tasks, while those optimizing information-gain require 18 fixations.

This quantifies the expected performance boost achievable over previous salience approaches in robots [4], 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.

5.5.7 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 foveal-peripheral



(a) Psychophysical (b) Exponential (c) Camera

Figure 5.9: 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 Najemnik & Geisler, which was used in this chapter'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.

operating characteristic (FPOC) shown in Figure 5.5. 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 because 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 (Figure 5.9b). 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 (Figure 5.9c).

The resulting control laws are strikingly different from the original (Figure 5.9a), 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.

5.5.8 Dependence on Search Target Dynamics

In Section 5.3.2, we argued that subject data that were interpreted by N&G to as forgetting were consistent with an alternate hypothesis: that the human perceptual system is tuned to search for and track search targets that are able to move.

We wanted to test the effect that such a tuning would have on the optimal search strategy. To study this, we added a simple dynamics model $p(x_t \,|\, x_{t-1}, u_t) = p(x_t \,|\, x_{t-1})$. Under this model, the target moved according to Brownian motion in a fashion that was independent of the eye-movement u_t.

In the presence of even small amounts of target motion, the optimal search strategy changed dramatically: rather than a double-peaked convolution kernel (as in Figure 5.8a, Top), the learned policy changed to be single-peaked, nearly identical to the convolution kernel that was optimal for directly fixating the target (as in 5.8a Bottom). This speaks to the importance of studying the natural conditions that humans are situated in when making claims about the sub-optimality of their performance. Human subjects may be slightly less than optimal (and appear to forget) in an artificial task that runs counter to their every day experiences.

5.6 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 5.5.7 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 is provided online as part of Nick's Machine Perception Toolbox [129].

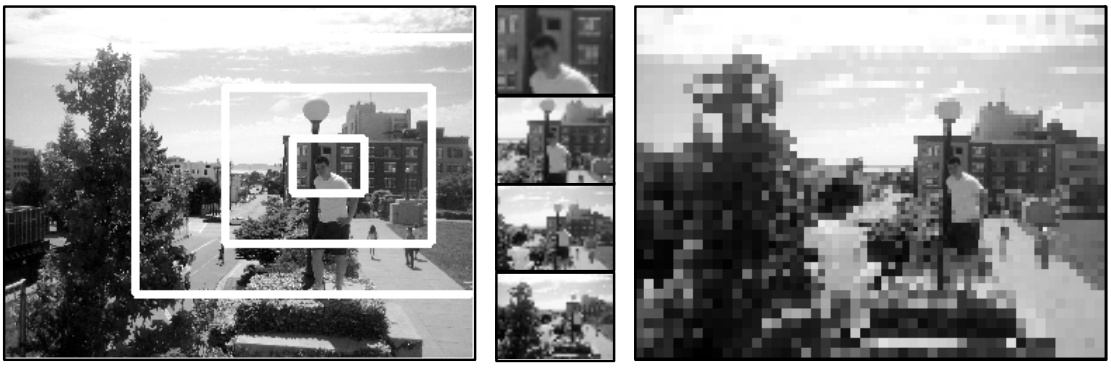


Figure 5.10: A digital fovea: Several concentric image patches (IPs) (*Top*) are arranged around a point of fixation. The image portions contained within each rectangle are reduced to a common size (*Middle*). In a reconstruction from the downsampled images, detail is preserved around the fixation point, but decreases with eccentricity (*Bottom*).

5.6.1 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 [130] (see Figure 5.10). Each image patch (IP) is then shrunk to a common reference size that is much smaller than the original image, typically $1/100^{th}$ 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 information about low spatial frequencies. Small IPs will maintain resolution and high spatial frequency information, but only around a small region of the image.

Figure 5.10 shows an example of the digital fovea at work. In this case we used 4 IPs per fixation, operating at 4 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 4 IPs are shrunk to the same small size, an object detector with linear complexity will search all 4 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.[4]

5.6.2 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 Figure 5.11. 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.

An observation model $p(y_t|x_t)$ 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 y_t such that it gives relevant and meaningful information about the state x_t, *i.e.* the location of the search target.

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

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

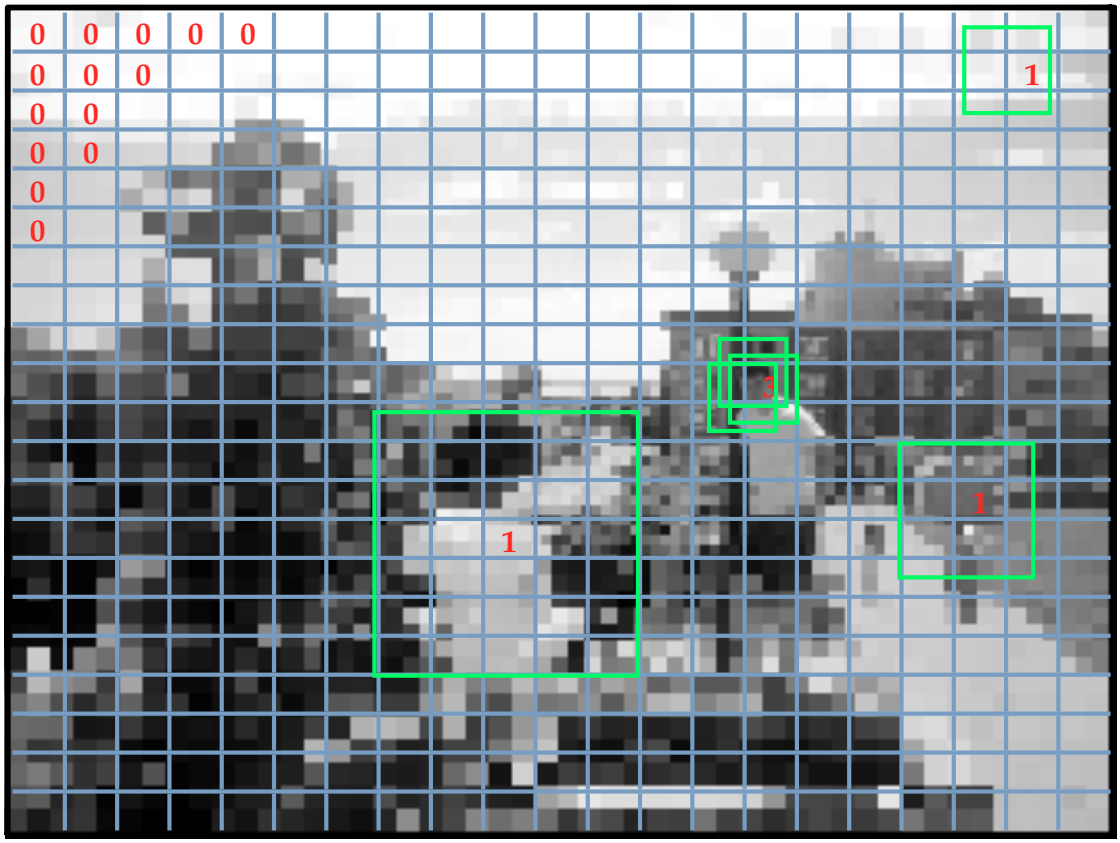


Figure 5.11: Generative model for the observation vector Y_t in MI-POMDP: An object detector for the search target returns several candidate boxes. The image is discretized into grid cells. The number of boxes centered in each cell j gives an element y_t^j of the observation vector (all empty grid cells have count 0)

target. These detectors often fire in clusters around the object (hits), but also have false alarms, misses, and correct rejections (Figure 5.11).

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 $Y_t \in \{0, 1, ..., c_{max}\}^N$.

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 x-distance $\in [0, 1, ..., M-1]$, y-distance $\in [0, 1, ..., L-1]$, and object presence / absence. We refer to the I-POMDP with this modified multinomial observation model as the multinomial I-POMDP, or MI-POMDP.

In images, the target we are searching for does not move, and the POMDP belief update equation in Equation (5.12) can be used. In active cameras or video streams, the target might move between each fixation. In this case, the dynamics are modeled by $p(X_t = i|X_{t-1} = g)$, and the belief update becomes

$$b_t^i \propto \frac{p(y_t^i|X_t = i, U_t = k)}{p(y_t^i|X_t \neq i, U_t = k)} \sum_{g=1}^{N} p(X_t = i|X_{t-1} = g) b_{t-1}^g \tag{5.21}$$

5.6.3 Fitting the Multinomial Observation Model

In order to estimate the information properties of the digital eye, we had the eye scan each grid cell 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 3,500 images in which faces were present in equal amounts across all scales. Specifically, $\frac{1}{5}$th were $< 10\%$ of the image major axis, and $\frac{1}{5}$th 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 [131].

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 $c_{max} = 9$.

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 $|\text{dist}_x(j,k)|$ and $|\text{dist}_y(j,k)|$, the absolute x- and y-distance from that location to the point of fixation, and incremented the table element $T[0, |\text{dist}_x(j,k)|, |\text{dist}_y(j,k)|, c]$. For the one location i with a face, we incremented the table element $T[1, |\text{dist}_x(i,k)|, |\text{dist}_y(i,k)|, c]$.

After this procedure, the estimates

$$P(Y_t^j = c | X_t \neq j, U_t = k) = \\ = \frac{T[0, |\text{dist}_x(j,k)|, |\text{dist}_y(j,k)|, c]}{\sum_{c'=0}^{c_{max}} T[0, |\text{dist}_x(j,k)|, |\text{dist}_y(j,k)|, c']} \tag{5.22}$$

$$P(Y_t^i = c | X_t = i, U_t = k) = \\ = \frac{T[1, |\text{dist}_x(i,k)|, |\text{dist}_y(i,k)|, c]}{\sum_{c'=0}^{c_{max}} T[0, |\text{dist}_x(i,k)|, |\text{dist}_y(i,k)|, c']} \tag{5.23}$$

correspond to the Bayesian MAP parameter estimates of the multinomial parameters, starting with a uniform Dirichlet conjugate prior [132].

Figure 5.12 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 0 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.

5.6.4 Comparison to other multiresolution approaches.

The search strategies proposed here relate to recent work on optimal image search, like efficient subwindow search (ESS) [133]. Our approach is data driven and detector independent, where the ESS approach is more analytic. We chose

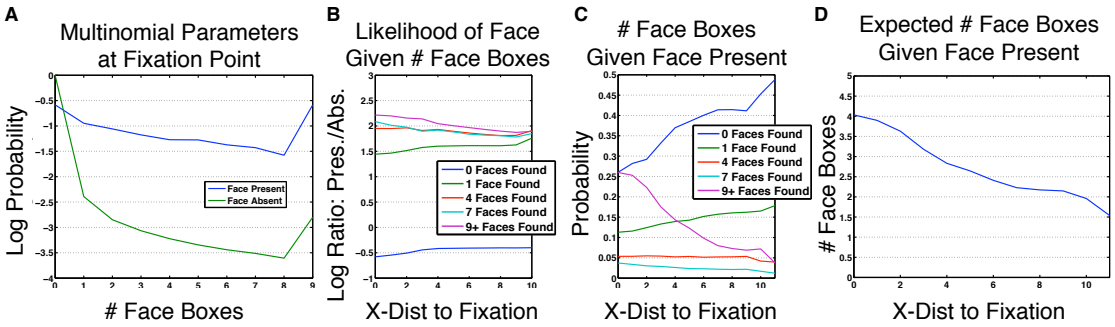


Figure 5.12: Parameters of the multinomial observation model inferred from data: **A:** Probability of counting 0, 1, ... faces at the point of fixation if the face is there, and if it's not there. (In A&C, boundary effects can be seen where all observations of size 9 and greater are binned together.) **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.

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.

5.6.5 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 [63, 134]. For this study, we chose to tile all images with a 21×21 grid, meaning the face could be localized to any of 441 locations.[5] 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 [129].

In the previous section, we fit the 8,820 parameters of the Multinomial detector output model to our full dataset of images. In this and following sections, all results were gathered using 7-fold cross-validation. The images were randomly assigned to 7 groups of 500 images. In each Fold, 6 groups were used to fit the multinomial parameters, and 1 group was used to test performance. All performance results were averaged by repeating this procedure across all 7 folds. All timing experiments were done on quad-core Intel Xeon processors at 2.8GHz. Absolute (wall clock) time was used, with a precision of 1μs. 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.

5.6.6 Default Performance

The OpenCV 1.0 Viola-Jones face detector 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.

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

To plan fixations in a way that gathered information close to optimally, we used a convolutional logistic policy, as above in Equation (5.20). 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. Figure 5.13 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.

The runtime of both algorithms increases as a function of image size is shown in Figure 5.14. 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 2.18x 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 below. Both methods on average placed the face between one and two grid-cells off the true face location.

5.6.7 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

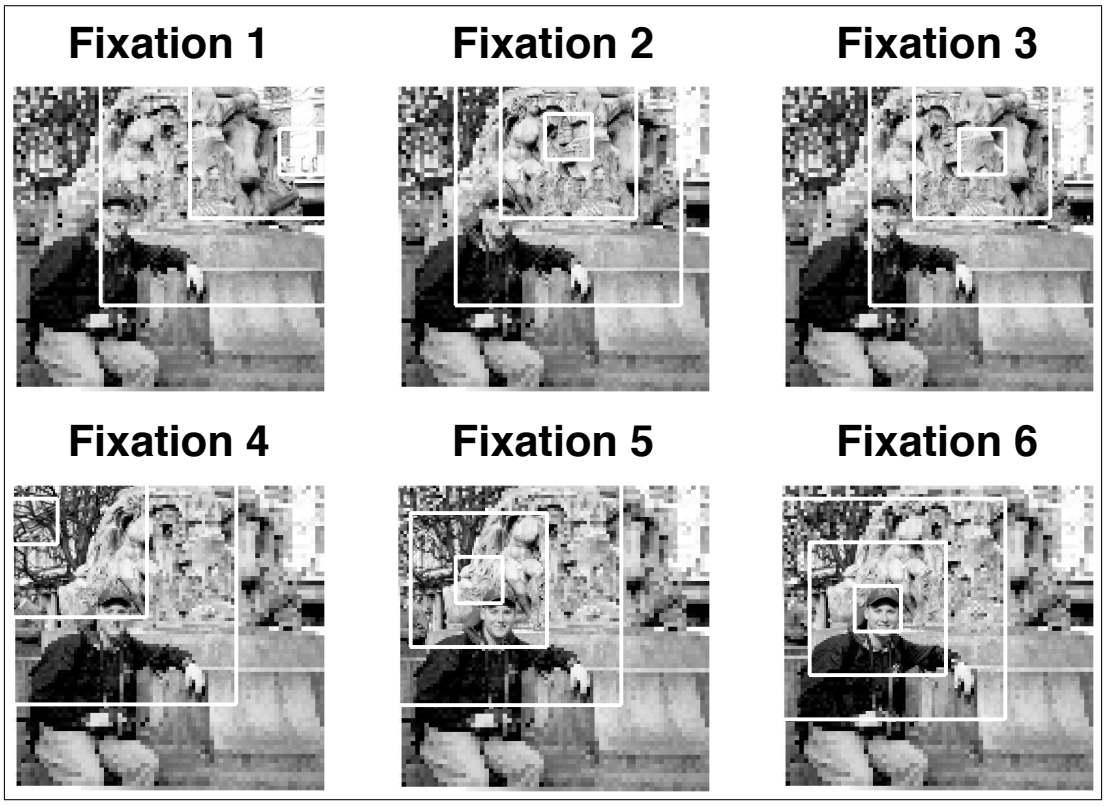


Figure 5.13: Successive fixation choices by the MI-POMDP policy. The face is found in six fixations. The final estimation of the face location is one grid-cell diagonal from the labeled location, giving a Euclidean distance error of 1.4 grid-cells.

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 [134]. 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.

Figure 5.15 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

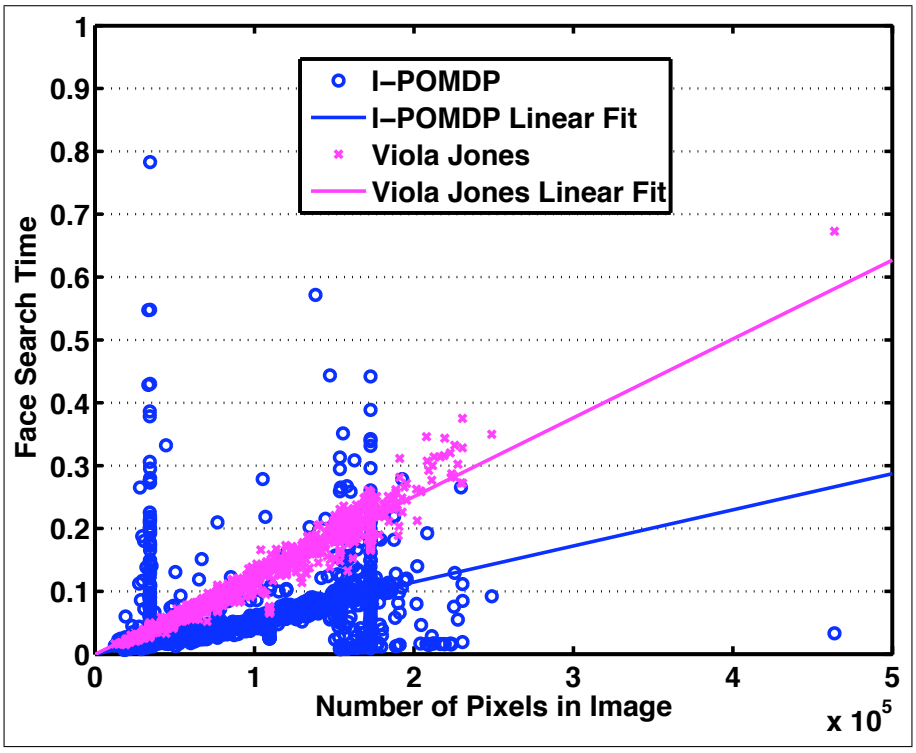


Figure 5.14: Time needed to search for faces, as a function of image size. A mode of the dataset image size distribution was 180×190 (2300/3500 images), explaining apparent spike at 34,000 pixels. Similar modes explain the other spikes.

tradeoff.

5.6.8 Search Target Temporal Dynamics

Target dynamics can be considered when searching 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

Table 5.2: 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**

explore this approach to object detection in high definition video. By assuming the search target moves according to simple Brownian motion dynamics as in 5.5.8, we can reduce the computation time per frame to about $\frac{1}{20}^{th}$ of that required for full frame search.

5.6.9 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 complementary to salience based search strategies, and in fact can be integrated with such approaches, like those taken in [135]. 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 the ones in Chapter 3, 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?

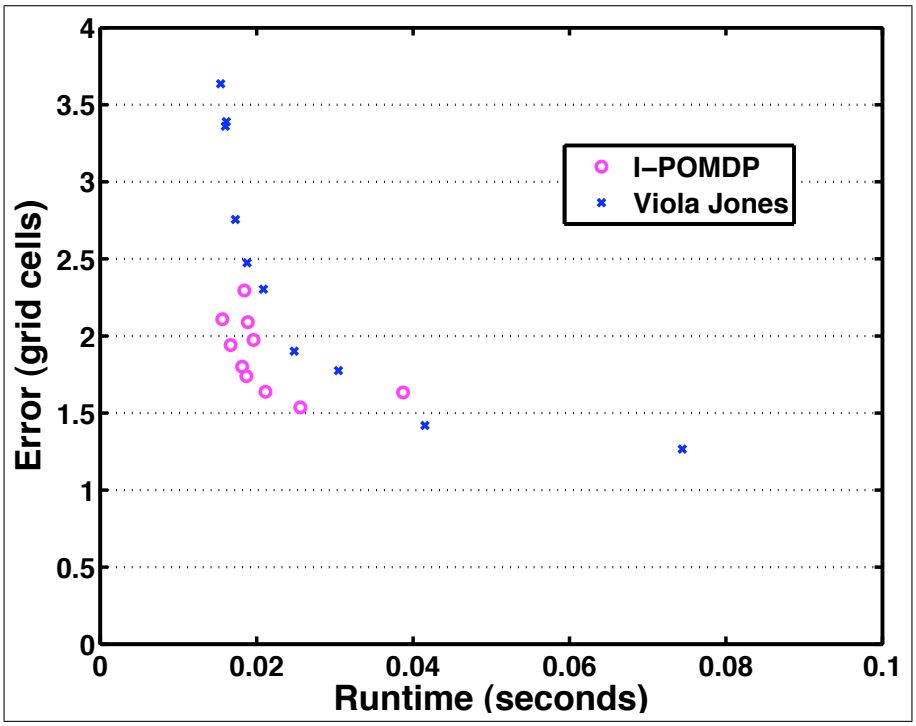


Figure 5.15: By changing the Viola Jones scaling factor, both Viola Jones and I-POMDP become faster and less accurate. MI-POMDP is usually closer to the origin on a time-error curve, showing that it gives a better speed-accuracy tradeoff than just applying Viola Jones.

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.

5.7 On the Role of Learning and Development

The main focus of this chapter 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 salience 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 (Figure 5.16), there are important sources of uncertainty that cannot be

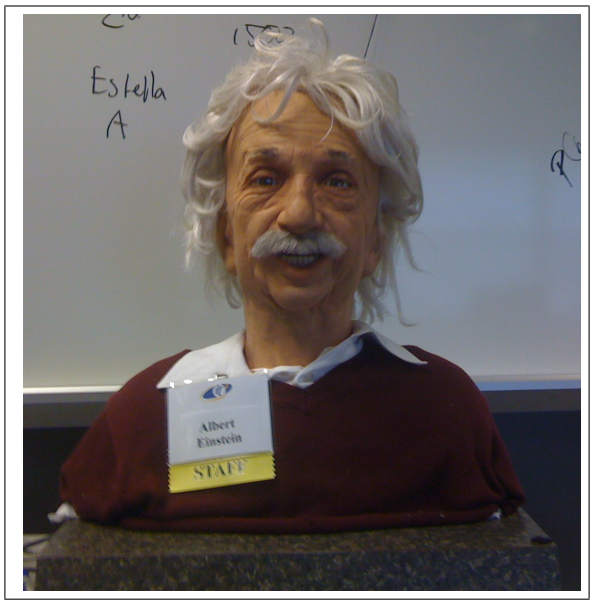


Figure 5.16: The Einstein robot.

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.

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 [61, 136]. 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 chapter 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 don't 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 [137]:

> 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 don't start to move until about 200 ms after a motor command is requested, that there is rapid movement from about 200-300ms, and small jitter and position refinements until about 500ms.

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 500ms 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 vs. 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.

5.8 Appendix

In this appendix, we adopt the shorthand notation $y \in Y$ as an enumeration of the outcome space Ω of the random variable Y. *I.e.*, for a random variable $Y : \Omega \rightarrow \mathbb{R}$, we use $y \in Y$ to mean that y takes on all the values in the set Ω in turn.

5.8.1 General Policy Gradients

Ultimately, in all policy gradient methods, the quantity we care about is

$$\frac{\nabla_\theta p(b_t|b_{t-1},\theta)}{p(b_t|b_{t-1},\theta)} \tag{5.24}$$

where

$$p(b_t|b_{t-1},\theta) = \sum_{y_t \in Y} \sum_{u_t \in U} p(b_t|y_t,u_t,b_{t-1}) p(y_t|u_t,b_{t-1}) p(u_t|b_{t-1},\theta) \tag{5.25}$$

Note that the belief update is deterministic given y_t and u_t, and so $p(b_t|y_t,u_t,b_{t-1})$ always 1 or 0. Let $\hat{Y}_{ut}$ be the set of observations that cause a state transition to b_t from state b_{t-1} under action u_t, *i.e.* $\hat{Y}_{ut} \stackrel{\text{def}}{=} \{y_t \in Y_t \text{ s.t. } p(b_t|y_t,u_t,b_{t-1}) = 1\}$. Then

$$p(b_t|b_{t-1},\theta) = \sum_{u_t \in U} \sum_{y_t \in \hat{Y}_{ut}} p(y_t|u_t,b_{t-1}) p(u_t|b_{t-1},\theta) \tag{5.26}$$

$$\nabla_\theta p(b_t|b_{t-1},\theta) = \sum_{u_t \in U} \sum_{y_t \in \hat{Y}_{ut}} p(y_t|u_t,b_t) \nabla_\theta p(u_t|b_{t-1},\theta) \tag{5.27}$$

Thus the policy gradient update rule can be written as

$$\frac{\nabla_\theta p(b_t|b_{t-1},\theta)}{p(b_t|b_{t-1},\theta)} = \frac{\sum_{u_t \in U} \sum_{y_t \in \hat{Y}_{ut}} \nabla_\theta p(u_t|b_{t-1},\theta) p(y_t|u_t,b_{t-1})}{\sum_{u_t \in U} \sum_{y_t \in \hat{Y}_{ut}} p(u_t|b_{t-1},\theta) p(y_t|u_t,b_{t-1})} \tag{5.28}$$

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, $\nabla_\theta p(u_t|b_{t-1},\theta)$.

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 $\hat{Y}_{ut}$ is empty for all y_t other than the observation we actually just made, obviating the inner sum over the possibly large number of observations. Then we have

$$\frac{\nabla_\theta p(b_t|b_{t-1},\theta)}{p(b_t|b_{t-1},\theta)} = \frac{\sum_{u_t \in U} \nabla_\theta p(u_t|b_{t-1},\theta) p(y_t|u_t,b_{t-1}) 1(y_t \in \hat{Y}_{ut})}{\sum_{u_t \in U} p(u_t|b_{t-1},\theta) p(y_t|u_t,b_{t-1}) 1(y_t \in \hat{Y}_{ut})} \tag{5.29}$$

where $1(y \in \hat{Y}_{ut})$ simply indicates whether it would be possible for the same belief update to occur if observation y_t were made under a different action than the one we just saw.

Even without the above assumption, the numerator and denominator of the policy gradient Equation (5.29) will be *on average* correct, because each observation is made with the correct probability. However, computing the full sum gives a better, less variable estimate of the true gradient.

5.8.2 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:

$$p(U_t = k|b_t,\theta) = \frac{\exp(\theta^k \cdot b_t)}{\sum_{j=1}^{M} \exp(\theta^j \cdot b_t)} \tag{5.30}$$

$$= \frac{1}{1 + \sum_{j \neq k} \frac{\exp(\theta^j \cdot b_t)}{\exp(\theta^k \cdot b_t)}} \tag{5.31}$$

$$= \left[1 + \exp(-\theta^k \cdot b_t) C_k\right]^{-1},\ C_k \stackrel{\text{def}}{=} \sum_{j \neq k} \exp(\theta^j \cdot b_t) \tag{5.32}$$

$$= \left[1 + K_k \exp(\theta^j \cdot b_t) + C_j\right]^{-1},\ K_k \stackrel{\text{def}}{=} \exp(-\theta^k \cdot b_t),$$
$$C_j \stackrel{\text{def}}{=} K_k \sum_{i \neq k, i \neq j} \exp(\theta^i \cdot b_t) \tag{5.33}$$

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

$$\begin{aligned}
&\nabla_{\theta^{k,i}} p(U_t = k|b_t, \theta) = \\
&\quad = \nabla_{\theta^{k,i}} \left[1 + \exp(-\theta^k \cdot b_t) C_k\right]^{-1} \\
&\quad = b_t^i \left[1 + \exp(-\theta^k \cdot b_t) C_i\right]^{-2} \exp(-\theta^k \cdot b_t) C_i \\
&\quad = b_t^i \left[1 + \exp(-\theta^k \cdot b_t) C_k\right]^{-1} \left[1 - \left[1 + \exp(-\theta^k \cdot b_t) C_k\right]^{-1}\right] \\
&\quad = b_t^i p(U = k|b_t, \theta) \left[1 - p(U_t = k|b_t, \theta)\right]
\end{aligned} \tag{5.34}$$

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

$$\begin{aligned}
&\nabla_{\theta^{j,i}} p(A = k|b_t, \theta) = \\
&\quad = \nabla_{\theta^{j,i}} \left[1 + K_k \exp(\theta^j \cdot b_t) + C_j\right]^{-1} \\
&\quad = -b_t^i p(U = k|b_t, \theta) \left[1 - (1 - C_j) p(U = k|b_t; \theta)\right]
\end{aligned} \tag{5.35}$$

Acknowledgment

The text of Chapter 5, with some modification, is a reprint of the material as it appears in N.J. Butko and J.R. Movellan, "Infomax Control of Eye Movements," *IEEE Transactions on Autonomous Mental Development*, 2(2):91–107 (2010) [6]. I was the primary author of this publication; the co-author supervised the research that forms the basis of this chapter.

Part III

Model Creation: Learning to Extract Information

Chapter 6

Learning To Look

6.1 Abstract

How can autonomous agents learn to look at visual targets? For example, how can they learn the correct pattern of voltages to send to their motors in order to achieve a desired gaze shift? We explore this seemingly simple question, and show that learning to look at visual targets contains a deep, rich problem structure, relating sensory experience, motor experience, and development. By capturing this problem structure in a generative model, we study how an optimal observer should trade off different sources of uncertainty in order to discover how their sensors and actuators relate. We implement our approach on three different robots, and show that both of them can quickly learn reliable looking behavior.

6.2 From Simulations to Physical Systems

There are many situations in which a robot may want orient to its cameras toward objects in its environment. A security camera may want to track a person in a building, a social robot may want to make eye-contact, or a teaching robot may want to give a student a clue about what object the student should focus on for her task.

In order to fixate an object, a robot must know what signal to send to

Figure 6.1: Different robots like Nobody (left), Einstein (middle), and Diego-san (right) have different sensorimotor capabilities. It is tedious and impractical to measure the specific sensorimotor parameters of many different robots. It would be better if each robot could learn to use and make sense of its sensorimotor capabilities in terms of its own experience.

its motors. This signal can be calibrated in a straightforward manner by the robot's engineers. They send an arbitrary signal to its eye-motors, and measure how many degrees its eyes move. After repeating this procedure several times for several motion signals, the solution to "what signal to send the motors to achieve a desired rotation" becomes a straightforward learning problem.

This process is tedious, and it is impractical to calibrate the sensorimotor parameters of many different robots. Even for different versions of the same robot, there may be slight variations in motor calibration, making mass deployment difficult. More importantly, from a developmental point of view, this calibration process is infeasible: a scientist measuring the properties of an infant's eye is not a prerequisite for an infant being able to look at things.

In this chapter, we consider how infants and robots may use their own sensorimotor experiences to learn to look. We analyze the structure of the problem and try to find solutions in a principled manner [2]. It is well known how objects in the scene project a 2D image onto a robot's camera, as well as how a robot's

Figure 6.2: This robot is currently looking at the car, but he would like to look at the beach (starred). What command should he send to his servo motors? Can the robot learn what command to send from developmental experience?

cameras generally move through space. We can derive an algorithm for learning to look by encoding these physical constraints formally into a generative model [138]. The generative model begins with formal models of the relationships among three components:

1. How the appearance of the scene changes over time.

2. How the image collected by a robot's cameras changes after a motor command reorients its field of view.

3. How the physical parameters of a robot's motor system cause a specific reorientation of the field of view for any given motor command.

Given this formal structure, the robot can simultaneously infer the appearance of the scene, the kinematics of its eye motion, and the direction of the eyes. This inference problem has a special mathematical structure: it is a conditional Gaussian

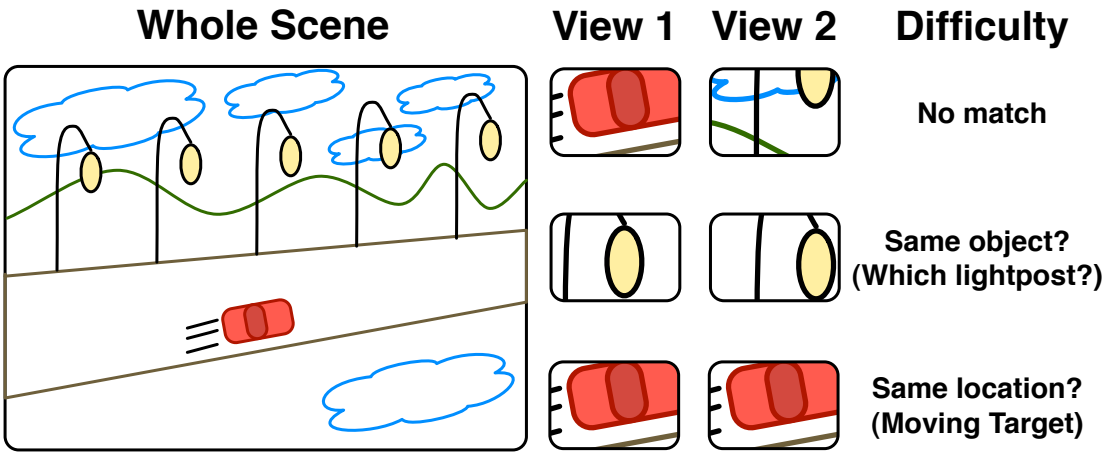


Figure 6.3: Matching objects in two consecutive images may fail for many reasons. 1) After moving its camera, there may be no objects in common. 2) Common objects may be present at regular intervals in the environment, and give systematic false matches. 3) Objects may move; assuming a matched object is in the same location may give a corrupt training signal.

process and thus efficient algorithms can be used to solve this inference problem robustly [138].

6.2.1 Teaching yourself

When engineers calibrate a robot, they measure how far its motors move in response to each different motor command; in order for a robot to calibrate itself, it must deduce the size of this motion autonomously. It may not be meaningful for the robot to measure such a motion in degrees; instead, "pixels of visual displacement" is more in keeping with the robot's sensorimotor experiences.

One way for the robot to discover how to look at things is to measure the distance of an object, in terms of pixel displacement before and after an eye movement. This idea, while basically sound, has some potential complications, as illustrated in Figure 6.3. In order to teach itself how to look, the robot needs a robust way to measure the motion offset caused by each camera movement.

Consider the robot's sensors and actuators and their relation to the basic

structure of the world. This relationship can be encoded in a generative probabilistic model. Generative models force us to explicitly explore the full structure of the problems that intelligent, developing agents face. In return, they often offer natural compromises to dealing with sources of ambiguity like those in Figure 6.3. For example, when Marks *et al.* considered the generative process in tracking non-rigid face deformation, they found an optimal tradeoff between optic-flow based tracking methods and template-based tracking methods [138].

Using the machinery of Bayesian inference, the robot can account for the exceptions in Figure 6.3 naturally, and in the right way. Specifically, we show that there is a tradeoff among three quantities: where you expect to look, what you expect to see, and how unsure you are about what you expect to see.

6.3 Generative Model

Two sources of information are available to the robot moment to moment. First, the robot knows what commands U_t it is sending to its motors (motor information). Second, it senses an array of pixels Y_t (sensory information). The robot must infer the hidden causes that explain the changes in Y_t induced by U_t. While the model is explained in full detail in the Appendix, Section 6.8.2, we present a brief overview here.

The generative process for the robot's experiences is illustrated in Figure 6.4. The variables in the generative model are summarized as:

- U_t: "Action," the motor command that the robot just sent. The camera movement starts when U_t is sent, and completes a short time later.

- Y_t: "Sensor," the entire image that the robot sees after the eye movement completes. Here, we assume that by the time Y_t is collected, the camera is no longer in motion. The image Y_t is sampled from the light values present in the scene, with zero-mean *i.i.d.* Gaussian noise on each pixel.

- $Y_{t,p}$: A single pixel value, at location p of that image.

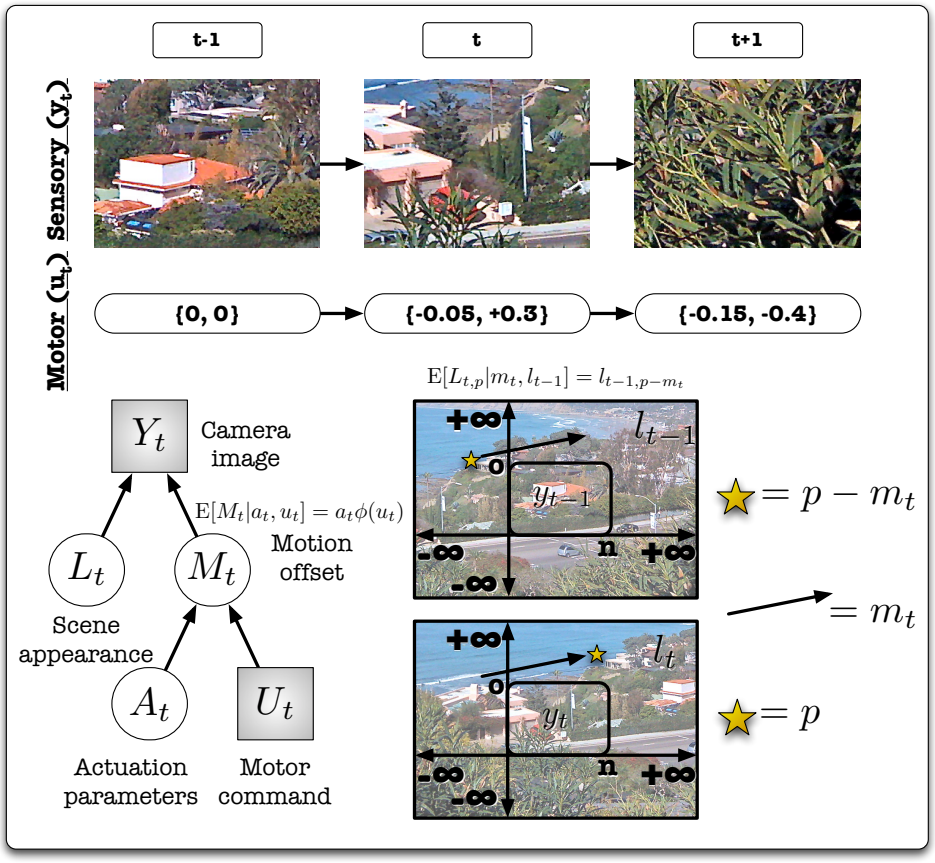


Figure 6.4: *Top:* The camera image Y_t changes after the robot sends a motor command U_t. *Left:* Open circles denote components of the hidden state, which together explain sensorimotor experience. *Right:* Illustration of the generative process.

- L_t: "Light," the appearance of the whole scene after the eye movement completes. The light values present in the scene are assumed to change over time with zero-mean, *i.i.d.* Gaussian noise at each point in the scene.

- $L_{t,p}$: The appearance of the scene at the single point, which is currently located at point p in the robot's sensor coordinate system. If p is outside the bounds of the sensor array, the robot cannot see this point in the scene right now, but it still has an appearance.

- A_t: "Actuation parameters," how the robot's motors work. Over time, a

robot's gears may rust and stiffen, or become loose. Here, we simulate these potential changes by letting A_t drift with zero-mean Gaussian noise and a fixed covariance.

- M_t: "Motion," an offset from the previous location, induced by an eye movement U_t. M_t is drawn from a Gaussian distribution with mean $A_t\phi(U_t)$ and a fixed covariance matrix, where $\phi(\cdot)$ is a known function of the action U_t.

- $L_{t-1,p-M_t}$: The previous appearance of $L_{t,p}$, before the robot started an eye movement by sending command U_t, causing motion offset M_t.

The goal of looking somewhere can be equated to the goal of achieving a particular motion offset $m_t^\star$. It is helpful for the robot to know the actuation parameters A_t in order to look at $m_t^\star$. Ultimately, in order for the robot to learn to look, it must infer A_t, *i.e.*, it has to learn how its own body works.

6.3.1 Implementation parameters

In implementing the approach above, many free parameters need to be chosen. The parameters used are listed in the Appendix, Table 6.3.

6.4 Learning Actuation Parameters

Learning the actuation parameters A_t entails computing

$$p(a_t \mid y_{1:t}, u_{1:t}) \tag{6.1}$$

While this computation can be difficult, it is easy to estimate the actuation parameters A_t if the trajectory of motion offsets, $m^*_{1:t}$, is known. If $m^*_{1:t}$ is known, the problem is Gaussian and thus

$$p(a_t \mid m^*_{1:t}, u_{1:t}) = \mathcal{N}(a_t, \bar{\mu}_{A_t}, \bar{\Sigma}_{A_t}) \tag{6.2}$$

can be computed using a Kalman filter, where $\bar{\mu}_{A_t}$ and $\bar{\Sigma}_{A_t}$ are the mean and variance estimates given by the Kalman filter [1]. It is likewise easy to learn the

appearance of each point in the scene $l_{t,p}$ when given a motion offset trajectory:

$$p(l_{t,p} \mid m^*_{1:t}, y_{1:t}) = \mathcal{N}(l_t, \bar{\mu}_{L_{t,p}}, \bar{\sigma}^2_{L_{t,p}}) \tag{6.3}$$

where $\bar{\mu}_{L_{t,p}}$ and $\bar{\sigma}^2_{L_{t,p}}$ are the mean and variance estimates given by separate Kalman filters for each point p in the scene.[1]

Both inference problems require knowing the motion offset trajectories $m^*_{1:t}$. In practice, at each t, we make the best guess possible about m^*_t given the previous guesses, $m^*_{1:t-1}$, and new available information:

$$m^*_t = \underset{m_t}{\text{argmax}}\ p(m_t \mid m^*_{1:t-1}, y_{1:t}, u_{1:t}) \tag{6.4}$$

$$= \underset{m_t}{\text{argmax}}\ p(m_t \mid y_t, u_t, \bar{\mu}_{A_{t-1}} \bar{\Sigma}_{A_{t-1}}, \bar{\mu}_{L_{t-1}}, \bar{\sigma}^2_{L_{t-1}}) \tag{6.5}$$

Thus, the previous Kalman filter estimates of A_{t-1} and L_{t-1} are critical for estimating m^*_t, which in turn is critical for estimating the Kalman filter estimates of A_t and L_t.

In the Appendix, Section 6.8.3, we show that m^*_t maximizes a function $g(m_t)$ of three terms:

$$g(m_t) = -\overbrace{\left[m_t - \hat{\mu}_{A_t}\phi(u_t)\right]^T \hat{\Sigma}^{-1}_{A_t} \left[m_t - \hat{\mu}_{A_t}\phi(u_t)\right]}^{Predicted\ Motion\ Match} - \sum_{p \in [1,n]\times[1,o]} \underbrace{\frac{(y_{t,p} - \hat{\mu}_{L_{t,p-m_t}})^2}{\hat{\sigma}^2_{L_{t,p-m_t}}}}_{Image\ Match} - \sum_{p \in [1,n]\times[1,o]} \underbrace{\log(\hat{\sigma}^2_{L_{t,p-m_t}})}_{Uncertainty\ Penalty} \tag{6.6}$$

where $\hat{\mu}_{A_t}$, $\hat{\Sigma}_{A_t}$, $\hat{\mu}_{L_t}$, and $\hat{\sigma}^2_{L_t}$ are functions of the Kalman filter estimates, defined in the Appendix, Section 6.8.3. Each term has an important meaning:

- **Predicted Motion Match:** $\hat{\mu}_{A_t}\phi(u_t)$ is the motion predicted for action u_t by the Kalman filter estimates of A_{t-1}. The predicted motion match puts a quadratic cost on m_t that deviate from this prediction, but mitigates that cost by $\hat{\Sigma}_{A_t}$, the sense of uncertainty in the motion parameters A_t.

[1] From an implementation perspective, this requires two images, a mean image, representing $\bar{\mu}_{L_t}$, and a variance image representing $\bar{\sigma}^2_{L_t}$. Using this two image representation, it is computationally easy to maintain and update millions of Kalman filters simultaneously.

- **Image Match:** $\hat{\mu}_{L_{t,p-m_t}}$ is the pixel appearance predicted by the Kalman filter estimate of L_{t-1}. The image match puts a quadratic cost on y_t that deviate from this prediction, but mitigates that cost by $\hat{\sigma}^2_{L_{t,p-m_t}}$. Thus, the penalty is lower in regions of the scene where the appearance is unknown because it has never been seen before, or hasn't been seen in a long time.

- **Uncertainty Penalty:** $\hat{\sigma}^2_{L_{t,p-m_t}}$ is related to the uncertainty of the appearance of the scene. The uncertainty penalty places a logarithmic cost on appearance uncertainty. Since the image match gets less penalty in regions of low confidence, the robot might be prone to infer that each new image comes from a region of the scene it has never seen before. The uncertainty penalty discourages such inferences.

The function $g(m_t)$ gives us a way to score and compare candidate motion offsets m_t, but evaluating a single m_t is somewhat expensive, involving every pixel $Y_{t,p}$ in the sensor image Y_t. In practice, we search for the maximum using a course to fine strategy.[2]

The general sketch of inferring the parameters of motion A_t can be described as:

1. Choose a new action u_t, observe a new image y_t.
2. Search the space of M_t for $m_t^* = \text{argmax}_{m_t}\, g(m_t)$.
3. Update the coordinates for the scene using m_t^*.
4. Update the Kalman filter estimates $\bar{\mu}_{L_{t,p}}$ and $\bar{\sigma}^2_{L_{t,p}}$ with m_t^* and y_t [1].
5. Update the Kalman filter estimates $\bar{\mu}_{A_t}$ and $\bar{\Sigma}_{A_t}$. with m_t^* and u_t [1].
6. Set $t = t + 1$. Go to step 1.

[2]For sensor images Y_t of size 320×240, we search a 213×160 window around the predicted motion offset $\bar{\mu}_{A_{t-1}}\phi(u_t)$ at a granularity of 10 pixels, and then search exhaustively in a 25×25 radius around the maximum of the first search. Thus a total of 961 candidate m_t^* are evaluated after each eye movement. For the coarse search, we decimate the observed image to 80×60 pixels, and for the fine search we decimate it to 160×120 pixels. The entire search process requires about 100 ms.

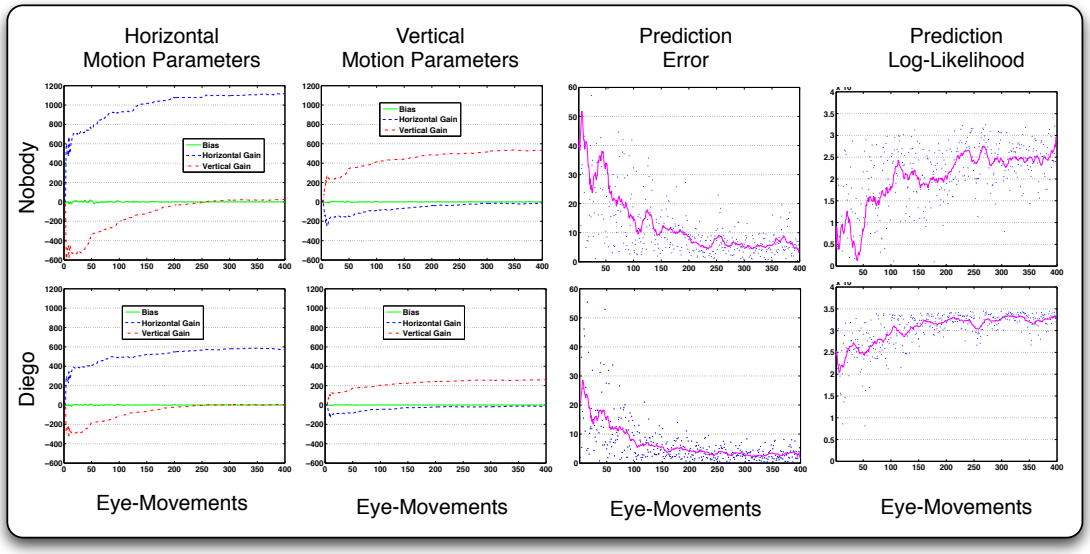


Figure 6.5: *Column 1:* Learning trajectory of $\bar{\mu}_{A_{t,1:3}}$, Kalman filter estimates of $A_{t,1:3}$ in Equation (6.7). *Column 2:* Learning trajectory of $\bar{\mu}_{A_{t,4:6}}$, Kalman filter estimates of $A_{t,4:6}$. *Column 3:* Euclidean distance from $\bar{\mu}_{A_{t-1}}\phi(u_t)$ to m_t^* decreases with learning. *Column 4:* Likelihood of the intended target $g(\bar{\mu}_{A_{t-1}}\phi(u_t))$ increases with learning.

6.5 Experiments 1: Learning A_t

6.5.1 Experiment 1.1: Nobody & Diego, 2 actuators

We implemented the above approach on two robots:

- **Nobody:** A simple surveillance robot consisting of a webcam and two servo-motors on a pan-tilt platform.
- **Diego-san:** A robot with similar level of complexity to the human body, consisting of 88 pneumatic degrees of freedom in the body, and 6 motors for facial expressions and eye movements.

For this initial set of experiments, we used a feature function $\phi(\cdot)$ that added a bias dimension, and both robots used two actuators to pan and tilt their cameras.

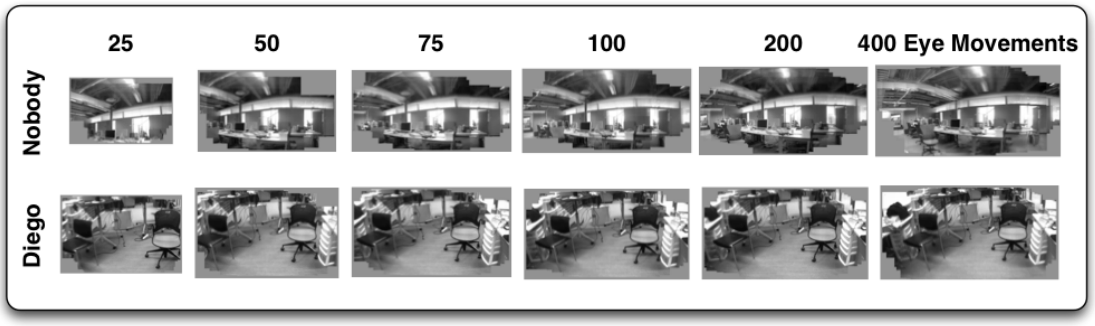


Figure 6.6: The mean estimates $\bar{\mu}_{l_t}$ of the appearance of the scene, at all locations, p, at time points, $t = \{25, 50, 75, 100, 200, 400\}$, during each robot's learning.

Thus, the motion model was

$$\mathrm{E}[M_t \mid A_t, U_t] = \begin{bmatrix} A_{t,1} & A_{t,2} & A_{t,3} \\ A_{t,4} & A_{t,5} & A_{t,6} \end{bmatrix} \begin{bmatrix} U_{t,1} \\ U_{t,2} \\ 1 \end{bmatrix} \tag{6.7}$$

Both robots were initialized with the same parameters (Table 6.3), and moved their eyes according to an identical Brownian motion trajectory for a total of 400 eye movements. On each fixation t, they computed $\mathrm{E}[M_t \mid \bar{\mu}_{A_{t-1}}, u_t] = \bar{\mu}_{A_{t-1}}\phi(u_t)$, the place that they expected to look, as well as m_t^*, their best guess of where they actually looked.

The learning trajectories of learning are shown in Figure 6.5. The learned mean estimates $\bar{\mu}_{A_t}$ of the parameters of motion stabilize given sufficient experience. For both robots, horizontal and vertical motion are learned to be independently controlled by different motors, *i.e.* $A_{t,2}$ and $A_{t,4}$ in Equation (6.7) were estimated to be near 0. Moreover, there was found to be no bias in the motor movement, *i.e.* $A_{t,3}$ and $A_{t,6}$ were estimated to be near 0.

Over the course of learning, the distance decreases between the expected fixation target $\bar{\mu}_{A_{t-1}}\phi(u_t)$ and the robot's best guess *a posteriori* of the actual fixation target, m_t^*. After learning, the robot is able to accurately predict the eye movement caused by a given motor signal. The likelihood of the intended target $g(\bar{\mu}_{A_{t-1}}\phi(u_t))$ increases over time, indicating that both robots are better able to

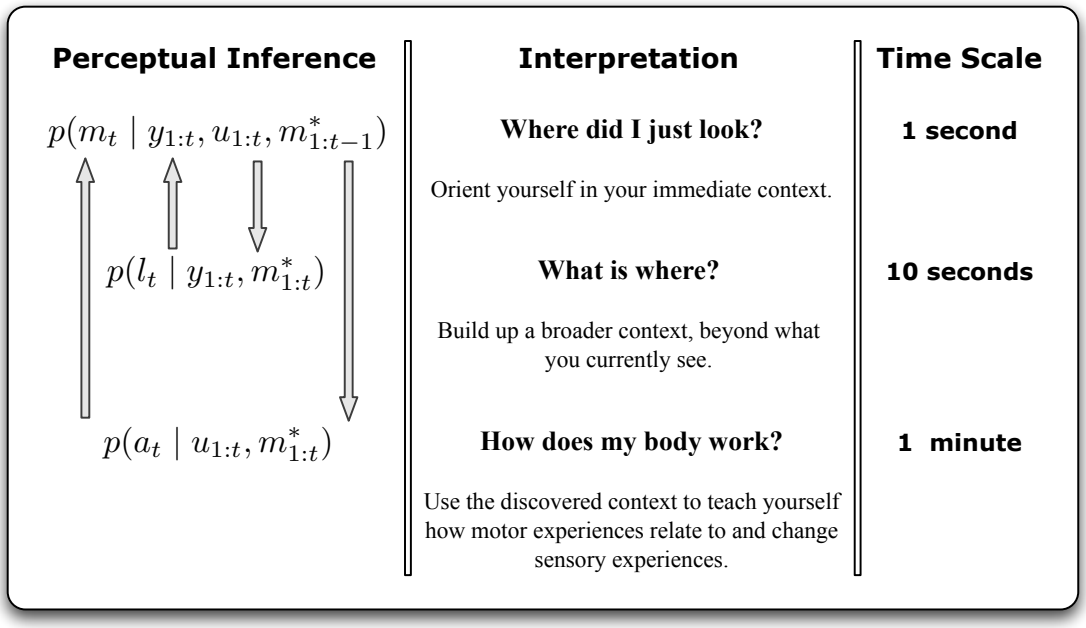


Figure 6.7: Temporal dynamics of sensorimotor active perception while learning to look.

predict the sensory consequences of their eye movement. Together, these show that both robots develop good models of the sensorimotor consequences of their actions.

The learning of the motor parameters A_t relies heavily on the robot's estimate $\bar{\mu}_{L_{t,p}}$, the robot's memory of the appearance of the scene around it, which is bigger than the things it can see with any single fixation. As the robot has more experience looking around its environment, it develops a better idea of "what's out there," as shown in Figure 6.6.

6.5.2 Temporal dynamics of sensorimotor learning

Learning to look entails making inferences about three hidden causes in the robot's sensorimotor environment, M_t, L_t, and A_t. Inferences about these variables occur over different timescales (Figure 6.7).

1. **Fast:** Every second of every minute, the robot is making perceptual inferences about M_t, orienting itself in its current environment. After each eye movement, it computes the optimal guess about the motion that just occurred, $m_t^* = \text{argmax}_{m_t} p(m_t \,|\, y_{1:t}, u_{1:t}, m_{1:t-1}^*)$. This fast perceptual inference takes place on a timescale smaller than a single second. These moment to moment inferences form the foundation for the rest of learning to look.

2. **Medium:** Across a few fixations, the robot can learn about L_t, the appearance of the scene it inhabits. This entails building up a representation of the current environment that is broader that what the robot can just see right now. After a few fixations, in less than 10 seconds, the robot has inferred enough of its surrounding context to improve the moment to moment inferences about where it is looking, M_t.

3. **Slow:** As the robot begins to reliably determine where it's looking, it learns about A_t, how its body works. In turn, this allows it to predict the sensorimotor output of each action. This improved ability to predict in turn increases the quality of its inferences about M_t.

6.5.3 Experiment 1.2: Einstein, 5 actuators

In our experiments with Nobody and Diego, the direction of eye gaze was controlled by 2 motors. In contrast, the Einstein robot has 5 motors that control the direction of his gaze. Two contribute to horizontal rotation of the cameras: (1) Eye left/right, (2) Head left/right. Three contribute to vertical rotation of the cameras: (3) Eye up/down, (4) Head upper nod, (5) Head lower nod.

Recall that $\text{E}[M_t \mid A_t, U_t] = A_t \phi(U_t)$. In Equation (6.7), there were two motors and a bias feature. In this experiment, we disregard the bias, because it played little role in learning previously. For Einstein's five motors, the motion

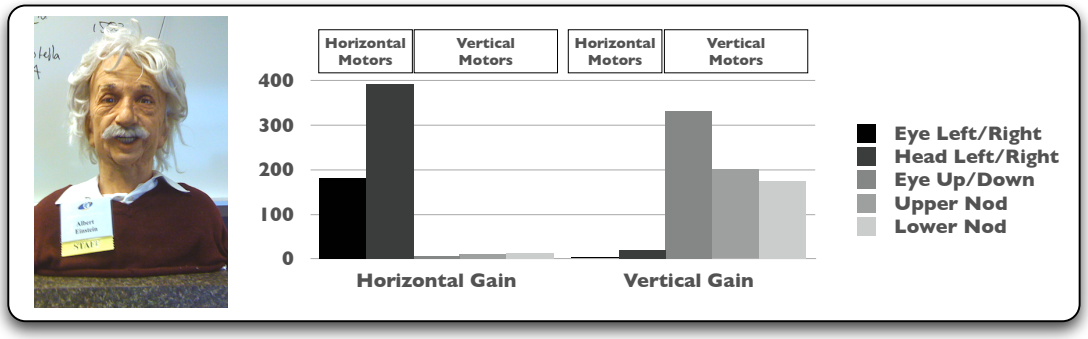


Figure 6.8: Einstein's learned mean estimates, $\bar{\mu}_{A_t}$ (absolute value). Horizontal Gain shows the estimate of $A_{t,1:5}$ in Equation (6.8), while Horizontal Gain shows the estimate of $A_{t,6:10}$.

model becomes

$$\mathrm{E}[M_t \mid A_t, U_t] = \begin{bmatrix} A_{t,1} & A_{t,2} & A_{t,3} & A_{t,4} & A_{t,5} \\ A_{t,6} & A_{t,7} & A_{t,8} & A_{t,9} & A_{t,10} \end{bmatrix} \begin{bmatrix} U_{t,1} \\ U_{t,2} \\ U_{t,3} \\ U_{t,4} \\ U_{t,5} \end{bmatrix} \tag{6.8}$$

Using exactly the same parameters of learning that were used by Diego and Nobody, Einstein learned the contribution of each of his motors to translations of pixels across the retina. Values of A_t near 0 represent little contribution, and values with high magnitude represent a large contribution. Figure 6.8 shows the absolute values of the elements of the A_t matrix that were learned by Einstein. The "Horizontal Gain" plot shows abs($\bar{\mu}_{A_{t,1:5}}$), the estimates of $A_{t,1:5}$, and the "Vertical Gain" plot shows abs($\bar{\mu}_{A_{t,6:10}}$), the estimates of $A_{t,6:10}$. Two motors, the Eye and Head left/right motors, were found by Einstein to contribute to horizontal translations of pixels across his retina. Three motors, the Eye up/down and Head lower and upper nod motors, were found by Einstein to contribute to vertical translation of pixels across his retina.

6.5.4 Experiment 1.3: Nobody, 5 actuators

While Einstein has five motors that affect his direction of gaze, the Nobody robot has no body beyond its two motors. In this experiment we endow the Nobody robot with three phantom limbs, so that it has the same number of motor commands at Einstein, but only two of these signals actually affect the motion of the eyes.

This is similar to a problem faced by infants in development; when figuring out how their bodies work, it is conceivable that infants may initially attempt to look using their tongues, toes, and elbows; eventually they discover that only a limited subset of motor outputs actually affects eye movement.

Figure 6.9 shows the learning trajectory of the gains learned by Nobody for each of its motor contributions to horizontal and vertical eye movement. Nobody quickly learns that one of its motors contributes to horizontal translation of pixels across the retina, one of its motors contributes to vertical motion of pixels across the retina, and three of its motors do not contribute to eye movement.

6.5.5 Experiment 1.4: kidnapping Nobody

Kidnapping is a standard robotics experimental paradigm. In this manipulation, the robot is physically carried by the experimenter to a new environment [139, 140]. It is desirable, but often difficult, to develop algorithms that are robust to kidnapping events. Because kidnapping is so disruptive to robotic learning, a modified paradigm is often applied in which the experimenter first informs the robot, "I'm going to kidnap you now."

We allowed the Nobody robot to learn to look for 100 fixations, and then kidnapped it, without informing it, and left it to discover on its own that it was in a new environment. We then observed the effect on learning.

The kidnapping was disruptive on a short timescale: Nobody was confused about where it was looking, and had to decide arbitrarily how to situate itself in a new context. However, after a few eye movements and less than ten seconds, Nobody was able to begin building a representation of its new surroundings. This

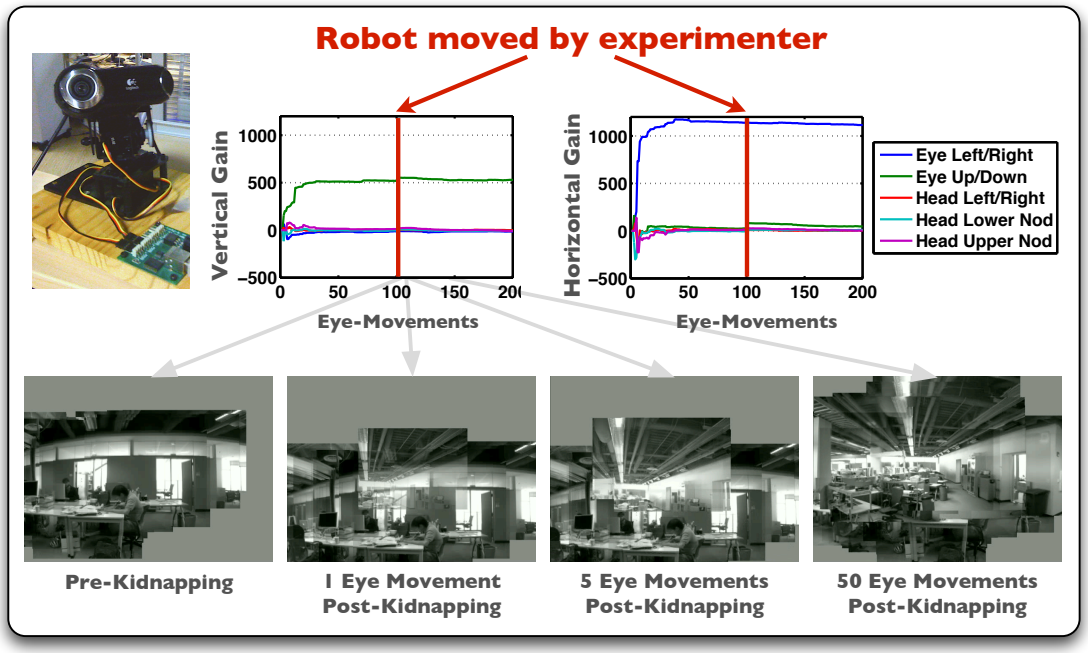


Figure 6.9: We performed two experimental manipulations to Nobody's learning: we endowed it with three phantom limbs, and, after 100 eye movements, we kidnapped it and brought it to a new environment. Nobody's learning was robust to both of these manipulations.

recovery on a medium timescale led to remarkable stability in the estimates $\bar{\mu}_{A_t}$ over a long timescale. In Figure 6.9, the time of kidnapping is marked. The disruption on a short time scale is visible in the estimate μ_{L_t} one eye movement post fixation. However, by just five eye movements, Nobody is beginning to build up a new context, which is overlaid on top of the old context. This new context allows the $\bar{\mu}_{A_t}$ estimates learned by Nobody to remain stable post-kidnapping. By fifty eye movements post abduction, Nobody has developed a detailed view of its new environment.

6.5.6 Observed failures

While the presented approach to learning to look was observed to be robust across several robots of different morphologies, as well as some extreme experimen-

tal manipulations, we did empirically observe some failures.

The Appendix, Section 6.8.1 gives the full list of parameters. The parameter values given in Table 6.3 were used in all experiments in this chapter, but there were configurations of parameters that made learning fail. Particularly, decreasing σ_Y^2 seemed to be detrimental. The σ_Y^2 parameter represents the reliability of the sensory apparatus. Keeping it somewhat high helps to account for lens distortions in the periphery, as well as the pixel rotations induced when the robot rotates horizontally at high vertical orientations.

Additionally, some environmental factors were disruptive to learning. When humans stood a few feet in front of any robot during early stages of learning, it was difficult for that robot to estimate the camera motion offset M_t reliably. This is because people took up most of the robot's relatively narrow field of view; also they moved, *e.g.* by gesturing during normal conversation. In such cases, most of the robot's visual scene was changing moment to moment in a way that was unrelated to its camera motion, and so it was difficult for the robot to estimate accurate translations of pixels across its image plane.

6.6 Two Model Extensions

In this section we consider two separate extensions to the model. The first allows us to estimate the temporal dynamics of eye movement, and the second gives a principled account for gaze component coordination.

6.6.1 Extension 1: Learning the temporal dynamics of our actions

Until now, we have not considered the times between when a motor command U_t is sent, and the motion offset M_t completes, and an image Y_t is captured. However, real eye movements take place along temporal trajectories. There is some latency between when a motor command is requested and when it is executed. Once the motor begins to move, it takes some time to reach its destination.

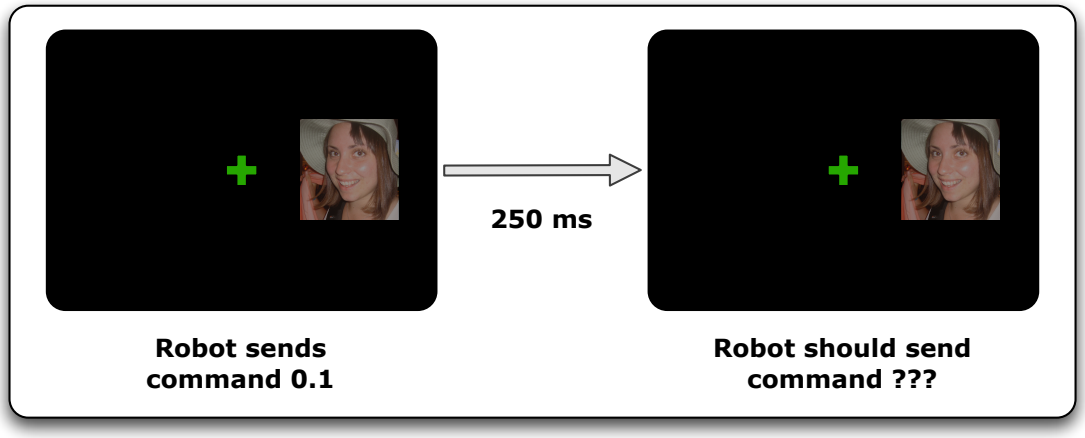


Figure 6.10: Einstein sees a face 15° to his right, and sends a command of 0.1 to look at the face. 250 ms later, he sees the face in the same spot, 15° to his right. Is the face moving? If so he should send another eye movement command to track it; if not, his eye movements are delayed greater than 250 ms, and sending a second eye movement will lead to overshooting the target.

From a practical perspective, it is very important for a robot to have knowledge of these temporal dynamics.

Consider the scenario described in Figure 6.10. In this scenario, Einstein's experiences are consistent with at least two hypothesis. (1) His eyes haven't moved yet. (2) The face is moving rightward. These outcomes have consequences for Einstein's decisions. If (2) is the case, he should generate another rightward eye movement. Failing to generate such a request will lead to sluggish, unresponsive tracking. If (1) is the case, a second eye movement request would cause him to overshoot the target.

This was the dilemma we found from our empirical investigations when we asked Einstein to track faces after he had learned to look: Einstein found faces faster than his motors moved, and either he could send too many eye movement commands, which led to overshoots, followed by corrections which were themselves overshoots, or we could artificially limit his rate of movement to some sufficiently high time delay, and have an unresponsive tracking system.

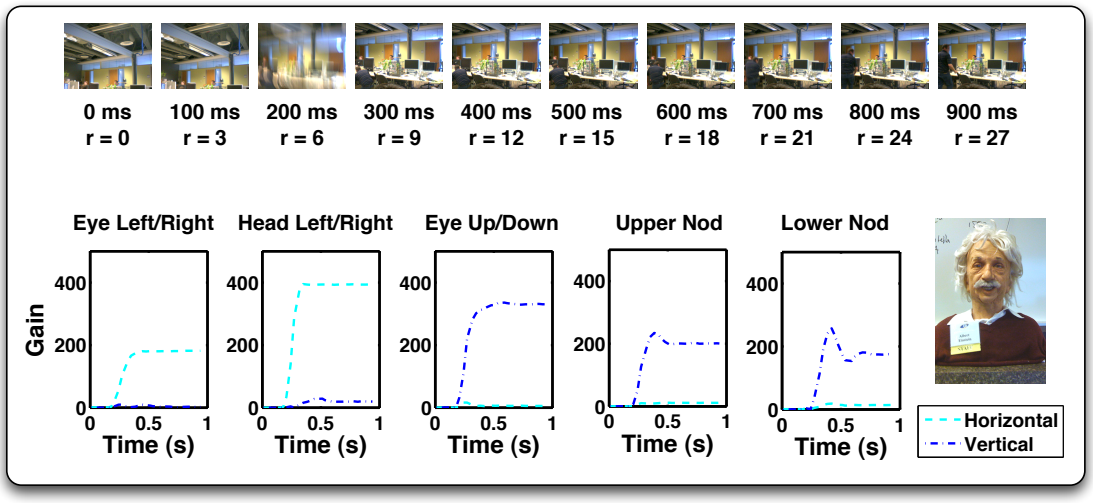


Figure 6.11: Einstein learns the temporal dynamics of each of his motors.

To overcome this limitation, Einstein needs a temporal model of his own eye movements. From the moment that Einstein sends a command to his motors, he saves a trajectory of images, collected at 30 Hz. for one second. Thus, for each eye movement, there are thirty frames. Each of these is used to learn a separate set of parameters $A_{r,t}$, where t indexes the eye movement number, and r indexes number of frames since an eye movement command was sent. Under this new model, the motion offset r frames after an eye movement is given by $\mathrm{E}[M_{r,t}|A_{r,t}, U_t] = A_{r,t}\phi(U_t)$. For a more detailed description, see Appendix, Section 6.8.5.

6.6.2 Experiment 2: Einstein, dynamics model

Einstein learned using the same motion model from Equation (6.8), but now using the extended "motor dynamics" model. Each learned $A_{r,t}$ is shown in Figure 6.11. As before, each motor is found to contribute exclusively to either horizontal or vertical motion. But now, it becomes clear that the motors only start moving between 150 and 200 ms after the motion command U_t is sent; they finish moving between 400 and 750 ms after the motor command. Different motors have different time courses, and specifically, the motors that move the head vertically seem to

recoil: they initially move too much, and then come back, and eventually stabilize on a mean position.

Einstein can then use this model of the knowledge of dynamics to answer the question of what he should do if he sees a face to the right after 250 ms: At that point in time, he has started moving, and is halfway to his destination. So the face has moved the right, but less than he initially thought: only 7.5° instead of 15°. He should send an eye movement signal that is half strength (0.05).

6.6.3 Extension 2: Learning control and coordination

Einstein is over actuated, in the sense that a gaze shift can be achieved by multiple motor commands. He can look 15° to the right by moving only his eyes, or look 15° to the right by moving only his head. He can even look 15° to the right by moving his head 25° to the right and his eyes 10° to the left. All three sets of motor commands lead to the same resulting gaze shift.

Given that Einstein has five motors that control his direction of gaze, how should he coordinate them to look at targets? In this section we appeal to the principle of maximum accuracy in the presence of signal dependent noise. In humans, the accuracy of eye movements decreases with target distance, with horizontal standard deviation increasing at a rate of 0.145 degrees per target degree [141].

We start by assuming that this relation holds true (with different parameters) for all gaze components. Previously, we had modeled

$$\mathrm{E}[M_t \mid A_t, U_t] = A_t \phi(U_t) \tag{6.9}$$

$$\mathrm{Cov}[M_t \mid A_t, U_t] = \Sigma_A \tag{6.10}$$

where Σ_A was a constant parameter. Now, let

$$\mathrm{E}[M_t \mid A_t, U_t] = A_t U_t \tag{6.11}$$

$$\mathrm{Cov}[M_t \mid A_t, N_t, U_t] = \mathrm{diag}(N_t U_t)^2 \tag{6.12}$$

where N_t is a random matrix of the same shape as A_t with elements that describe the contribution of each motor to horizontal and vertical standard deviation. The

goal is to find the action $u_t^\star$ that is expected to result in the camera offset m_t with minimum squared Euclidean distance to some desired gaze shift $m_t^\star$. In the Appendix, Section 6.8.7, we show that, for a fixed and known a_t, n_t,

$$u_t^\star = \left[\text{diag}(1^T(n_t \circ n_t)) + a_t^T a_t\right]^{-1} a_t^T m_t^\star \tag{6.13}$$

Note that $1^T(n_t \circ n_t)$ is a vector whose elements are the sum of the variances across each direction of gaze for each gaze component. For Einstein's 5 motors, $1^T(n_t \circ n_t)$ is a row vector of 5 elements, each corresponding to the sum of the variances induced by that motor. The matrix $\text{diag}(1^T(n_t \circ n_t))$ is 5×5.

The optimal gaze coordination rule given in Equation (6.13) has several intuitively appealing properties. First, when the elements of n_t are 0 (the noiseless condition), the optimal gaze rule is a pseudo-inverse. Second, as the overall variance increases, the value of the denominator decreases, yielding an undershoot in the presence of signal dependent noise. Third, the contribution of each motor is mitigated by its variance; thus, in the presence of multiple degrees of freedom, each with different reliabilities, the optimal gaze rule will weight more heavily the more reliable degrees of freedom. If head movement is less reliable than eye movement, then the eyes will contribute more to the shift in gaze direction than the head.

6.6.4 Comparison to human data

Abrams *et al.* measured the mean and standard deviation of saccades to targets at various horizontal eccentricities [141]. They observed a .145 degree of standard deviation per degree of target eccentricity (14.5%), and a 0.032 degree undershoot per degree of target eccentricity (3.2%). According to the optimal gaze rule for one degree of freedom and 14.5% signal dependent noise, the optimal undershoot is $1 - 1/(.145^2 + 1) = 2.1\%$ undershoot. This explains about 67% of the effect observed in humans.

Humans can move their heads in addition to their eyes. What does the optimal gaze rule tell us in this case? Let's assume that the head can move equally reliably as the eyes: 14.5% standard deviation. Then, Equation (6.13) says that the rotation should be split equally among the head and eyes, each rotating 49.48%

of the distance to the target. This gives a combined undershoot of 1.05%, half of the previous value. Table 6.1 shows how the gaze components change as a function of head movement accuracy. Even in the presence of 100% noise, the head should

Table 6.1: Modeled optimal head contribution to gaze shift.

Eyes std	Head std	Eye Rot. (deg)	Head Rot. (deg)	Undershoot (deg.)
14.5%	14.5%	49.48%	49.48%	1.05%
14.5%	25%	73.7%	24.8%	1.5%
14.5%	50%	90.5%	7.6%	1.9%
14.5%	75%	94.5%	3.5%	2.0%
14.5%	100%	96.0%	2.0%	2.0%
14.5%	125%	96.7%	1.3%	2.0%
14.5%	∞%	97.9%	0%	2.1%

account for 2% of the motion. To understand why this is, consider attempting to fixate a target at an eccentricity of 10°. If we attempt a gaze shift of 10° in an actuator with 100% signal dependent noise, on average, the resulting gaze shift will be 10° away from the target. Sometimes the gaze will shift too much, to 20°, and sometimes it will shift too little or not at all, to 0°. Attempting to shift gaze by 5° will, on average, result in a 5° motion error, to 0° or 10°. The expectation is that a 5° gaze shift attempt will on be closer to a 10° target than the 10° gaze shift attempt, and, in fact, attempting to shift gaze by 5° is better than not trying to shift the gaze at all. It only pays to not try to use a given actuator if the signal dependent noise is much greater than 100%.

In free viewing conditions, when subjects can coordinate both their head and their eyes, smaller total undershoots are predicted, regardless of the reliability of head movement. For realistic values of head movement accuracy (about 25% –50%), we would expect eye movement to account for about 75%–90% of the total gaze shift.

When the subject is restrained with their head fixed, and only able to move

their eyes, they may still attempt to move their head. In this case, on average, the head will move 100% less than intended. The optimal gaze coordination is to still attempt to move the head 2% of the distance to the target, and the eyes 96%. This would result in an average undershoot of 4%, because only the eyes actually move. This is closer to the undershoot observed in experiments, which was 3.3%.

The undershoot observed by Abrams *et al.* is compatible with an average head movement error of 125%. However, very little should be made of this. First, the experimental margin of error for the study of Abrams *et al.* is unknown, so it is unwise to put too much credit in a precise match to data. Second, our linear signal dependent noise model is only an approximation, and there are doubtless higher order noise effects. Thus, quantitative results in this domain have limited predictive power.

Rather, the following qualitative results are robust to slight experimental error and some higher order effects. In the presence of signal dependent eye movement noise that increases with eccentricity, and when the goal is to minimize the squared Euclidean distance to the desired saccade target,

- Undershoots are expected.
- When multiple gaze components can combine, a higher contribution is made by the components with less noise.
- When multiple gaze components can combine, undershoots are mitigated.
- When some gaze components are artificially restrained, undershoots are exacerbated.

The quantitative analysis in this section serves to illustrate these computationally motivated arguments.

6.6.5 Experiment 3: Einstein, noise model

An advantage of the signal dependent noise eye movement model in Equation (6.12) is that the noise parameters N_t can be learned by the robot using

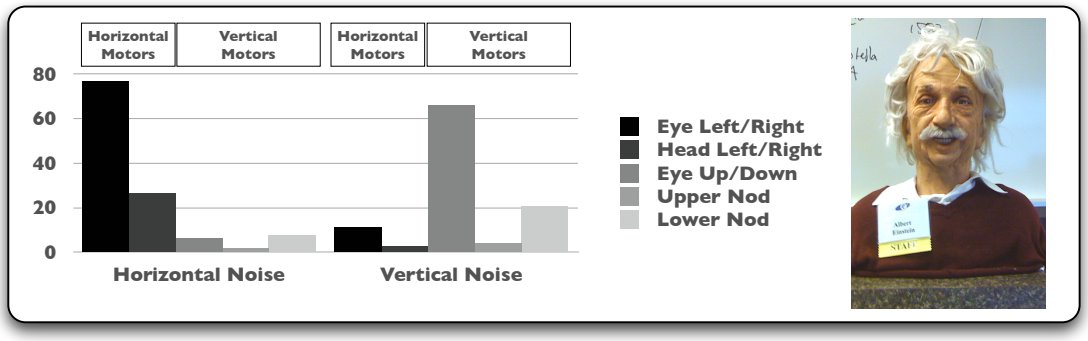


Figure 6.12: Einstein's learned motor noise parameters N_t.

exactly the same mathematical framework as the motion parameters A_t. The only difference is the training signal. While A_t was learned with a Kalman filter using the estimates m_t^* as observations, N_t can be learned with a Kalman filter using $\text{abs}(m_t^* - \bar{\mu}_{A_t} u_t)$ as observations (recall that N_t represents the contribution of each element of A_t to the average horizontal and vertical movement error, *i.e.* to the standard deviation).

In terms of the timescales of learning in Figure 6.7, learning about the motor noise N_t takes place on a long timescale (tens of minutes). This is because there must first be a reliable estimate $\bar{\mu}_{A_t}$ of the actuation parameters before the quantity $\text{abs}(m_t^* - \bar{\mu}_{A_t} U_t)$ gives a meaningful estimate of the motor uncertainty.

Based on Figure 6.11, we predicted that Einstein's eye movement were more reliable than head movements, and that vertical head movement would be particularly unreliable. We predicted this because the vertical head movement showed a strong recoil effect, and so the ending position may be less reliable. In fact, Einstein learned something quite different about his motors, which is shown in Figure 6.12.

As expected, horizontal motors were found to be the primary contributors to horizontal noise; vertical motors were the primary contributors to vertical noise. Unexpectedly, the eyes were found to lead to much higher error than head motors. Table 6.2 shows how much contribution the optimal gaze coordination rule gives to each motor for horizontal and vertical motion.

Table 6.2: Optimal contribution of each motor to horizontal / vertical motion.

	Eye L/R	Head L/R	Eye U/D	L. Nod	U. Nod
Horiz.	2.4%	96.8%	0 .0%	0.0%	0.3%
Vert.	0.0%	-0.3%	1.3%	95.5%	3.4%

The upper nod, which had a high amount of recoil, is used less in vertical gaze shifts than the lower nod, which had less recoil. But both of these are used more than the eyes for vertical motion. For horizontal motion, head motion is strongly preferred to eye motion.

Our model of motion is linear, and so any nonlinearity will inflate the noise estimates. Such nonlinearities may explain why eye motion is estimated to be so noisy when compared to head motion. First, the head is rotated directly by motors while the eyes are rotated indirectly through a series of parts, which may induce non-linearities in the transformation from motor rotation to eye rotation. Second, the head and eyes don't rotate about the same axis, which breaks the assumption that different gaze components simply add. Both of these sources of nonlinearity can be accounted for to reduce the estimated errors in eye movement.

In practice, when Einstein makes gaze shifts that he calculated to be optimal (by making very large head movements and very small eye movements), his gaze shifts are very accurate. Strikingly, when Einstein makes large head movements, there is often a large amount of recoil; when the jostling of his head stops, he is looking directly at the saccade target.

6.7 Discussion

Neural Implementation

The algorithms above describes the consequence of our generative model of learning to look. According to this model, at each fixation, the robot should shift its map of how the scene (L_t) looks to line up with what it's about to see.

A strikingly similar remapping process has been observed in monkey lateral infraparietal cortex (LIP) [67]. In this case, what the generative model tells us would be a good idea to do if you want to solve a particular problem, biology also seems to think is a good idea.

Nature or Nurture?

We present a model by which a robot may "learn to look." This is an important problem to consider because each robot may have a different configuration of motors. The motors from robot to robot may have different range of reasonable control values. Some robots may be fixed to a point in space, with only the ability to rotate their cameras; others may be able to move easily in three dimensional space. Thus it is critical that each robot be able to use its own sensorimotor experience to figure out how to use its motors.

However, we present a generative model by which the robot can anchor its motor experience in its sensory experience, and discover how its motors work. Our three robots were "born" with a generative probabilistic model and machinery for doing Bayesian inference.

Our experiments do not show how people learn to look. They do not show what cognitive processes people are born with, or what they must learn from experience. However, our experiments do show that the environment that people live in contains a statistically rich structure that supports learning from experiences. This approach provides an effective counter to poverty of the stimulus arguments, which have been historically used to argue that certain aspects of intelligence and behavior must be innate.

Sensorimotor Development

In this document we have laid out three properties of physical eye movement that we targeted for developmental learning: (1) learning to look, (2) learning about the temporal dynamics of eye movements, (3) learning about signal dependent noise. However, there are many other properties of physical eye movements that a robot may find it useful to be aware of:

1. 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.

2. The quality of image frames collected during an eye movement.

3. The likelihood that objects in the robot's environment will move spontaneously.

We set out to discover how a robot could learn to look based only on its sensorimotor experiences; the approach that we took was powerful and robust enough to enable three separate robots to each learn to look at intended visual targets. The same approach is rich enough to ultimately afford solutions to at least these other four problems.

Problem 1) Figure 6.6 shows each robot's entire scene, from one motion extreme to the other. By comparing its instantaneous field of view to the total, the robot can solve this problem.

Problem 2) The likelihood function $g(m_t^*)$ measures how well what the robot sees matches what it remembers. As the camera image becomes blurred and distorted from motion, the match between $Y_{t,p}$ and $\bar{\mu}_{L_{t,p}}$ will plummet, as reflected in the dynamics of $g(m_t^*)$ over the course of an eye movement. This can give a robot an idea of when to trust its sensors, and when to ignore them.

Problem 3) Motion is captured by temporal variation in the appearance of the robot's scene. By empirically estimating this variance at each location, the robot not only can estimate how much objects in its scene move, but it can also figure out *where* they are likely to move. *E.g.* objects on the floor are more likely to move than objects on the ceiling.

6.8 Appendix

Unless otherwise stated, capital roman letters are used for random variables, small letters for specific values taken by random variables, indexes, and dimensions. Greek letters are reserved for parameters and for values computed from given information. All random variables are defined with respect to a common probability space $(\Omega, \mathcal{F}, \mathbf{P})$. For random variables, we use $A \in \mathbb{R}$ as shorthand for $A : \Omega \to \mathbb{R}$. When the context makes it clear, we identify probability functions by their arguments: *e.g.*, $p(a, b)$ is shorthand for the joint probability mass or joint probability density that the random variable A takes the specific value a and the random variable B takes the value b. We work with discrete time stochastic processes, with the parameter $\Delta t \in \mathbb{R}$ representing the sampling period. We use subscripted colons to indicate collections or sequences: *e.g.*, $A_{1:t} \stackrel{\text{def}}{=} \{A_1 \cdots A_t\}$. $\mathrm{E}[A]$ denotes the expected value of A. $\mathrm{Var}[A]$ denotes the variance of scalar random variable $A \in \mathbb{R}$ and $\mathrm{Cov}[B]$ denotes the covariance matrix of a random vector. For fixed μ and Σ we use the notation

$$p(a) = \mathcal{N}(a, \mu, \Sigma) \stackrel{\text{def}}{=} \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\{-\frac{1}{2}(a-\mu)^T\Sigma^{-1}(a-\mu)\} \tag{6.14}$$

to represent the value of the normal probability density function with mean μ and covariance Σ evaluated at a. When a is a scalar, we use σ^2 to denote the variance. When A is drawn from a normal distribution with a given mean μ and covariance Σ we write

$$A \sim \mathrm{Normal}(\mu, \Sigma) \tag{6.15}$$

6.8.1 Learning to Look model components definitions

Dimensions

$k' \in \mathbb{Z}^+$: The number of motors

$k \in \mathbb{Z}^+,\ k \geq k'$: Number of motion features

$n \in \mathbb{Z}^+$: Width of the pixel array

$o \in \mathbb{Z}^+$: Height of the pixel array

$n' \in \mathbb{Z}^+,\ n' \geq n$: Width of the scene

$o' \in \mathbb{Z}^+,\ o' \geq o$: Height of the scene

Indexes

$t \in \mathbb{Z}^+$: Discrete time index

$p = [p_1, p_2]^T \in \mathbb{Z}^2$: Index of a point in a spatial grid, in retinal coordinates. If $p_1 \in [1, n]$ and $p_2 \in [1, o]$, then p is the index of a point within the camera's field of view.

Random Variables

$U_t \in \mathbb{R}^{k'}$: Control signal.

$Y_t \in \mathbb{R}^{n \times o}$: Camera image.

$L_t \in \mathbb{R}^{n' \times o'}$: Scene appearance.

$M_t \in \mathbb{Z}^2$: Pixel displacement induced by a camera movement.

$A_t \in \mathbb{R}^{2 \times k}$: Actuation parameters.

$\tilde{Y}_t, \tilde{L}_t, \tilde{M}_t, \tilde{A}_t$: Zero mean, independently distributed Gaussian noise sources.

Parameters

$\mu_{L_0} \in \mathbb{R} = \mathrm{E}[L_{0,p}]$ for all p.

$\sigma^2_{L_0} \in \mathbb{R} = \mathrm{Var}[L_{0,p}]$ for all p.

$\mu_{A_0} \in \mathbb{R}^{2 \times k} = \mathrm{E}[A_0]$.

$\Sigma_{A_0} \in \mathbb{R}^{2k \times 2k} = \mathrm{Cov}[A_0]$.

$\sigma^2_L \in \mathbb{R} = \mathrm{Var}[\tilde{L}_{t,p}]$ for $t > 0$ and all p.

$\sigma^2_Y \in \mathbb{R} = \mathrm{Var}[\tilde{Y}_{t,p}]$ for $t > 0$ and $p \in [1, n] \times [1, o]$.

$\Sigma_A \in \mathbb{R}^{2k \times 2k} = \mathrm{Cov}[\tilde{A}_t]$ for $t > 0$.

$\Sigma_M \in \mathbb{R}^{2 \times 2} = \mathrm{Cov}[\tilde{M}_t]$ for $t > 0$.

Functions

$\phi : \mathbb{R}^{k'} \to \mathbb{R}^k$: A feature function that extracts relevant information from the control signal U_t.

$c : \mathbb{R}^k \to \mathbb{R}^{2\times 2k}$: A function that takes a vector $A \in \mathbb{R}^k$ and outputs a block diagonal matrix with blocks A^T.

Vec : $\mathbb{R}^{2\times k} \to \mathbb{R}^{2k}$: A function that vectorizes a matrix, in row-major order.

Estimates

$m^*_{1:t}$: Conditional MAP estimate of $M_{1:t}$ given $y_{1:t}$, $u_{1:t}$, and $m^*_{1:t-1}$.
$\bar{\mu}_{A_t}$, $\bar{\Sigma}_{A_t}$: Kalman filter mean and variance for A_t given $u_{1:t}$ and $m^*_{1:t}$.
$\bar{\mu}_{L_{t,p}}$, $\bar{\sigma}^2_{L_{t,p}}$: Kalman filter mean and variance for $L_{t,p}$ given $y_{1:t}$ and $m^*_{1:t}$.

6.8.2 Generative Model

Summary

The Learning to Look model is governed by these equations:

$$L_{0,p} \sim \text{Normal}(\mu_{L_0}, \sigma^2_{L_0}) \text{ for all } p \tag{6.16}$$

$$A_0 \sim \text{Normal}(\mu_{A_0}, \Sigma_{A_0}) \tag{6.17}$$

$$\tilde{L}_{t,p} \sim \text{Normal}(0, \sigma^2_L) \text{ for all } p \tag{6.18}$$

$$\tilde{Y}_{t,p} \sim \begin{cases} \text{Normal}(0, \sigma^2_Y) & , \ p \in [1:n] \times [1:o] \\ \text{Normal}(0, \sigma^2_H) & , \ p \notin [1:n] \times [1:o], \ \sigma^2_H >> \sigma^2_Y \end{cases} \tag{6.19}$$

$$\tilde{M}_t \sim \text{Normal}(0, \Sigma_M) \tag{6.20}$$

$$\tilde{A}_t \sim \text{Normal}(0, \Sigma_A) \tag{6.21}$$

$$Y_{t,p} = L_{t,p} + \tilde{Y}_{t,p} \tag{6.22}$$

$$L_{t,p} = L_{t-1,p-M_t} + \tilde{L}_{t,p} \tag{6.23}$$

$$M_t = A_t \phi(U_t) + \tilde{M}_t \tag{6.24}$$

$$A_t = A_{t-1} + \tilde{A}_t \tag{6.25}$$

Description

In the convention of this thesis, X_t, U_t, and Y_t are random variables representing state, control, and observation respectively, and t is a discrete time index.

Within a discrete time slice, the following steps occur in order. (1) A motor command U_t is sent to the motors. (2) The motors start moving. (3) The motors stop moving at a motion offset M_t relative to where they started. (4) An image Y_t is collected. The goal of learning to look is to learn the mapping from U_t to M_t, which is parameterized by A_t. This is challenging because only Y_t and U_t are observable; M_t is not.

Control signals are given by $U_t \in \mathbb{R}^{k'}$, where k' is the number of motors. The control signal produces a displacement of the motor relative to its current position. *E.g.*, a small positive signal, like 0.01, may cause a small rightward displacement, and a large negative signal, like -0.1, may cause a large leftward displacement.

There are $n \times o$ discrete sensors arranged on a two-dimensional grid. The random variable $Y_{t,p}$ represents the observed value taken by pixel the at grid location $p \in \mathbb{Z}^2$ at time t. Values of $p \notin [1, n] \times [1, o]$ are outside the camera's current field of view. We simulate this limited field of view by thinking of pixels outside the field of view as uninformative, *i.e.* having very large white noise added.

The state X_t is a three-tuple, $\{L_t, M_t, A_t\}$, representing scene appearance, camera motion (displacement), and parameters of actuation respectively. The goal of "learning to look" is to estimate A_t.

The current scene appearance is given by $L_t \in \mathbb{R}^{n' \times o'}$, where $n' \geq n$ and $o' \geq o$ are the width and height of the scene. Thus, there are $n' \times o'$ discrete locations in the scene arranged on a two-dimensional grid. Scene locations are indexed in retinal coordinates: $L_{t,p}$ is the true light intensity emitted by the scene that renders $Y_{t,p}$ with independently distributed zero mean Gaussian noise $\tilde{Y}_{t,p}$. For $p \in [1, n] \times [1, o]$, $\text{Var}[\tilde{Y}_{t,p}] = \sigma_Y^2$. For $p \notin [1, n] \times [1, o]$, $\text{Var}[\tilde{Y}_{t,p}] = \sigma_H^2 >> \sigma_Y^2$, *i.e.* the observations are uninformative. As in Equation (6.22),

$$Y_{t,p} = L_{t,p} + \tilde{Y}_{t,p}$$

Applying a control signal U_t to the motors induces a remapping of scene coordinates according to Equation (6.23):

$$L_{t,p} = L_{t-1,p-M_t} + \tilde{L}_{t,p}$$

Let p' be the sensor index of a point in the scene before a motor command, and p the sensor index of the same point in the scene after a motor command. After a camera movement, the scene representation remaps: the mean of the new scene appearance $L_{t,p}$ at each location is taken from its previous value at its previous location, $p' = p - m_t$. Thus $\mathrm{E}[L_{t,p}] = L_{t-1,p-M_t}$, where $M_t \in \mathbb{R}^2$ gives the offset induced by camera motion. For example, consider a point in the scene located at camera pixel location $p' = [250, 250]^T$ prior to camera movement. After a camera movement, that pixel moves up and to the right by an offset of 100 pixels. Its new location is given by $p = [350, 150]^T = [250, 250]^T + [100, -100]^T$ where $m_t = [100, -100]^T$ is the horizontal and vertical pixel displacement caused by the camera movement, measured in a retinal coordinate system. The intensity of light coming from each location changes appearance over time with *i.i.d.* zero mean Gaussian noise $\tilde{L}_{t,p}$ with $\mathrm{Var}[\tilde{L}_{t,p}] = \sigma_L^2$ for all p.

M_t is the shift in camera position induced by a control signal U_t. It is determined by the actuation parameters $A_t \in \mathbb{R}^{2\times k}$. A feature function $\phi(U_t)$ extracts a length k vector of relevant information from U_t. Then, as in Equation (6.24),

$$M_t = A_t\phi(U_t) + \tilde{M}_t$$

where $\tilde{M}_t$ is *i.i.d.* zero mean Gaussian noise with $\mathrm{Cov}[\tilde{M}_t] = \Sigma_M$.[3] The motion parameters A_t may change over time with *i.i.d.* zero mean Gaussian noise $\tilde{A}_t$ with $\mathrm{Cov}[\tilde{A}_t] = \Sigma_A$, as in Equation (6.25):

$$A_t = A_{t-1} + \tilde{A}_t$$

For example, the gears may become looser, or rust and may become more resistant.

Distribution parameters

There are 8 total free parameters in our model. For all experiments in this paper, we fixed these eight parameters to constant values, which are given in Table 6.3.

[3]Some rounding operation is also required for converting from real values to discrete ones. In practice, the discretization is very fine, and we omit this operation from further analysis.

Table 6.3: Model Parameters & Implementation Values

	Scene Appearance Model		Motor Model	
	Parameter	Value	Parameter	Value
Prior Mean	$\mu_{0,L}$	0.5	$\mu_{0,A}$	$\vec{0}$
Prior Variance	$\sigma^2_{0,L}$	0.5^2	$\Sigma_{0,A}$	$500^2 I$
Dynamics Variance	σ^2_L	0.01^2	Σ_A	$5^2 I$
Output Variance	σ^2_Y	0.1^2	Σ_M	$20^2 I$

6.8.3 Inferring A_t

Vector representation of A_t

In order to estimate A_t with a Kalman filter, it is useful to vectorize it. We can do this by defining $c(\phi(U_t))$ to be a block diagonal matrix with blocks $\phi(U_t)^T$. Then we have

$$M_t = c(\phi(U_t))\text{Vec}(A_t) + \tilde{M}_t \tag{6.26}$$

For example, consider a robot with two actuators with outputs given by $U_{t,1}$ and $U_{t,2}$, and a feature vector $\phi(U_t)$ that simply adds a bias dimension the control signal. Under the original model, we have

$$\underbrace{\begin{bmatrix} M_{t,1} \\ M_{t,2} \end{bmatrix}}_{M_t} = \underbrace{\begin{bmatrix} A_{t,1} & A_{t,2} & A_{t,3} \\ A_{t,4} & A_{t,5} & A_{t,6} \end{bmatrix}}_{A_t} \underbrace{\begin{bmatrix} U_{t,1} \\ U_{t,2} \\ 1 \end{bmatrix}}_{\phi(U_t)} + \tilde{M} \tag{6.27}$$

where $A_{t,1:3}$ described how much the camera moves in the horizontal direction, and $A_{t,4:6}$ describe how much the camera moves in the vertical direction. This can be represented in a fashion suitable for Kalman filter estimation in the following identical way:

$$\underbrace{\begin{bmatrix} M_{t,h} \\ M_{t,v} \end{bmatrix}}_{M_t} = \underbrace{\begin{bmatrix} U_{t,1} & U_{t,2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & U_{t,1} & U_{t,2} & 1 \end{bmatrix}}_{c(\phi(U_t))} \underbrace{\begin{bmatrix} A_{t,1} \\ A_{t,2} \\ A_{t,3} \\ A_{t,4} \\ A_{t,5} \\ A_{t,6} \end{bmatrix}}_{\mathrm{Vec}(A_t)} \tag{6.28}$$

For legibility and intuitive understanding, throughout this chapter we write example A_t as matrices, and $\phi(U_t)$ as vectors. However, when it is necessary for correctness, we use the $c(\phi(U_t))$ matrix and $\mathrm{Vec}(A_t)$ vector in equations.

Inference

The goal of inference is to estimate the filtering distribution,

$$p(l_t, m_t, a_t \mid y_{1:t}, u_{1:t}) \tag{6.29}$$

Although each component of the model presented above is conditionally Gaussian, the filtering distribution is non-Gaussian, and difficult to estimate, due to the index remapping function $p = p' + m_t$.

In contrast, given a specific hypothetical trajectory $m'_{1:t}$, it is easy to compute the filtering distribution of l_t and a_t conditioned on $m'_{1:t}$:

$$p(l_t, a_t \mid m'_{1:t}, y_{1:t}, u_{1:t}) = p(a_t \mid m'_{1:t}, u_{1:t}) p(l_t \mid m'_{1:t}, y_{1:t}) \tag{6.30}$$

$$= \mathcal{N}(a_t, \bar{\mu}_{A_t}, \bar{\Sigma}_{A_t}) \prod_p \mathcal{N}(l_{t,p}, \bar{\mu}_{L_{t,p}}, \bar{\sigma}^2_{L_{t,p}}) \tag{6.31}$$

where $\bar{\mu}_{A_t}$ and $\bar{\Sigma}_{A_t}$ are the mean and variance of Kalman filter estimates of the posterior distribution of A_t given $m'_{1:t}$ and $u_{1:t}$, and $\bar{\mu}_{L_{t,p}}$ and $\bar{\sigma}^2_{L_{t,p}}$ are the means and variances of $n' \times o'$ separate Kalman filter estimates of the posterior distribution of the appearance $L_{t,p}$ of each scene location given $y_{1:t}$ and $m'_{1:t}$.

In such a situation, a Rao-Blackwellized particle filter (RBPF) might be used to sample trajectories of proposed $m'_{1:t}$ according to the posterior distribution

$p(m_{1:t} \mid y_{1:t}, u_{1:t})$, while maintaining Kalman filter estimates for $p(a_t \mid m'_{1:t}, u_{1:t})$ and $p(l_t \mid m'_{1:t}, y_{1:t})$ for each sampled $m'_{1:t}$ trajectory.

We take a simpler approach, and restrict the particle filter to a single particle containing the single trajectory $m^*_{1:t}$. At every time t, m^*_t is the maximum conditional *a posteriori* estimate given the previous estimated trajectory $m^*_{1:t-1}$, *i.e.* $m^*_t = \text{argmax}_{m_t} p(m_t \mid m^*_{1:t-1}, y_{1:t}, u_{1:t})$, where

$$p(m_t \mid m^*_{1:t-1}, y_{1:t}, u_{1:t}) = \frac{p(m_t \mid m^*_{1:t-1}, u_{1:t}) p(y_{1:t} \mid m^*_{1:t-1}, u_{1:t})}{p(y_{1:t} \mid u_{1:t-1}, m^*_{1:t-1})} \tag{6.32}$$

$$= p(m_t \mid m^*_{1:t-1}, u_{1:t})\ p(y_t \mid y_{1:t-1}, m^*_{1:t-1})\ \frac{p(y_{1:t-1} \mid m^*_{1:t-1})}{p(y_{1:t} \mid m^*_{1:t-1})} \tag{6.33}$$

$$= \mathcal{Z}\ \mathcal{N}(m_t,\ c(\phi(u_t))\bar{\mu}_{A_{t-1}},\ c(\phi(u_t))\bar{\Sigma}_{A_{t-1}} c(\phi(u_t))^T + \Sigma_A)$$
$$\prod_p \mathcal{N}(y_{p,t},\ \bar{\mu}_{L_{t-1,p-m_t}},\ \bar{\sigma}^2_{L_{t-1,p-m_t}} + \sigma^2_Y) \tag{6.34}$$

where $\mathcal{Z}$ is a constant with respect to m_t, and can be ignored in finding the argmax. Rewriting this as a log function and ignoring terms that don't depend on m_t, which preserves the maximum, gives a function $g(m_t)$ with three terms:

$$g(m_t) =$$
$$-\overbrace{(m_t - c(\phi(u_t))\bar{\mu}_{A_{t-1}})^T (c(\phi(u_t))\bar{\Sigma}_{A_{t-1}} c(\phi(u_t))^T + \Sigma_A)^{-1} (m_t - c(\phi(u_t))\bar{\mu}_{A_{t-1}})}^{Predicted\ Motion\ Match}$$
$$- \sum_{p \in [1,n] \times [1,o]} \underbrace{\frac{(y_{t,p} - \bar{\mu}_{L_{t-1,p-m_t}})^2}{(\bar{\sigma}^2_{L_{t-1,p-m_t}} + \sigma^2_L)}}_{Image\ Match} - \sum_{p \in [1,n] \times [1,o]} \underbrace{\log(\bar{\sigma}^2_{L_{t-1,p-m_t}} + \sigma^2_L)}_{Uncertainty\ Penalty} \tag{6.35}$$

For conciseness, we define:

$$\hat{\mu}_{A_t} \stackrel{\text{def}}{=} \bar{\mu}_{A_{t-1}} \tag{6.36}$$

$$\hat{\Sigma}_{A_t} \stackrel{\text{def}}{=} (c(\phi(u_t))\bar{\Sigma}_{A_{t-1}} c(\phi(u_t))^T + \Sigma_A) \tag{6.37}$$

$$\hat{\mu}_{L_{t,p}} \stackrel{\text{def}}{=} \bar{\mu}_{L_{t-1,p}} \tag{6.38}$$

$$\hat{\sigma}^2_{L_t,p} \stackrel{\text{def}}{=} \sigma^2_{L_{t-1},p} + \sigma^2_L \tag{6.39}$$

and then can rewrite the predicted motion match in an equivalent way:

$$g(m_t) = -\overbrace{\left[m_t - \hat{\mu}_{A_t}\phi(u_t)\right]^T \hat{\Sigma}_{A_t}^{-1} \left[m_t - \hat{\mu}_{A_t}\phi(u_t)\right]}^{Predicted\ Motion\ Match} - \sum_{p\in[1,n]\times[1,o]} \underbrace{\frac{(y_{t,p} - \hat{\mu}_{L_{t,p-m_t}})^2}{\hat{\sigma}^2_{L_{t,p-m_t}}}}_{Image\ Match} - \sum_{p\in[1,n]\times[1,o]} \underbrace{\log(\hat{\sigma}^2_{L_{t,p-m_t}})}_{Uncertainty\ Penalty} \tag{6.40}$$

Note that $\bar{\Sigma}_{A_t}^{-1}$ is function of $\bar{\Sigma}_{A_{t-1}}$ and u_t, but not m_t.

6.8.4 Extended model 1: motor dynamics

The first extended Learning to Look model considers the temporal dynamics of camera movement. Previously, the following steps occurred, in order, within a discrete time slice. (1) A motor command U_t is sent to the motors. (2) The motors start moving. (3) The motors stop moving at a motion offset M_t relative to where they started. (4) An image Y_t is collected.

The first expanded learning to look model attempts to unpack this sequence. To this end, we add a more fine grained timescale, indexed by r. Now, within a discrete time slice t, the following steps occur. (1) A motor command U_t is sent to the motors. (2) The motors start moving. (3) ρ images $Y_{r,t}$ are collected in sequence. At time r, the motors are still moving, and there is a cumulative motion offset $M_{r,t}$ relative to where they started. (4) The motors stop moving at a motion offset M_t relative to where they started. (5) An image Y_t is collected.

Additional Indexes

$r \in \mathbb{Z}^+, r \leq \rho$: A time index smaller than t. We define new random variables at this new timescale, *e.g.* $Y_{r,t}$. Note that Y_t is not a collection of $Y_{1:\rho,t}$, but is a separate random variable. In practice we take $Y_t = Y_{\rho,t}$.

Additional Random Variables

$Y_{r,t} \in \mathbb{R}^{n\times o}$: Camera image on frame r.

$M_{r,t} \in \mathbb{Z}^2$: Cumulative motion offset on frame r relative to the camera position at frame 0.

$A_{r,t} \in \mathbb{R}^{2\times k}$: Motion parameters.

$\tilde{Y}_{r,t},\ \tilde{M}_{r,t},\ \tilde{A}_{r,t},\ \tilde{L}_{r,t}$: Sources of zero mean Gaussian noise.

Additional Parameters

$\rho \in \mathbb{N}$: The number of r timesteps required to finish a camera movement.

6.8.5 Temporal dynamics of eye movement

Summary

The "motor dynamics" extension to the Learning to Look model is governed by these additional equations:

$$A_{r,0} \sim \text{Normal}(\mu_{A_0}, \Sigma_{A_0}) \tag{6.41}$$

$$\tilde{L}_{r,t,p} \sim \text{Normal}(0, \sigma_L^2) \text{ for all } p \tag{6.42}$$

$$\tilde{Y}_{r,t,p} \sim \begin{cases} \text{Normal}(0, \sigma_Y^2) & ,\ p \in [1:n] \times [1:o] \\ \text{Normal}(0, \sigma_H^2) & ,\ p \notin [1:n] \times [1:o],\ \sigma_H^2 >> \sigma_Y^2 \end{cases} \tag{6.43}$$

$$\tilde{M}_{r,t} \sim \text{Normal}(0, \Sigma_M) \tag{6.44}$$

$$\tilde{A}_{r,t} \sim \text{Normal}(0, \Sigma_A) \tag{6.45}$$

$$Y_{r,t,p} = L_{t-1,p-M_{r,t}} + \tilde{Y}_{r,t,p} + \tilde{L}_{r,t,p} \tag{6.46}$$

$$M_{r,t} = A_{r,t}\phi(U_t) + \tilde{M}_{r,t} \tag{6.47}$$

$$A_{r,t} = A_{r,t-1} + \tilde{A}_{r,t} \tag{6.48}$$

Description

We consider a simplified model of camera movement dynamics. Under this model, all camera movements require ρ time to complete, regardless of signal magnitude. Movement unfolds in discrete time steps r, which are smaller than t. Thus, at each point in time, the pixel offset from the camera position at frame

$r = 0$, $M_{r,t}$, is given by

$$M_{r,t} = A_{r,t}\phi(u_t) + \tilde{M}_{r,t} \tag{6.49}$$

On each frame r, the camera observes an image $y_{r,t}$. Since r is smaller than t, observations are generated from the previous scene, L_{t-1}, with additional uncertainty. Thus, observations are generated by

$$Y_{r,t,p} = L_{t-1,p-M_{r,t}} + \tilde{Y}_{r,t} + \tilde{L}_{r,t} \tag{6.50}$$

An advantage of this model of temporal dynamics is that the same inference procedure used to infer M_t can be used to infer $M_{r,t}$, and thus $A_{r,t}$. In practice, we take $M_t = M_{\rho,t}$, $Y_t = Y_{\rho,t}$, and $A_t = A_{\rho,t}$.

6.8.6 Extended model 2: signal dependent noise

The second extended Learning to Look model accounts for the reliability of each degree of freedom. The goal is to model signal dependent motor noise, where the standard deviation scales linearly with the control signal.

Additional Random Variables

$N_t \in \mathbb{R}^{+2\times k}$: Motor noise (standard deviation)
$\tilde{W}_t \in \mathbb{R}^{2\times 2}$: Spherical Gaussian white noise with unit standard deviation.

Additional Parameters

$\mu_{N_0} \in \mathbb{R}^{+2k}$: Motor noise prior mean.
$\Sigma_{N_0} \in \mathbb{R}^{+2k\times 2k}$: Motor noise prior variance.
$\Sigma_N \in \mathbb{R}^{+2k\times 2k}$: Motor noise drift, variance.

6.8.7 Control: coordinating multiple gaze components

Summary

The "signal dependent noise" extension to the Learning to Look model is governed by these additional equations:

$$N_0 \sim \text{Normal}(\mu_{N_0}, \Sigma_{N_0}) \tag{6.51}$$

$$\tilde{N}_t \sim \text{Normal}(0, \Sigma_N) \tag{6.52}$$

$$\tilde{W}_t \sim \text{Normal}(0, I) \tag{6.53}$$

$$N_t = N_{t-1} + \tilde{N}_t \tag{6.54}$$

In addition, Equation (6.24) is amended to read

$$M_t = A_t U_t + N_t U_t \tilde{W}_t \tag{6.55}$$

Description

A robot must discover how to coordinate multiple gaze components to make a desired camera movement $m_t^\star$. In this section we appeal to the principle of maximum accuracy in the presence of signal dependent noise. In humans, the accuracy of eye movements decreases with target distance, with horizontal standard deviation increasing at a rate of 0.145 degrees per target degree [141].

We start by assuming that this signal dependent noise relation holds true for all gaze components. Previously, we had modeled

$$M_t = A_t \phi(U_t) + \tilde{A}_t \tag{6.56}$$

Now we work exclusively with an identity feature function $\phi(U_t) = U_t$. When we consider noise that has a standard deviation that scales linearly with the input signal U_t, this becomes

$$M_t = A_t U_t + N_t U_t \tilde{W}_t \tag{6.57}$$

where N_t is a random matrix of the same shape as A_t, and $\tilde{W}_t$ is zero mean, unit variance, spherical white noise. The elements of N_t describe the contribution of

each motor to horizontal and vertical standard deviation. This gives

$$p(m_t \mid a_t, u_t, n_t) = \mathcal{N}(m_t, a_t u_t, \mathrm{diag}(u_t n_t)^2) \tag{6.58}$$

The goal is to find the action $u_t^\star$ that results in the motion offset m_t with minimum squared Euclidean distance to some desired motion offset $m_t^\star$. For fixed and given a_t, n_t, this cost function can be written as

$$\begin{aligned}
\mathrm{cost}(u_t) &= \int (m_t^\star - m_t)^T (m_t^\star - m_t) p(m_t \mid a_t, u_t, n_t) dm_t && (6.59)\\
&= \int (m_t^\star - m_t)^T (m_t^\star - m_t)\, \mathcal{N}(m_t, a_t u_t, \mathrm{diag}(n_t u_t)^2) dm_t && (6.60)\\
&= m_t^{\star T} m_t^\star - 2 m_t^{\star T} a_t u_t + \mathrm{Tr}(\mathrm{diag}(n_t u_t)^2)) + (a_t u_t)^T a_t u_t && (6.61)\\
&= m_t^{\star T} m_t^\star - 2(a_t^T m_t^\star)^T u_t + 1^T (n_t \circ n_t)(u_t \circ u_t) + u_t^T a_t^T a_t u_t && (6.62)\\
&= m_t^{\star T} m_t^\star - 2(a_t^T m_t^\star)^T u_t + u_t^T \,\mathrm{diag}(1^T (n_t \circ n_t)) u_t + u_t^T a_t^T a_t u_t && (6.63)
\end{aligned}$$

where $A \circ B$ denotes the Hadarmard (elementwise) product between matrix A and matrix B [142], and 1^T is a row of 1s. Differentiating with respect to u_t gives

$$\begin{aligned}
\frac{\partial\, \mathrm{cost}(u_t)}{\partial u_t} &= -2 a_t^T m_t^\star + 2\,\mathrm{diag}(1^T (n_t \circ n_t)) u_t + 2 a_t^T a_t u_t && (6.64)\\
&= -2 a_t^T m_t^\star + 2\left[\mathrm{diag}(1^T (n_t \circ n_t)) + a_t^T a_t\right] u_t && (6.65)
\end{aligned}$$

Setting the derivative to 0 and solving for $u_t^\star$ gives

$$\begin{aligned}
a_t m_t^\star &= \left[\mathrm{diag}(1^T (n_t \circ n_t)) + a_t^T a_t\right] u_t^\star && (6.66)\\
u_t^\star &= \left[\mathrm{diag}(1^T (n_t \circ n_t)) + a_t^T a_t\right]^{-1} a_t^T m_t^\star && (6.67)
\end{aligned}$$

Note that $1^T (n_t \circ n_t)$ is a vector whose elements are the sum of the variances across each direction of gaze for each gaze component.

The optimal gaze coordination rule given in Equation (6.67) has several intuitively appealing properties. First, when the variances are 0 (the noiseless condition), the optimal gaze rule is a pseudo-inverse. Second, as the overall variance increases, the value of the denominator decreases, yielding an undershoot in the presence of signal dependent noise. Third, the contribution of each motor is down weighted by its variance; thus, in the presence of multiple degrees of freedom, each

with different reliabilities, the optimal gaze rule will weight more heavily the more reliable degrees of freedom. If head movement is less reliable than eye movement, then the eyes will contribute more to the shift in gaze direction than the head.

Acknowledgment

The text of Chapter 6, with some modification, is a reprint of the material as it appears in N.J. Butko and J.R. Movellan, "Learning to Look," *Proceedings of the 2010 IEEE International Conference on Development and Learning*, 70–75 (2010) [7]. I was the primary author of this publication; the co-author supervised the research that forms the basis of this chapter.

Chapter 7

Learning about Humans During the First 6 Minutes of Life

7.1 Abstract

We report results of an experiment with a baby robot that learned to discover visual categories using a simple acoustic contingency detector as its training signal. With less than 6 minutes of experience sampled from 90 minutes of interaction with the world, the robot learned to find people in novel images. In addition, it developed a preference for drawings of human faces over drawings of non-faces, even though it had never been exposed to such schematic face drawings before. During the 6 minutes of training, the baby robot was never told whether or not people were present in the images, or whether people were of any particular relevance at all. It simply discovered that to make sense of the images and sounds it received, it was a good idea to use feature detectors that happen to discriminate the presence of people. The results illustrate that visual preferences of the type typically investigated in human neonates can be acquired very quickly, in a matter of minutes. Previous studies that were thought to provide evidence for innate cognitive modules, may actually be evidence for rapid learning mechanisms in a neonate brain exquisitely tuned to detect the statistical structure of the world.

7.2 The Rapid Learning Hypothesis

There is strong experimental evidence that newborn infants tend to orient towards human faces. Opinions are divided, however, about the causes for this early preference. Kleiner and Banks [143] divided the prevalent views into *the social hypothesis*, which contends that infants are predisposed to attend visually to conspecifics (other humans), and *the sensory hypothesis*, which offers that infants attend to basic informative cues (contrast, motion, *etc.*) that happen to compose faces. Neither hypothesis questions that the face preference exhibited by infants is present at birth. Morton and Johnson [62] published an influential study showing that 40-minute-old neonates responded preferentially to face drawings over other controlled stimuli, replicating an earlier result [144]. This helped establish the now prevalent view that infants are born with an innate mental architecture for handling visual "species knowledge".

Johnson vigorously argued against a third hypothesis, *the rapid learning hypothesis*, and so this hypothesis has remained untested. According to this hypothesis the human brain may be endowed with fast, general purpose learning mechanisms to efficiently encode sensorimotor signals. These mechanisms may be responsible for the visual preferences found in human neonates. Johnson claimed that it would be all but impossible for such learning to occur within the first 40 minutes of life, specifically targeting the notion that the neonate brain could generalize from natural 3-dimensional scenes encountered just after birth to the abstract 2-dimensional face schematics used in infant experiments.

There is evidence for evidence for rapid learning in infants. In particular, Bushnell *et al*. report that 2 day old infants fixate longer to images of their mothers than to images of other women with similar hair colors and facial complexion [68]. This, combined with recent advances in the field of machine perception, has caused us to revisit the rapid learning hypothesis.

Multiple machine perception researchers have pursued the idea of combining multiple, low-level cues in one modality to bootstrap learning in another modality. For example Hershey & Movellan [145] showed that it is possible to locate faces

by focusing on regions of the image plane that correlate highly with the acoustic signal. Beal *et al*. [146] developed a probabilistic generative model under which a common cause generates both auditory and visual data, and used this model to infer templates of human appearance from video without supervision. Triesch & von der Malsburg [147] used multiple low-level visual features for unsupervised person tracking in video, and de Sa [148] developed an unsupervised learning method in which audio and video systems trained each other to classify spoken syllables. Finally, Blum & Mitchell [149] used a similar technique for classifying web pages, in which the links to web pages and the words in the page are treated as separate modalities.

Cohen & Cashon have suggested that the combination of multiple, low-level cues may also be a basic mechanism used in infant learning [150]. Rather than innate visual biases, simple low-level cues such as auditory, tactile, or proprioceptive input may be used as evidence for the presence or absence of objects. John Watson proposed that the infant brain is particularly sensitive to the presence of contingencies between sensory channels, and that this contingency drives the definition and recognition of caregivers. Based on experiments in which 2-month-old infants displayed social responses to non-human contingent agents (mobiles rigged to respond to head movements), he has hypothesized that human faces become attractive because they tend to occur in high contingency situations. [11]. Recent neuroscience work shows that the short-latency dopamine system is involved in the perception of novel contingencies [151], adding credence to Watson's speculation that a contingency modality may be present at birth [152].

While John Watson's contingency detection hypothesis has been influential in the literature, it is unclear whether or not it is computationally plausible. Is it really possible to learn about faces using sensorimotor contingency as the only training signal? How much exposure to the world would be required to develop such preferences? Is there something special about contingency that is required for identifying caregivers?

In this chapter, we present a study aimed at clarifying these questions from a computational point of view. To this effect we developed a simple robot

in the shape of a baby doll and endowed it with machine perception primatives recently developed at our laboratory. The baby robot interacted with members of our laboratory while recording images of the world it saw. During this time, the real-time contingency detection algorithm that we developed in Chapter 4 analyzed the sound signal for the presence of contingencies in the auditory domain. When auditory contingencies were detected, the image was automatically sent to a learning module with the label *contingency present*. When not, the image was sent with the label *contingency absent*. The robot's task was to discover what made these two categories of images different.

This is a challenging machine learning task due to the fact that the training signal in this case is rather weak: (1) The contingency detector did not indicate where the object causing contingencies was located on the image plane. It just indicated whether or not a contingency was detected in the auditory domain. (2) The information provided by the contingency detector was contaminated by errors. For example, sometimes contingencies were detected in the auditory domain while a person was not visible. Other times a person was visible but auditory contingencies were either not present, or not found by the contingency detector.

Our goal was to explore whether contingency information would be sufficient for the robot to develop preferences for human faces, and to get a sense for the time scale of the learning problem (would it require months to establish such preferences?), and test whether those preferences would transfer to abstract stimuli, like 2-D drawings. Contrary to previous assumptions, we found our robot was able to rapidly learn face preferences, relying on only six minutes of visual experience. Furthermore, it learns to identify and locate people in the visual scene reliably, even when their faces are not present.

7.3 Infant robot

In order to make claims about the visual information that infants can realistically be expected to gather and learn from, it was important to collect data from a baby's eye view (BEV). To this end we built a simple interactive baby

Figure 7.1: The baby robot, Beverly. Two types of beginning experimental conditions, "stroller" and "crib", are shown (left and middle respectively). The robot infant did not remain in a constant position as subjects were allowed to pick it up if they liked (right).

robot named Beverly. Beverly was a plush baby doll two sensors and one actuator. The first sensor was an IEEE1394a webcam located in the forehead. The second sensor was a microphone. The actuator was a small speaker located inside the chest (Figure 7.1).

Beverly's microphone and speaker formed a closed-loop system controlled by a computer which ran a social contingency detection algorithm developed developed in Chapter 4, and consists of (a) an *infomax controller* which schedules vocalizations so as to maximize the information gained about the presence or absence of contingencies, and (b) a Bayesian inference algorithm that computes the probability that a contingency is present given the observed sequences of auditory signals.

Based on this controller, Beverly makes vocalizations and listens to the environment to determine as quickly as possible if a contingent agent is present. The continuous audio input was converted to binary auditory "events" by thresholding the instantaneous power from the microphone. Whenever an auditory event occurred and the posterior probability of social contingency given by the contingency detector was simultaneously above 97.5%, an image was saved with the label

"contingent". Whenever an auditory event occurred and the posterior probability of social contingency was simultaneously below 2.5%, an image was saved with the label "not contingent".

The camera sensor was in an open-loop, and merely collected visual data without using that visual data to affect Beverly's behavior. These data were then processed offline using segmental Boltzmann fields (SBFs), a class of weakly-supervised learning architectures. The weakly supervised learning problem of discovering object categories from images which are only labeled as "containing," or "not containing," objects of interest has recently seen tremendous progress [153–158]. Fasel [159] developed SBFs, new learning architecture that achieved the best reported detection rates on Caltech-6, a popular collection of image datasets for automatic discovery of object categories [153]. SBFs can be seen as an instance of convolutional neural-nets and are also unique among weakly supervised architectures in their capacity to localize objects of interest in real time at video frame rate.

Learning with SBFs only requires that the images are weakly labeled as containing with high or low probability the object of interest, without the need to indicate where the objects are located on the image plane. In our case, Beverly's "objects of interest" are ones that cause the real time contingency detection engine to register auditory contingencies based on acoustic signals.

7.4 Data Collection

We allowed Beverly to act and observe in her environment for a total of 88 minutes across two sessions. During this time, she made vocalizations and observed images, which she associated with perceptions arising from her contingency detector. Based on the rule for collecting images in Section 7.3, 3701 images were collected. Of these, Beverly's auditory contingency detection engine believed 2877 scenes to be associated with auditory contingencies, while 824 scenes were believed to be void of contingent interaction.

During the 88 minutes of the experiment Beverly was placed in three dif-

ferent conditions: a *chair* condition, a *stroller* condition, and a *crib* condition. For each condition Beverley was moved so as to face initially one of three different backgrounds. Each condition was presented in one of two lighting conditions *bright* and *dim*. This provided 18 different background conditions (see Figure 7.1 for two example starting conditions). Within each background condition subjects could move Beverly and handle her freely, so constant backgrounds could not be assumed.

Nine members of the Machine Perception Laboratory at UCSD were asked to interact with Beverly and instructed to try to make her "excited". They were told that she would make excited noises if she thought somebody was responding to her, and would make bored noises otherwise. Beverly's vocal repertoire comprised 5 different baby sounds, ranked in level of excitation by the experimenters, each corresponding to a different level of the posterior probability of contingency as estimated by the contingency detector.

The nine subjects interacted with Beverly for two trials of two minutes each; each of the 18 trials, comprising 36 minutes of total interaction, began in a different background condition chosen randomly. Beverly acted autonomously and continuously during the 88 minutes of the experiment, with one break in the middle. This continuous action time included 52 minutes in which Beverly was being moved to different starting conditions and as subjects entered and left the room and were given instructions. The experimental room was noisy due to a computer cluster in the same room, the background conversations from adjacent offices, and during initial instruction periods between the subjects and the experimenter.

7.5 Visual Learning and Performance

Of the data that were collected by Beverly's visual system, a small fraction of them were used to train SBFs as described in [159]. In all, a total of 185 SBFs were trained. The only difference between any two SBFs was the set of images used in training. Each different SBF can be seen as the mental representation that Beverly would form of her visual world given a different set of experiences. Thus

each SBF can be thought of as a different baby. SBFs that have more experience (use more examples for training) can be thought of as the visual representations of older babies than SBFs that use fewer.

In all, 31 infants' SBF representations each were trained using 2, 4, 8, 16, or 32 examples of what Beverly's contingency detection engine believed were contingent images, based on auditory contingencies. In all cases, the number of negative example images was 6 times the number of positive example images. Thus the minimum number of examples in training was 14, and the maximum was 224. The remainder of the images (3477–3687) formed a test set that were never used to train a given SBF, but were used to evaluate its performance.

A model baby's SBF learns an accurate representation if people were found to be the cause of auditory contingencies. This would be indicated by high likelihood being assigned to images containing people. Each SBF assigned likelihood values to all images in its test set based on how well it matches the images used in training that were associated with auditory contingencies.

Several 2-Alternative Forced Choice (2AFC) tasks were performed based on these likelihood values.[1] In each task, the test set was split into two groups of images. Three tasks used the full test set: "contingent vs. non-contingent," "person vs. no-person," "face vs. no face." One task, "face vs. no person," excluded images from the test set that contained people's bodies but not their faces. The "person" and "face" labels were coded by hand *post hoc* and were never used in training.

For the 31 SBFs that had the most experience, the 2AFC performance was: face vs. no face – 86.1%, contingent vs. non-contingent – 89.97%, person vs. no person – 94.3%, face vs. no person – 95.31%. Figure 7.2 details the effect of experience on performance, and indicates that this order was preserved across all levels of experience.

The performance order is quite interesting. When the task was "face vs. no face," the system was shown all possible pairings of face images with non-

[1] 2AFC performance is equivalent to area under the ROC [160], and is an indicative measure because 50% always indicates chance performance.

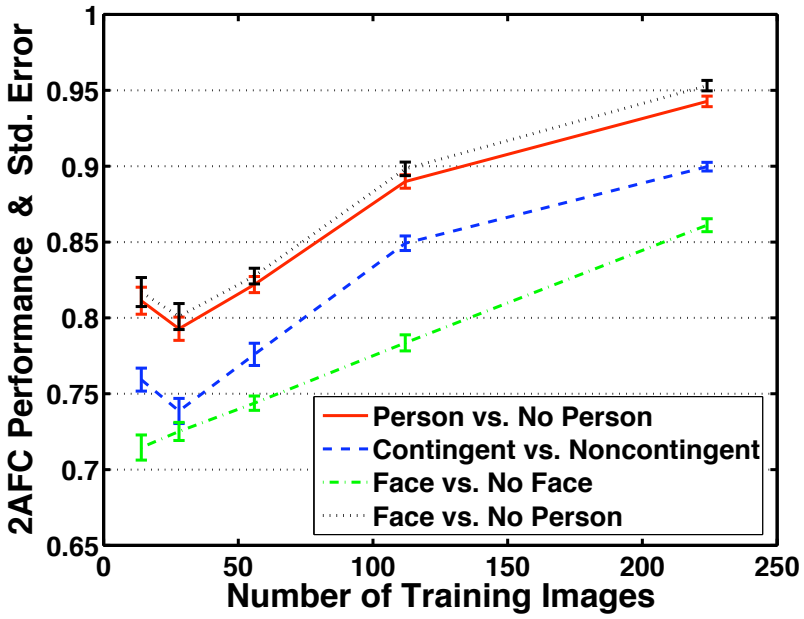


Figure 7.2: Which of two images was visually classified as a likely source or contingent experience? For example, with a few hundred training images (less than six-minutes into the experiment) Beverly reliably picks out visual regions with faces to be more likely causes of contingency than visual regions with no people.

face images; the intended negative choice could still have a faceless body, and it would be reasonable for the SBF to associate both images strongly with auditory contingencies. In such a case, it may be random which image was judged to have higher likelihood, and so the trial has a high chance of failing. This is reflected in the poor performance of the "face vs. no face" task. SBFs were trained with image labels provided by the contingency detector, which misrepresented the presence versus the absence of people by about 7.3% (Table 7.1), but after learning, Beverly only misjudged the presence of people by 5.7%: *i.e.*, by the end of rapid visual learning, Beverly's visual system was more reliable at detecting caregivers than her contingency perception was. The learner surpassed the teacher. She succeeded in identifying the causal-structure underlying her experiences.

The labels did not provide any information about *where* people were located

Table 7.1: Disagreement between contingency detector *vs*. human labels

Experimenter Label	Disagreement Type		
	Contingent but doesn't contain...	Not contingent but contains...	Total Disagreement...
"Face"	25.73% (212/824)	14.84% (427/2877)	17.27% (639/3701)
"Person"	16.99% (140/824)	4.48% (129/2877)	7.27% (269/3701)

in the image, and there were no constraints on the views and poses of people, which sometimes contained faces, sometimes only bodies, and had wide variability in orientation, scale, and lighting conditions. These things were all learned by Beverly's SBF visual representation. Some examples of Beverly's predictions are shown in Figure 7.3. Notice that both head and body tend to be identified as likely sources of contingent interaction.

Indeed, there was nothing special about the use of contingency to identify people, except to provide a label automatically and with some accuracy higher than chance. One can imagine low-level cues other than contingency with the potential to allow for the higher than chance definition of characters, such as "things that move me" or "things that I see moving", for example. Any such cue would be sufficient to provide an anchor for rapid learning.

7.6 Generalization to New Situations

One advantage of rapid learning is that it is necessarily appropriate to the environment that an infant lives in. That is, evolution does not have to guess whether you will be born in a bright or dim place, or what the distracting objects around you might be. However, the drawback of specific learning is that it is not clear whether the learned knowledge will transfer to novel situations in different environments.

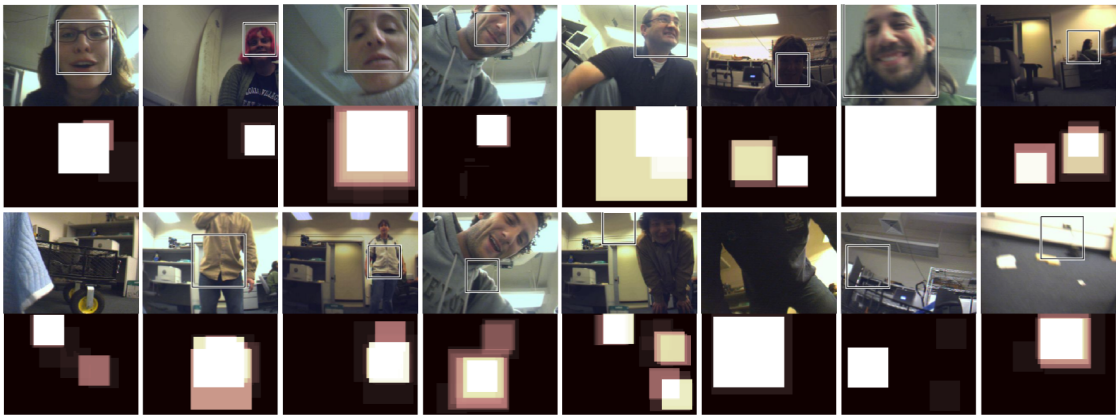


Figure 7.3: Beverly's visual experiences and her estimate of how likely each region of the image is to cause a contingent interaction. *Top*: Good localization and detection results. *Bottom*: (1) correct rejection, (2)-(4) correct detections, where the body was preferred over the face, (5) the most probable location was incorrect, however the image was correctly classified, (6) an incorrect rejection, (7)-(8) incorrect detections.

7.6.1 Real people

We tested the generality of Beverly's visual representations by assessing performance on a novel data set. The Caltech-6 data set [153] contains images with and without faces taken from around Caltech's campus. These images differ greatly from the visual experience of Beverly. Not only do they contain people that Beverly has never seen before, they contain a variety of backgrounds, lighting conditions, and facial expressions that Beverly is not familiar with (most people smile intensely to Beverly, but not in the Caltech database). Examples of these new images are shown in Figure 7.4A.

Nevertheless, when we tested Beverly on the Caltech-6 database using a two-alternative forced choice task, she achieved over 80% performance with just 8 positive contingency experiences. Interestingly, as she got more information from her own experiences, she started to perform *worse* on the more general task,

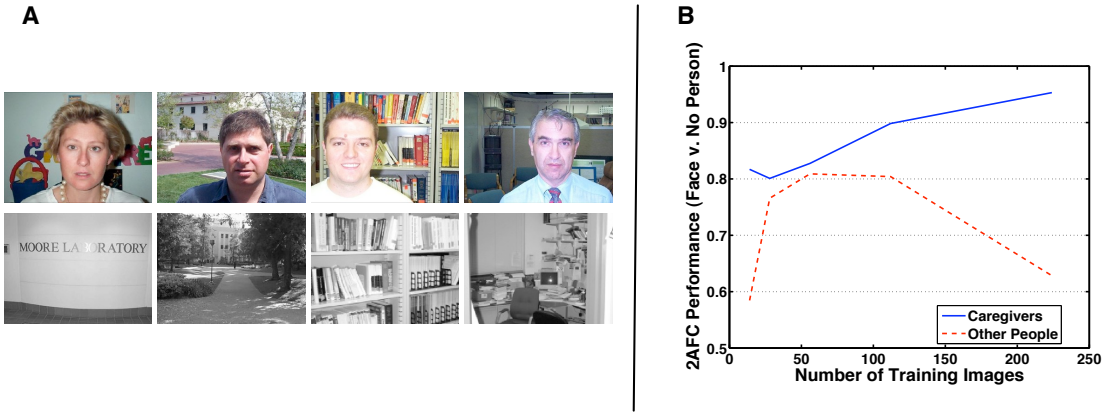


Figure 7.4: **A**: Typical faces and backgrounds from the Caltech-6 data set. **B**: Mean performance on familiar and unfamiliar people and places by SBFs that learned only on Beverly's experiences. Unfamiliar examples are drawn from the Caltech-6 data set. While performance continues to improve throughout learning on familiar examples, performance *decreases* on unfamiliar faces, which may reflect an infant's early learned preference for the mother's face.

indicating that Beverly starts to overspecialize in her environment. These results are illustrated in Figure 7.4B.

This may be consistent with the finding that infants quickly learn to prefer their mother's face over the faces of similar-looking women [68]. With more experience, Beverly learns more about the *specific* caregivers available to her, at the expense of more general visual knowledge of people. This hypothesis is strengthened by the observation that, while generalization performance begins to go down, performance on novel experiences from Beverly's own environment continues to rise.

7.6.2 Schematic face stimuli

Johnson *et al.* [161] presented 40-minute-old human neonates with 3 types of visual stimuli to study their visual preferences (See Figure 7.5): (a) A drawing of a frontal face; (b) A drawing with the same features of the face but scrambled arrangement while maintaining symmetry; (c) An empty face-outline. They found

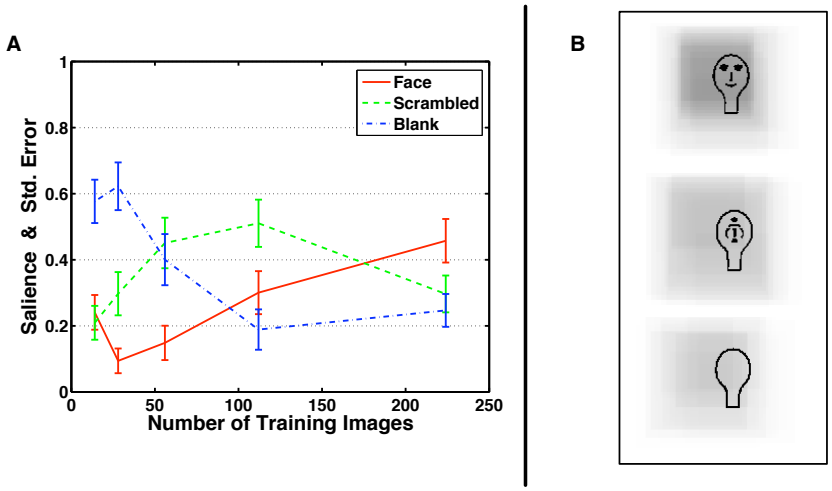


Figure 7.5: **A**: Average preference given to Johnson's Face Stimuli. By the end of learning with less than 6 minutes of data, Beverly shows the same preference for faces over scrambled stimuli and for scrambled over blank that Johnson observed in neonate human infants. **B**: Average salience of schematic stimuli. The salience (grey) is overlaid on the original stimulus. Darker indicates "more salient," and so the salience order matches the ordering observed in infants.

that infants showed an order of tracking preference in favor the face stimulus, followed by the scrambled stimulus, followed by the empty stimulus.

Johnson explicitly expressed doubt as to whether it was possible to generalize visual experience of three-dimensional real-world objects observed within the first few minutes of life to two dimensional schematic line drawings. In response to this strong assertion of computational impossibility, we set out to investigate whether such generalization was indeed as difficult as Johnson believed.

We presented Beverly with the same three stimuli used in [161]. Recall that Beverly had been exposed only real visual scenes, along with an experience of social contingency. She had no prior notion of what faces meant, who conspecifics were, and had certainly never seen line drawings of faces.

Despite this, under certain parameter regimes, Beverly was reliably able to reproduce the average preference order reported in [161] perfectly. We assessed the preference of each SBF by presenting Johnson's three stimuli in distinct regions of a

single input stimulus, and then taking the maximum salience pixel in each of those three regions to be the preference for a given stimulus. These saliences were then averaged over all 31 SBFs trained at a given amount of learning experience. This assessment method was designed to give maximum analogy to Johnson's method of averaging head-angle-turn over all infants. The parameters manipulated were amount of training images used, and size of test-image.

After learning had completed with 32 positive examples and 192 negative examples, we found that on average, the area around the face drawing was given the highest probability of coming from the contingency category, the area around the scrambled face was given somewhat less probability, and the area around the empty face was given even less probability. These results are illustrated in Figure 7.5.

This shows that learning to generalize from visual experience of the real world to schematic cartoon drawings is not as computationally difficult as previously believed. While it is still an open question whether infants could learn quickly from the information available to them in the first few minutes of life, we have shown at least that enough information is present for such learning to be possible.

7.7 Developmental implications

From a sample of only 32 images labeled as "contingent" and 192 images labeled as "not contingent", the robot Beverly's visual system was capable of detecting the presence of people in novel images from her environment with high accuracy (over 94% correct on a 2AFC task). In doing so she developed a preference for human faces that was detectable in 2d-face line-drawings that she had never been exposed to. She was never told by a human whether or not people were present in the images, or whether people were of any particular relevance at all. However she discovered that the only consistent visual explanation for two sets of scenes with differing auditory response statistics was a combination of feature detectors that happened to discriminate the presence of people. While these feature detectors

initially began to discern the general presence of people in general environments, they eventually specialized in detecting the particular people in Beverly's particular environment.

There was nothing special about the auditory contingency domain – similar results could undoubtedly be obtained using other modalities as long as those cues indicate the presence of people at a higher than chance rate. The results illustrate that from a computational point of view, the visual preferences of the type typically investigated in human neonates can be acquired very quickly, in a matter of minutes. Previous studies that were thought to provide evidence for innate cognitive modules may actually be evidence for rapid learning mechanisms in a neonate brain exquisitely tuned to detect the statistical structure of the world. This further adds to a body of evidence that simple cues from one or several other modalities are sufficient to learn visual concepts without supervision (*e.g.*, [148, 162]).

The results show that rapid learning is a viable explanation for empirical results that had previously been thought to require innate "units of mental architecture." They provide computational credibility to John Watson's views about the role of contingency on infant development [11]. Most importantly the results illustrate the importance of understanding the problems faced by the developing brain via computational experiments with real-world images and sounds.

Acknowledgment

The text of Chapter 7 is unpublished work, to be submitted with authors N.J. Butko, I.R. Fasel, and J.R. Movellan. I was the primary author and researcher on this project, designing the experiments, obtaining results, and drafting the manuscript. Fasel developed the software systems used, and Movellan supervised the research that forms the basis for this chapter.

Bibliography

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

[2] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. Henry Holt and Co., Inc. New York, NY, USA, 1982.

[3] Nicholas J. Butko and Jochen Triesch. Exploring the role of intrinsic plasticity for the learning of sensory representations. *Neurocomputing*, 70(7–9):1130–1138, 2007.

[4] Nicholas J. Butko, Lingyun Zhang, Garrison W. Cottrell, and Javier R. Movellan. Visual saliency model for robot cameras. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 2398 –2403, May 2008.

[5] Nicholas J. Butko and Javier R. Movellan. Detecting contingencies: An infomax approach. *Neural Networks*, 23(8–9):973–984, 2010.

[6] Nicholas J. Butko and Javier R. Movellan. Infomax control of eye movements. *IEEE Transactions on Autonomous Mental Development*, 2(2):91–107, June 2010.

[7] Nicholas J. Butko and Javier R. Movellan. Learning to look. In *Proceedings of the 2010 IEEE International Conference on Development and Learning*, pages 70–75, August 2010.

[8] Burkhart Fischer. The preparation of visually guided saccades. *Reviews of Physiology, Biochemistry and Pharmacology*, 106:1–35, 1987.

[9] A. L. Yarbus. *Eye Movements and Vision*. Plenum, New York, 1967.

[10] Donald R. Griffin and Robert Galambos. The sensory basis of obstacle avoidance by flying bats. *Journal of Experimental Zoology*, 86(3):481–506, 1941.

[11] John S. Watson. Smiling, cooing, and "the game". *Merrill-Palmer Quarterly*, 18:323–339, 1972.

[12] Javier R. Movellan and John S. Watson. Perception of directional attention. In *Infant Behavior and Development: Abstracts of the 6th International Conference on Infant Studies*, NJ, 1987. Ablex.

[13] David Kirsch and Paul Maglio. On distinguishing epistemic from pragmatic action. *Cognitive Science*, 18:513–549, 1994.

[14] D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):985–1005, 1956.

[15] Edwin Hutchins. *Cognition In The Wild*. The MIT Press, 1995.

[16] Wikipedia. Earth's energy budget. `http://en.wikipedia.org/wiki/Earth's_energy_budget`.

[17] Wikipedia. Pupil. `http://en.wikipedia.org/wiki/Pupil`.

[18] Christopher M. Harris. Does saccadic undershoot minimize saccadic flight-time? A Monte-Carlo study. *Vision research*, 35(5):691–701, 1995.

[19] Noam Chomsky. *Knowledge of language: its nature, origin and use*. Praeger, 1986.

[20] David C. Knill and Whitman Richards, editors. *Perception as Bayesian Inference*. Cambridge University Press, 1996.

[21] David C. Knill, Daniel Kersten, and Alan Yuille. Introduction: A bayesian formulation of visual perception. In *Perception as Bayesian Inference*, chapter 0, pages 1–21. Cambridge University Press, 1996.

[22] Yair Weiss, Eero P. Simoncelli, and Edward H. Adelson. Motion illusions as optimal percepts. *Nature Neuroscience*, 5(6):598–604, 2002.

[23] Zili Liu, David C. Knill, and Daniel Kersten. Object classification for human and ideal observers. *Vision Research*, 35(4):549–568, 1995.

[24] Marc O. Ernst and Martin S. Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415:429–433, 2002.

[25] Scott O. Murray, Daniel Kersten, Bruno A. Olshausen, Paul Schrater, and David L. Woods. Shape perception reduces activity in human primary visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 12(23):15164–15169, 2002.

[26] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[27] Christof Teuscher and Jochen Triesch. To each his own: The caregiver's role in a computational model of gaze following. *Neurocomputing*, 70:2166–2180, 2007.

[28] Richard E. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38:716–719, 1952.

[29] W. Schultz, P. Dayan, and P.R. Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593, 1997.

[30] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[31] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Point-based value iteration: an anytime algorithm for pomdps. In *IJCAI*, pages 1025–1032, Acapulco, Mexico, 2003.

[32] Jonathan Baxter and Peter L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, November 2001.

[33] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *Proceedings of the 22nd Conference on Artificial Intelligence*, 2007.

[34] Horace B. Barlow. Possible principles underlying the transformation of sensory messages. In W. A. Rosenblith, editor, *Sensory Communication*, pages 217–234. MIT Press, 1961.

[35] Anthony J. Bell and Terrence J. Sejnowski. An information-maximzation to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[36] Simon Laughlin. A simple coding procedure enhances a neuron's information capcity. *Z. Naturforsch*, 36:910–912, 1981.

[37] Anthony J. Bell and Terrence J. Sejnowski. The "independent components" of scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[38] Michael S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, 2002.

[39] Horace B. Barlow. Banishing the homunculus. In David C. Knill and Whitman Richards, editors, *Perception as Bayesian Inference*, chapter 12, pages 425–450. Cambridge University Press, 1996.

[40] Kobi Levi and Yair Weiss. Learning object detection from a small number of examples: the importance of good features. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[41] Honghao Shan and Garrison W. Cottrell. Looking around the backyard helps to recognize faces and digits. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[42] Neil D. B. Bruce and John K. Tsotsos. Saliency based on information maximization. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 155–162, Cambridge, MA, 2006. MIT Press.

[43] Lingyun Zhang, Matthew H. Tong, Tim K. Marks, Honghao Shan, and Garrison W. Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):1–20, 2008.

[44] Ian R. Fasel, Andrew Wilt, Nassim Mafi, and Clayton T. Morrison. Intrinsically motivated information foraging. In *Proceedings of the 2010 IEEE International Conference on Development and Learning*, pages 101–107, 2010.

[45] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. I.* Athena Scientific, 3 edition, 2005.

[46] Michael Duff. *Optimal Learning: Computational Procedure for Bayes-Adaptive Markov Decision Processes.* PhD thesis, University of Massachusetts, Amherst, 2002.

[47] M. Stone. Application of a measure of information to the design and comparison of regression experiments. *Annals of Mathematical Statistics*, 30(1):55–70, 1959.

[48] V. V. Fedorov. *Theory of Optimal Experiments.* Academic Press, New York, 1972.

[49] J. M. Bernardo. *The use of information in the design and analysis of scientific experimentation.* PhD thesis, University of London, 1976.

[50] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

[51] S. Hoi, R. Jin, J. Zhu, and M. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 417–424, 2006.

[52] Jiri Najemnik and Wilson S. Geisler. Simple summation rule for optimal fixation selection in visual search. *Vision Research*, 49:1286–1294, 2009.

[53] J. D. Nelson and J. R. Movellan. Active inference in concept learning. In T. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 45–51. MIT Press, Cambridge, Massachusetts, 2001.

[54] Jonathan D. Nelson, Craig R. M. McKenzie, Garrison W. Cottrell, and Terrence J. Sejnowski. Experience matters: Information acquisition optimizes probability gain. *Psychological Science*, 21(7):960–969, 2010.

[55] J. Lewi, R. Butera, and L Paninski. Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21(3):619–687, 2009.

[56] Maya Cakmak, Crystal Chao, and Andrea Thomaz. Designing interactions for robot active learning. *Transactions on Autonomous Mental Development*, 2(2), June 2010.

[57] Emanuel Todorov. Optimality principles in sensorimotor control. *Nature Neuroscience*, 7(9):907–915, 2004.

[58] Laurent Itti, Cristof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov 1998.

[59] Laurent Itti and Pierre Baldi. A principled approach to detecting surprising events in video. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[60] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[61] Jiri Najemnik and Wilson S. Geisler. Optimal eye movement strategies in visual search. *Nature*, 434:387–391, March 2005.

[62] John Morton and Mark H. Johnson. CONSPEC and CONLERN: A two-process theory of infant face recognition. *Psychological Review*, 98(2):164–181, 1991.

[63] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[64] Daniel Kersten and Alan Yuille. Bayesian models of object perception. *Current Opinion in Neurobiology*, 13(2):150–158, 2003.

[65] Angela J. Yu and Jonathan D. Cohen. Sequential effects: Superstition or rational behavior? In *Advances in Neural Information Processing Systems*, volume 21, pages 1873–1880, 2009.

[66] E.S. Bromberg-Martin and O. Hikosaka. Midbrain dopamine neurons signal preference for advance information about upcoming rewards. *Neuron*, 63(1):119–126, 2009.

[67] Jean-René Duhamel, Carol L. Colby, and Michael E. Goldberg. The updating of the representation of visual space in parietal cortex by intended eye-movements. *Science*, 255(5040):90–92, January 1992.

[68] I. W. R. Bushnell, F. Sai, and J. T. Mullin. Neonatal recognition of the mother's face. *British Journal of Developmental Psychology*, 7:3–15, 1989.

[69] R. Linsker. From basic network principles to neural architecture: emergence of oriented columns. *Proceedings of the National Academy of Sciences of the United States of America*, 83:8779–8783, 1986.

[70] K. D. Miller. A model for the development of simple cell receptive fields and the ordered arrangement of oriented columns through activity-dependent competition between ON- and OFF-center inputs. *The Journal of Neuroscience*, 14(1):409–441, 1994.

[71] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61:183–193, 1954.

[72] R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[73] J. P. Nadal and N. Parga. Nonlinear neurons in the low-noise limit: a factorial code maximizes information transfer. *Network: Computation in Neural Systems*, 5(4):565–581, 1994.

[74] Eero P. Simoncelli and Bruno A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.

[75] P. Lennie. The cost of cortical computation. *Current Biology*, 13:493–497, 2003.

[76] R. J. Baddeley, L. F. Abbot, M. C. Booth, F. Sengpiel, T. Freeman, E. A. Wakeman, and E. T. Rolls. Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proceedings of the Royal Society of London, B*, 264:1775–1783, 1998.

[77] P. Földiák. Sparse coding in the primate cortex. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge, MA, 2 edition, 2002.

[78] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[79] W. Zhang and D. J. Linden. The other side of the engram: experience-dependent changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, 4:885–900, 2003.

[80] N. S. Desai, L. C. Ruthorford, and G. Turrigiano. Plasticity in the intrinsic excitability of cortical pyramidal neurons. *Nature Neuroscience*, 2(6):515–520, 1999.

[81] D. DeSieno. Adding a conscience to competitive learning. In *Proceedings of the 1988 IEEE International Conference on Neural Networks*, pages 117–124, 1988.

[82] M. S. Falconbridge, R. S. Stamps, and D. R. Badcock. A simple hebbian / anti-hebbian network learns the sparse, independent components of natural images. *Neural Computation*, 18:415–429, 2005.

[83] P. Földiák. Forming sparse representation by local anti-hebbian learning. *Biological Cybernetics*, 64:165–170, 1990.

[84] M. Stemmler and Christof Koch. How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nature Neuroscience*, 2:521–527, 1999.

[85] Jochen Triesch. Synergies between intrinsic and synaptic plasticity in individual neurons. In *Advances in Neural Information Processing Systems*, volume 17, pages 1417–1424, 2005.

[86] Jochen Triesch. Synergies between intrinsic and synaptic plasticity mechanisms. *Neural Computation*, 19:885–909, 2007.

[87] B. T. Vincent, R. J. Baddeley, T. Troscianko, and I. D. Gilchrist. Is the early visual system optimised to be energy efficient? *Network: Computation in Neural Systems*, 16(2–3):175–190, 2005.

[88] L. N. Cooper, N. Intrator, B. S. Blais, and H. Z. Shouval. *Theory of Cortical Plasticity*. World Scientific, London, 2004.

[89] Jochen Triesch. A gradient rule for the plasticity of a neuron's intrinsic excitability. In *Proceedings of the International Conference on Artificial Neural Networks*, volume 65–70, 2005.

[90] J. Lücke and Christoph von der Malsburg. Rapid processing and unsupervised learning in a model of the cortical macrocolumn. *Neural Computation*, 16(3):501–533, 2004.

[91] J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings of the Royal Society of London, B*, 265:359–366, 1998.

[92] Aapo Hyvärinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[93] R. Miikkulainen, J. Bednar, Y. Choe, and J. Sirosh. *Computational Maps In The Visual Cortex*. Springer, 2005.

[94] Aapo Hyvärinen, P. O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13:1527–1558, 2001.

[95] Cornelius Weber. Self-organization of orientation maps, lateral connections, and dynamic receptive fields in the primary visual cortex. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 1147–1152, 2001.

[96] Antonio Torralba, Aude Oliva, Monica S. Castelhano, and John M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113(4):766–786, 2006.

[97] V. Navalpakkam and L. Itti. An integrated model of top-down and bottom-up attention for optimal object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2056, New York, NY, Jun 2006.

[98] Lingyun Zhang, Matthew H. Tong, and Garrison W. Cottrell. Information attracts attention: A probabilistic account of the cross-race advantage in visual search. In *Proceedings of the 29th Annual Cognitive Science Conference*, 2007.

[99] Dashan Gao and Nuno Vasconcelos. Bottom up saliency is a discriminant process. In *IEEE International Conference on Computer Vision*, 2007.

[100] Wolf Kienzle, Felix A. Wichmann, Bernhard Schölkopf, and Matthias Franz. A nonparametric approach to bottom-up visual saliency. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[101] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, Jan 1980.

[102] Javier R. Movellan, Fumihide Tanaka, Bret Fortenberry, and Kazuki Aisaka. The rubi project: Origins, principles and first steps. In *Proceedings of the International Conference on Development and Learning (ICDL)*, 2005.

[103] Javier R. Movellan, Fumihide Tanaka, Ian R. Fasel, Cynthia Taylor, Paul Ruvolo, and Micah Eckhardt. The rubi project: A progress report. In *HRI*, pages 333–339, 2007.

[104] Ian Fasel, Bret Fortenberry, and J. R. Movellan. A generative framework for real-time object detection and classification. *Computer Vision and Image Understanding*, 98:182–210, 2005.

[105] L. R. Bahrick and J. S. Watson. Detection of intermodal proprioceptive-visual contingency as a potential basis of self-perception in infancy. *Developmental Psychology*, 21:963–973, 1985.

[106] John S. Watson. Contingency perception in early social development. In T. M. Field and N. A. Fox, editors, *Social perception in infants*, pages 157–176. Ables, New Jersey, 1985.

[107] A. E. Bigelow. Infant's sensitivity to imperfect contingency in social interaction. In P. Rochat, editor, *Early social cognition: understanding others in the first months of life*, pages 241–256. LEA, New York, 1999.

[108] John S. Watson. The perception of contingency as a determinant of social responsiveness. In E. B. Thoman, editor, *Origins of the Infant's Social Responsiveness*, pages 33–64. LEA, New York, 1979.

[109] Javier R. Movellan and John S. Watson. The development of gaze following as a Bayesian systems identification problem. In *Proceedings of the 2002 IEEE International Conference on Development and Learning (ICDL02)*. IEEE, 2002.

[110] S. Johnson, V. Slaughter, and B. Carey. Whose gaze will infants follow? the elicitation of gaze-following in 12-month-olds. *Developmental Science*, 1(2):233–238, 1998.

[111] Jonathan D. Nelson, Joshua B. Tenenbaum, and Javier R. Movellan. Active inference in concept learning. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 692–697. LEA, Edinburgh, Scotland, 2001.

[112] Nicholas J. Butko and Javier R. Movellan. Optimal scanning for faster object detection. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2009.

[113] Nicholas J. Butko and Javier R. Movellan. I-POMDP: An infomax model of eye movement. In *Proceedings of the 2008 IEEE International Conference on Development and Learning*, pages 139–144, August 2008.

[114] Dimitri P. Bertsekas and S. Shreve. *Stochastic Optimal Control*. Athena Scientific, 1996.

[115] R.A. Wise. Dopamine, learning and motivation. *Nature Reviews Neuroscience*, 5(6):483–494, 2004.

[116] P.R. Montague, S.E. Hyman, and J.D. Cohen. Computational roles for dopamine in behavioural control. *Nature*, 431(7010):760–767, 2004.

[117] S. Edelman and L. M. Vaina. David marr. *International Encyclopedia of the Social and Behavioral Sciences*, 2001.

[118] J.M. Wolfe, K.R. Cave, and S.L. Franzel. Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology: Human perception and performance*, 15(3):419–433, 1989.

[119] Laurent Itti and Pierre Baldi. Bayesian surprise attracts human attention. In *Advances in Neural Information Processing Systems, Vol. 19 (NIPS*2005)*, pages 1–8, Cambridge, MA, 2006. MIT Press.

[120] Nathan Sprague and Dana Ballard. Eye movements for reward maximization. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[121] A. Simpkins, R. de Callafon, and E. Todorov. Optimale trade-off between exploration and exploitation. In *Proceedings of the American Conference on Control*, pages 33–38, 2008.

[122] Satinder Singh, Richard L. Lewis, Andrew G. Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *Transactions on Autonomous Mental Development*, 2(2), June 2010.

[123] Christopher M. Kanan, Matthew H. Tong, Lingyun Zhang, and Garrison W. Cottrell. Sun: Top-down saliency using natural statistics. *Visual Cognigiton*, 17(6&7):979–1003, 2009.

[124] Niko Sam, R. Hari, O, S. Lu, and J. Simola. Seeing speech: visual information from lip movements modifies activity in the human auditory cortex. *Neuroscience letter*, 127:141–145, 1991.

[125] Riikka Mottonen, Christina M. Krause, Kaisa Tiippana, and Mikko Sams. Processing of changes in visual speech in the human auditory cortex. *Brain Res Cogn Brain Res*, 13(3):417–25, May 2002.

[126] Maunsell JH and Treue S. Feature-based attention in visual cortex. *Trends in Neuorosciences*, 29(6):317–322, June 2006.

[127] Anthony J. Bell and Terrence J. Sejnowski. Edges are the independent components of natural scenes. In *Advances in Neural Information Processng Systems*, volume 9. MIT, 1996.

[128] Martin Handford. *Where's Waldo*. Candlewick, 1987.

[129] Nicholas J. Butko. Nick's Machine Perception Toolbox. `http://mplab.ucsd.edu/~nick/NMPT`, 2008.

[130] Rafael Beserra Gomes, Luiz Marcos Garcia Gonalves, and Bruno Motta de Carvalho. Real time vision for robotics using a moving fovea approach with multi resolution. In *International Conference on Robotics and Automation (ICRA)*, May 2008.

[131] `http://mplab.ucsd.edu`. The MPLab GENKI Database, GENKI-SZSL Subset.

[132] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[133] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hoffman. Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.

[134] `http://www.cs.indiana.edu/cgi-pub/oleykin/website/OpenCVHelp/`. The OpenCV 1.0 API.

[135] Julia Vogel and Nando de Freitas. Target-directed attention: Sequential decision-making for gaze planning. In *International Conference on Robotics and Automation (ICRA)*, May 2008.

[136] Max Lungarella and Olaf Sporns. Mapping information flow in sensorimotor networks. *PLoS Computational Biology*, 2(10), 2006.

[137] Richard S. Sutton. Verification, the key to AI. `http://webdocs.cs.ualberta.ca/~sutton/IncIdeas/KeytoAI.html`, November 2001.

[138] Tim K. Marks, John R. Hershey, and Javier R. Movellan. Tracking motion, deformation, and texture using conditionally gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2), February 2010.

[139] Sean P. Engelson and Drew V. McDermott. Error correction in mobile robot map learning. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 2555 – 2560, 1992.

[140] Sebastian Thrun. Particle filters in robotics. In *Proceedings of the Proceedings of the Eighteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 511–519, San Francisco, CA, 2002. Morgan Kaufmann.

[141] Richard A. Abrams, David E. Meyer, and Sylvan Kornblum. Speed and accuracy of saccadic eye movements: Characteristics of impulse variability in the oculomotor system. *Journal of Experimental Psychology: Human Perception and Performance*, 15(3):529–543, 1989.

[142] Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook. Techical University of Denmark, October 2008.

[143] K. A. Kleiner and M. S. Banks. Stimulus energy does not account for 2-month-old preferences. *Journal of Experimental Psychology: Human Perception and Performance*, 13:594–600, 1987.

[144] C. C. Goren, M. Sarty, and P. Y. K. Wu. Visual following and pattern discrimination of face-like stimuli by newborn infants. *Pediatrics*, 9:415–421, 1975.

[145] John Hershey and Javier R. Movellan. Using audio-visual synchrony to locate sounds. In S. A. Solla, T. K. Leen, and K. R. Muller, editors, *Advances in Neural Information Processing Systems, 12*, pages 813–819, Cambridge, MA, USA, 2000. MIT Press.

[146] M. J. Beal, N. Jojic, and Hl Attias. A graphical model for audiovisual object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25:828, 2003.

[147] Jochen Triesch and Christof von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13:2049, 2001.

[148] Virginia R. de Sa. Learning classification with unlabeled data. In Jack D. Cowan, Gerald Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems, 4*, pages 112–119. Morgan Kaufmann, 1994.

[149] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 92–100, 1998.

[150] L. B. Cohen and C. H. Cashon. Infant object segregation implies information integration. *Journal of Experimental Child Psychology*, 78(1):75–83(9), January 2001.

[151] Peter Redgrave and Kevin Gurney. The short-latency dopamine signal: a role in discovering novel actions? *Nature Reviews Neuroscience*, 7:967–975, November 2006.

[152] John S. Watson. The development and generalization of "contingency awareness" in early infancy. *Merrill-Palmer Quarterly of Behavior and Development*, 12(2):124–135, 1966.

[153] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.

[154] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

[155] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107:1135, 2003.

[156] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems, 17*, 2004.

[157] Thomas Serre, Lior Wolf, and Tomaso Poggio. Object recognition with features inspired by visual cortex. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 994–1000, 2005.

[158] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 756–763, 2005.

[159] Ian R. Fasel. *Learning to Detect Objects in Real-Time: Probabilistic Generative Approaches*. PhD thesis, University of California at San Diego, June 2006.

[160] C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. In Sebastian Thrun, Lawrence Saul, and Bernhard Scholkpf, editors, *Advances in Neural Information Processing Systems, 16*, Cambridge, MA, USA, 2004. MIT Press.

[161] Mark H. Johnson, Suzanne Dziurawiec, Hadyn Ellis, and John Morton. Newborns' preferential tracking of face-like stimuli and its subsequent decline. *Cognition*, 40:1–19, 1991.

[162] Jochen Triesch. The role of a priori biases in unsupervised learning of visual representations: a robotics experiment. In *Developmental Embodied Cognition: DECO-2001, Workshop at the Universty of Edinburgh*, July 2001.