Learning To Look

Nicholas J. Butko and Javier R. Movellan
Machine Perception Laboratory, UC San Diego
{nick,movellan}@mplab.ucsd.edu

***Abstract*—How can autonomous agents with access to only their own sensory-motor experiences learn to look at visual targets? We explore this seemingly simple question, and find that naïve approaches are surprisingly brittle. Digging deeper, we 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 show how a Bayesian observer should trade off different sources of uncertainty in order to discover how their sensors and actuators relate. We implement our approach on two very different robots, and show that both of them can quickly learn reliable intentional looking behavior without access to anything beyond their own experiences.**

I. From Simulations to Physical Systems

There are many situations in which a robot may want orient 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 intentionally fixate an object, a robot must know what signal to send to 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 regression problem.

This process is tedious and it is impractical to calibrate the sensory-motor properties 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.

Robots with calibration parameters endowed by their experimenters violate Sutton's verification principle [1]:

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

In this paper, we consider how infants and robots may use their own developmental experiences to learn to look. We explore this seemingly simple question, and find that rule-based approaches are brittle. An alternative is to explore the computational structure of the problem [2]. It is well known how objects in the world project a 2D image onto a robot's camera, as well as how a robot's cameras generally move

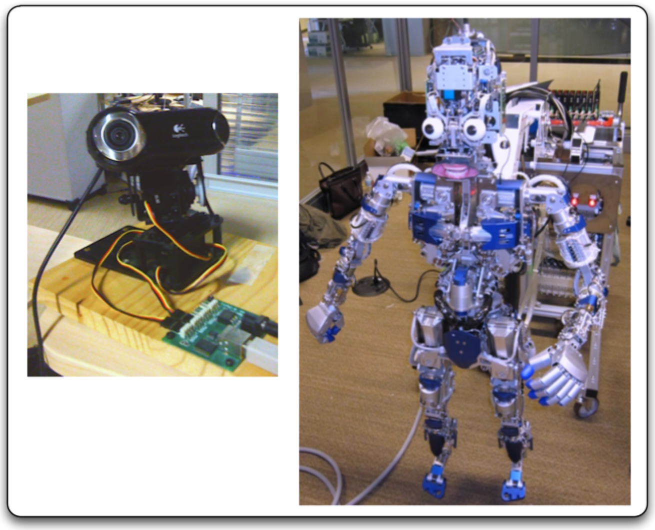

Fig. 1: Different robots like Nobody (left) and Diego-san (right) have sensory and motor capabilities. It is tedious and impractical to measure the sensory-motor properties of many different robots. It would be better if each robot could learn to use and make sense of its sensory-motor capabilities in terms of its developmental experience.

through space. We can derive an algorithm for learning to look by encoding this knowledge formally into a generative model [3]. The generative model begins with formal models of the relationships among three components:

1) How the appearance of the world changes over time.
2) How the physical parameters of a robot's motor system cause a motor signal to re-orient the robot's cameras in space.
3) How the orientation of a robot's cameras in space cause the robot to see an image of the appearance of the world.

Given this formal structure, the robot can simultaneously infer the appearance of the world, 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 process and thus efficient algorithms can be used to solve this inference problem robustly [3].

A. Nature or Nurture?

In this document, we present a model by which a robot may "learn to look." This is a necessary problem to consider because each robot may have a different configuration of motors. The motors from robot to robot may have different

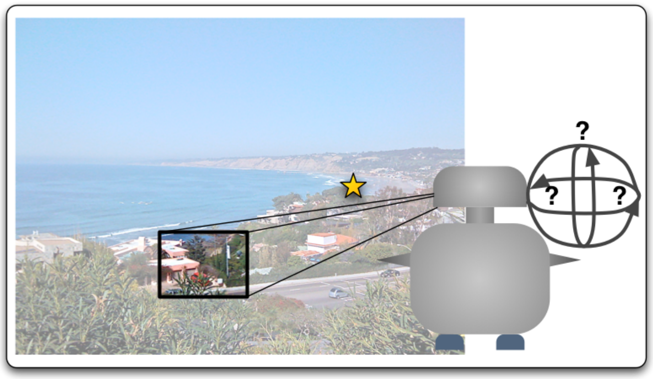

Fig. 2: This robot is currently looking at the car, but he would like to look at the beach (starred). What command should he send to his servo motors? Can the robot learn what command to send from developmental experience?

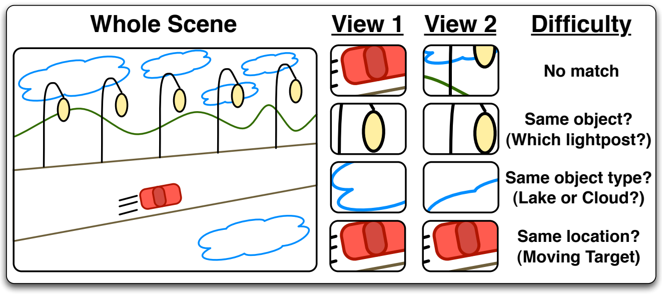
Whole Scene
View 1
View 2
Difficulty
No match
Same object? (Which lightpost?)
Same object type? (Lake or Cloud?)
Same location? (Moving Target)

Fig. 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 have a similar appearance to different objects. 4) Objects may move; assuming a matched object is in the same location may give a corrupt training signal.

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 developmental sensory-motor experience to figure out how to use its motors, seemingly putting us on the "nurture" side of the "nature vs. nurture" question.

However, we present a generative model by which the robot can anchor its motor experience in its sensory experience, and use Bayesian inference to discover how its motors work. It is "born" with a generative probabilistic model and machinery for doing Bayesian inference. This seems to argue more for the "nature" side of the "nature vs. nurture" question.

We argue that in our case, both the "nature" and "nurture" views are correct: Each robot or developing organism is "born" with a framework for learning from sensory-motor experiences how their individual bodies are organized.

B. A rule-based approach

How can a robot learn how to look at some interesting visual target? A naïve algorithm might look something like this:

1) Calculate Δ_1, the distance from your center of gaze to some object.
2) Make an arbitrary eye movement a.
3) Calculate Δ_2, the new distance from your center of gaze to the same object.
4) Calculate $\Delta_3 = \Delta_2 - \Delta_1$, the actual eye-movement caused by a.
5) Use Δ_3 as a training signal to learn the functional mapping $f(a) \rightarrow \Delta$.
6) Repeat 1–5 for many training examples, until the mapping $f(a) \rightarrow \Delta$ becomes reliable.
7) Calculate Δ_4, the distance from your center of gaze to something you want to look at.
8) Calculate $a \leftarrow f^{-1}(\Delta_4)$ to look at the target.

This approach, while basically sound, suffers from being brittle. Specifically, it requires reliable identification the same object before and after an eye-movement. Many things can cause this process to go wrong, as illustrated in Figure 3.

In order to overcome these difficulties, one possibility is to explicitly list the things we can think of that could go wrong, and encode a series of "if-then" style coping heuristics the robot can use. An alternative is to 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 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 [3].

Using the machinery of Bayesian inference, the robot can account for the exceptions in Figure 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.

II. Generative Model

Two sources of information are available to the robot moment to moment. First, the robot knows what commands it is sending to its motors (motor information). Second, it senses an array of pixels (sensory information).

Let each pixel in the sensor array be a random variable Ψ_t^x, where $x \in \mathcal{R}^2$ denotes the location of the pixel in the array, and t the time at which the image was collected. The value of the pixel, ψ_t^x, is rendered from the conditional probability distribution $p(\psi_t^x|\lambda_t^x, \tau_t)$, where λ_t^x is the appearance of the world at that point.

When the robot issues a motor command a, its camera moves to a new position. However, its retinal coordinate system has not changed. The effect of an action a is that the appearance of the world $\lambda^{x'}$ at a previous location x' moves

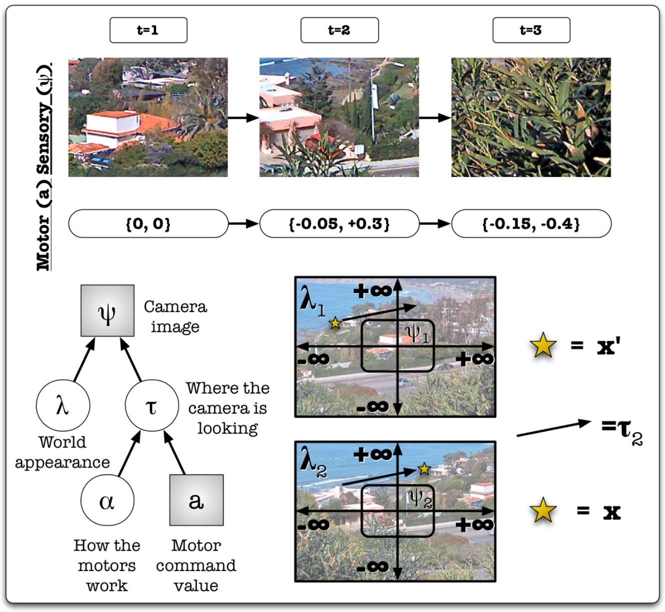

Fig. 4: *Top:* The information available to the robot moment to moment (sensory & motor). *Left:* Graphical model that gives rise to the robot's experiences. Circles denote hidden states that the robot must infer. Greek letters denote random variables. *Right:* Graphical depiction of the generative process.

to a new location x in the sensor array. This transformation is parameterized by τ, for some $f(x', \tau_t) \to x$. While τ_t literally represents the parameters of a coordinate transformation, it is useful to think of it as intuitively representing "How did the robot's eyes just move?"

The particular transform τ_t that describes the movement of the robot's cameras through the world is drawn moment-to-moment from the conditional probability distribution $p(\tau_t|a_t, \alpha_t)$, where a_t is the change to previous setting of the robot's actuators (e.g. "look down a little, look right a lot"), and α_t are the unknown motion parameters that relate the setting of the actuators to the robot's physical orientation in space. While α_t literally represents the current parameters of a model of motion, it is useful to think of it as intuitively representing "How the robot's motors work," which is ultimately what we want the robot to learn.

This generic sketch of the generative process for the robot's experiences is illustrated in Figure 4. The variables in the generative model are summarized as:

- ψ_t: "Sensor," the entire image a robot sees right now
- ψ_t^x: A single pixel at location x of that image.
- λ_t: "Light," the appearance of the whole world right now.
- λ_t^x: The appearance of the world at the single point, which is currently located at point x in the robot's sensor coordinate system. If x is outside the bounds of the sensor array, the robot cannot see this point in the world right now, but it still has an appearance.
- $\lambda_{t-1}^{x'}$ The same point in the world, which was previously at a different point, x', in the robot's sensor coordinate system at time $t-1$, before the robot moved.
- τ_t: "Transform," where the robot is looking now, relative to where it was looking previously . Recall $x = f(x', \tau_t)$.
- a_t: "Action," The command the robot just sent to change its eye-motor position.
- α_t: "Actuation Parameters," The robot's current parameters of motion; how the robot's motors currently work.

The desire to look somewhere can be equated to the desire to achieve a particular coordinate transformation τ_t, which cannot be done unless the robot knows the motion parameters α_t. Ultimately, in order for the robot to learn to look, it must infer the parameters of motion α_t. These describe the consequence of a particular actuation value, a_t.

A. Model details

The above model is generic and flexible. For the sake of implementing a solution, we are forced to fill in the probability distributions and functions. In general, we will take a "conditional Gaussian" approach, where all conditional probabilities are Gaussian. However, the overall filtering distribution will be highly non-Gaussian.

1) $p(\psi_t|\lambda_t, \tau_t)$: At every moment in time, each pixel value ψ_t^x in the sensor array is drawn from the Gaussian distribution, $\psi_t^x \sim N(\lambda_t^x, q_\lambda^2)$.[1] For things that the robot cannot currently see (i.e. x is a point beyond the bounds of the sensor array), it is useful to think of these regions as being drawn from $\psi_t^x \sim N(\lambda_t^x, \infty)$.

2) $p(\lambda_t^x|\lambda_{t-1}^{x'})$: Since objects may move, the world may change appearance over time. Rather than explicitly modeling the movement of objects, we simply say that the world changes appearance over time according to Brownian motion with drift r_λ^2: $\lambda_t^x \sim N(\lambda_{t-1}^{x'}, r_\lambda^2)$.

3) $p(\lambda_0)$: The intensity of light in an image can be represented as a real number from 0 to 1. We consider each λ_0^x to be initially drawn $i.i.d.$ from $\lambda_0^x \sim N(0.5, \sigma_{\lambda 0}^2)$.

4) $f(x', \tau_t)$: General robots may have a rich set of actuators that can move them around in space as well as change the orientation of their cameras. Based on their motions through space, they may experience translation, rotation, scaling, and sheer of the visible environment around them. For this paper we begin by considering a less general class of robots that only can rotate their cameras up and down, and left and right. We model this constrained class by considering only translation transforms. Let τ be a two-dimensional vector,

$$\tau_t \overset{\text{def}}{=} \begin{bmatrix} \tau_t^h \\ \tau_t^v \end{bmatrix}$$

then $x = x' + \tau_t$.

[1]Throughout this document, we use the notation $x \sim N(\mu, \sigma^2)$ to denote that the random variable X has its value x drawn from the normal probability distribution with mean μ and variance σ^2. We use the notation $N_x(\mu, \sigma^2)$ to denote the value of the normal probability density function for that distribution, evaluated at x. When x and μ are vectors, the variance is given by a matrix Σ.

5) $p(\tau|a,\alpha)$: For stationary robots that can pan and tilt their cameras, there are two actuators: a^A and a^B.[2] The robot does not know which is horizontal and which is vertical, and actually they may not be axis aligned, so we let horizontal translation τ_t^h and the vertical translation τ_t^v be linear functions of both motor commands:

$$\underbrace{\begin{bmatrix} \tau_t^h \\ \tau_t^v \end{bmatrix}}_{\tau_t} = \underbrace{\begin{bmatrix} a_t^A & a_t^B & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_t^A & a_t^B & 1 \end{bmatrix}}_{C_t} \underbrace{\begin{bmatrix} \alpha_t^1 \\ \alpha_t^2 \\ \alpha_t^3 \\ \alpha_t^4 \\ \alpha_t^5 \\ \alpha_t^6 \end{bmatrix}}_{\alpha_t}$$

where $\alpha_t^{1:3}$ described how much the camera moves in the horizontal direction, and $\alpha_t^{4:6}$ describe how much the camera moves in the vertical direction. The learned motion parameters $\alpha_t^{1:3}$ for two robots are shown in Figure 5 Row 1, and $\alpha_t^{4:6}$ are shown in Row 2.

For less simple robots, more complicated features of the motors may be useful, such as $\sin(a^A)$ or $(a^A + a^B)^2$. Generally, let $\phi(a)$ be a function that outputs a length-m feature vector of the actuation values, and let τ have n elements; then C is an nxmn block-diagonal matrix where each block is $\phi(a)^T$, and α is a vector of length mn.

In practice, it may be useful to think of the robot's actuators as somewhat unreliable. In this event, we say that on each moment, the elements of τ_t are drawn from the Gaussian distribution, $\tau_t \sim N(C_t\alpha_t, Q_\alpha)$, where Q_α is an nxn covariance matrix describing the reliability of the motors with respect to the elements of τ_t.

6) $p(\alpha_t|\alpha_{t-1})$: In principle, the motion parameters α may change over time. Infants' developing bodies change radically throughout their development, but even in robots, gears may become looser or stiffer over time and change the parameters of motion slightly, according to Brownian motion with drift R_α: $\alpha_t \sim N(\alpha_{t-1}, R_\alpha)$, where R_α is mnxmn.

7) $p(\alpha_0)$: Initially we are very uncertain about the mn parameters of motion. We consider each as being drawn from some prior belief distribution, $\alpha_0 \sim N(\vec{0}, \Sigma_{\alpha 0})$.

B. Implementation Parameters

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

III. Inferring parameters of motion

Although each component of the model presented above is conditionally Gaussian, the filtering distribution $p(\lambda_t, \tau_t, \alpha_t|\psi_{1:t}, a_{1:t})$ is highly non-Gaussian. This arises from the index remapping function $f(x', \tau_t) \to x$, in which changing τ_t slightly may lead to a large change in $p(\psi_t|\lambda_t, \tau_t)$.

[2]We encode actions in a relative coordinate system where positive is one direction of motion, negative is another, and the value 1 represents the entire range of motion (i.e. a value of -.1 moves $1/10^{th}$ of the full range of motion in the negative direction).

TABLE I: Model Parameters & Implementation Values

	World Appearance Model		Motor Model	
	Parameter	Value	Parameter	Value
Prior Value	$\mu_{\lambda 0}$	0.5	$\vec{\mu}_{\alpha 0}$	$\vec{0}$
Prior Variance	$\sigma^2_{\lambda 0}$	0.5^2	$\Sigma_{\alpha 0}$	$500^2 I$
Dynamics Variance	r^2_λ	0.01^2	R_α	$5^2 I$
Sensor Variance	q^2_λ	0.1^2	Q_α	$20^2 I$

However, if we were given $\tau_{1:t}$, we could write the filtering distribution

$$\begin{aligned} p(\lambda_t, &\alpha_t|\tau_{1:t}, \psi_{1:t}, a_{1:t}) = \\ &= p(\alpha_t|\tau_{1:t}, a_{1:t}) p(\lambda_t|\tau_{1:t}, \psi_{1:t}) \\ &= N_{\alpha_t}(\bar{\mu}_{\alpha t}, \bar{\Sigma}_{\alpha t}) \prod_x N_{\lambda^x}(\bar{\mu}_{\lambda^x t}, \bar{\sigma}^2_{\lambda^x t}) \end{aligned}$$

where $\alpha_t \sim N(\bar{\mu}_{\alpha t}, \bar{\Sigma}_{\alpha t})$ is a Kalman Filter estimate of the posterior motion parameter filtering distribution, and $\lambda_x \sim N(\bar{\mu}_{\lambda^x t}, \bar{\sigma}^2_{\lambda^x t})$ is a Kalman Filter estimate of the posterior filtering distribution for the appearance of pixel-location x (i.e. there is a separate Kalman Filter per pixel of the world).

In such a situation, a Rao-Blackwellized Particle Filter might be used to sample from trajectories of $\tau_{1:t}$ according to the posterior distribution $p(\tau_{1:t}|\psi_{1:t}, a_{1:t})$, while maintaining Kalman Filter estimates for $p(\alpha_t|\tau_{1:t}, a_{1:t})$ and $p(\lambda_t|\tau_{1:t}, \psi_{1:t})$ for each sampled τ trajectory. This approach was taken in [3] and is commonly used in SLAM applications [4]. While a full RBPF implementation may be helpful for general mobile robots, in our case we consider simpler, stationary robots that can only move their cameras.

A simpler approach is to construct the single trajectory $\tau^*_{1:t}$, for which at every time t, τ^*_t is the coordinate transform with maximum probability given the previous estimated trajectory,[3] i.e. $\tau^*_t = \text{argmax}_{\tau_t} p(\tau_t|\tau_{1:t-1}, \psi_{1:t}, a_{1:t})$:

$$\begin{aligned} p(&\tau_t|\tau_{1:t-1}, \psi_{1:t}, a_{1:t}) = \\ &= \frac{p(\tau_t|\tau_{1:t-1}, a_{1:t}) p(\psi_{1:t}|\tau_{1:t-1}, a_{1:t})}{p(\psi_{1:t}|a_{1:t-1}, \tau_{1:t-1})} \\ &= p(\tau_t|\tau_{1:t-1}, a_{1:t}) p(\psi_t|\psi_{1:t-1}, \tau_{1:t-1}) \frac{p(\psi_{1:t-1}|\tau_{1:t-1})}{p(\psi_{1:t}|\tau_{1:t-1})} \\ &= Z\ N_{\tau_t}(C_t\bar{\alpha}_t, \quad C_t\bar{\Sigma}_{\alpha t}C_t^T + Q_\alpha) \\ &\quad \prod_x N_{\psi_t^x}(\bar{\mu}_{\lambda^x t}, \quad \bar{\sigma}^2_{\lambda^x t} + q^2_\lambda) \end{aligned}$$

where Z is a constant with respect to τ_t, and can be ignored in finding the argmax. Rewriting this as a log function and ignoring terms that don't depend on τ_t, which preserves the

[3]Note that this may not be the most likely coordinate transform of all, which would be $\tau_t^{**} = \text{argmax}_{\tau_t} p(\tau_t|\psi_{1:t}, a_{1:t})$.

maximum, gives a function $g(\tau_t)$ with three terms:

$$g(\tau_t) =$$

$$= -.5\,\overbrace{(\tau_t - C_t\bar{\mu}_{\alpha t})^T(C_t\bar{\Sigma}_{\alpha t}C_t^T + Q_\alpha)^{-1}(\tau_t - C_t\bar{\mu}_{\alpha t})}^{Predicted\ Motion\ Match}$$

$$-\,.5\sum_x \underbrace{\frac{(\psi_t^x - \bar{\mu}_{\lambda^x t})^2}{(\bar{\sigma}^2_{\lambda^x t} + q^2_\lambda)}}_{Image\ Match} - .5\sum_x \underbrace{\log(\bar{\sigma}^2_{\lambda^x t} + q^2_\lambda)}_{Uncertainty\ Penalty}$$

Each term has an important meaning:

- *Predicted Motion Match:* Prefer coordinate-transforms that are close to what we expect for the current action, C_t, based on the current estimate of the motion model, $\bar{\mu}_{\alpha t}$.
- *Image Match:* Prefer coordinate-transforms that give a match, pixel-for-pixel, between what we are seeing now, ψ_t, and what we remember the world should look like, $\bar{\mu}_{\lambda^x t}$. Note that $\bar{\mu}_{\lambda^x t}$ depends on τ_t through the transform $x = x' + \tau_t$.
- *Uncertainty Penalty:* If possible, prefer a transform where you know what the world looks like. This term is important for discouraging inferences like "I've never seen that part of the world before, so let's just assume that this is what it looks like." Note that $\bar{\sigma}^2_{\lambda^x t}$ depends on τ_t through the transform $x = x' + \tau_t$.

The function $g(\tau_t)$ gives us a way to score and compare candidate transforms τ_t, but evaluating a single τ_t is somewhat expensive, involving every pixel ψ^x in the sensor image ψ. There are efficient ways to search the space of τ_t to find a local maximum [5], but for this paper, we just search exhaustively for the maximum.

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

1) Choose a new action a_t, observe a new image ψ_t.
2) Search the space of τ_t for $\tau_t^* = \text{argmax}_\tau\, g(\tau_t)$.
3) Update the coordinates for the Kalman Filter estimates of world appearance, $\bar{\mu}_{\lambda^{x'} t}$, and the uncertainties in those appearances, $\bar{\sigma}^2_{\lambda^{x'} t}$, by $x = x' + \tau_t^*$.
4) Update the Kalman Filters for $p(\alpha_t|\tau_{1:t}, a_{1:t})$ and $p(\lambda_t|\tau_{1:t}, \psi_{1:t})$ given the current τ_t^*, ψ_t, and a_t.
5) Repeat steps 1 – 4 forever.

A. Neural Implementation

The algorithm 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 world (λ) looks to line up with what it's about to see. A strikingly similar remapping process has been observed in monkey LIP [6]. 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.

IV. Nobody & Diego Learn to Look

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.

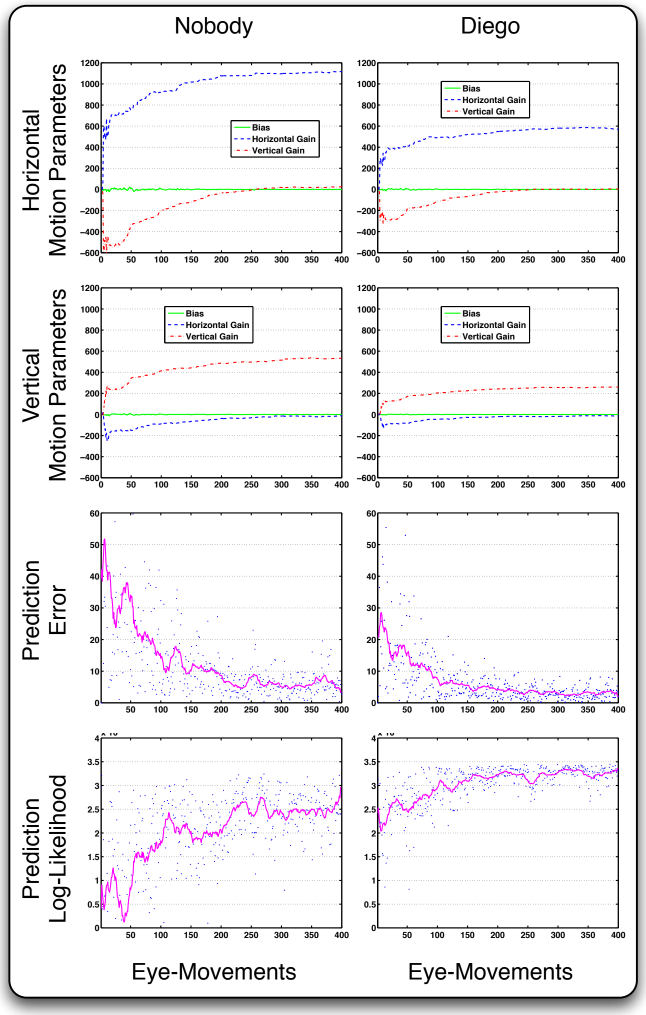

Fig. 5: *Rows 1&2:* Stable parameters of motion, α are learned. *Row 3:* Euclidean distance from the intended fixation target τ_t^I to the robot's best guess of the actual fixation target, τ_t^*. *Row 4:* Likelihood of the intended target $g(\tau_t^I)$ increases over time.

- **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 eye-movements and facial expressions.

Both robots were initialized with the same parameters (Table I), and moved their eyes according to an identical Brownian motion trajectory for a total of 400 eye-movements. On each fixation t, they computed τ_t^I, the place that they intended to look, as well as τ_t^*, their best guess of where they actually looked.

The trajectories of learning are shown in Figure 5. The learned parameters of motion, α_t, stabilize given sufficient experience. For both robots, horizontal and vertical motion are learned to be independently controlled by different motors. From the intended fixation target τ_t^I to the robot's best guess of the actual fixation target, τ_t^* the Euclidean distance

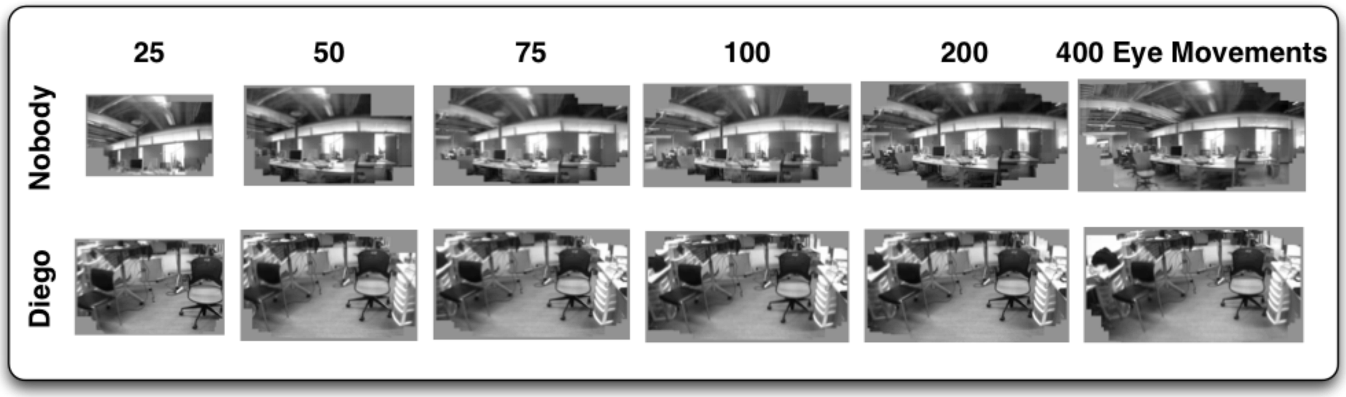

Fig. 6: Estimates of $\bar{\mu}_{\lambda^x t}$, the appearance of the world, at all locations, x, at time points, $t = \{25, 50, 75, 100, 200, 400\}$, during each robot's learning.

decreases, until the robot is able to look at intended targets. The likelihood of the intended target $g(\tau_t^I)$ increases over time, indicating that both robots are better able to predict the consequences of their actions. Together, these show that both robots develop intentional looking behavior without access to anything but subjective, verifiable experiences.

The learning of the motor parameters α_t relies heavily on the robot's estimate $\bar{\mu}_{\lambda^x t}$, the robot's memory of the appearance of the world 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.

V. Sensory-Motor Development

Initially, we laid out one property of physical eye-movement that we targeted for developmental learning. However, there are many other properties of physical eye-movements that a robot may find it useful to be aware of:

1) The time course from execution to completion of an eye-movement.
2) 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.
3) The quality of image frames collected during an eye movement.
4) 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 sensory-motor experiences; the approach that we took was powerful and robust enough to enable two 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) Rather than simply analyzing images after an eye-movement, the same analysis can be applied to the whole series of camera frames from the time of issuing a motor command to the time of its completion. This gives the position trajectory of the camera throughout an eye-movement.

Problem 2) Figure 6 shows each robot's entire world, from one motion extreme to the other. By comparing its instantaneous field of view to the total, the robot can solve this problem.

Problem 3) The likelihood function $g(\tau^*)$ measures how well what the robot sees matches what it remembers. As the camera image becomes blurred and distorted from motion, the match between ψ_t^x and $\bar{\mu}_{\lambda^x t}$ will plummet, as reflected in the dynamics of $g(\tau^*)$ 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 4) Motion is captured by temporal variation in the appearance of the robot's world. By empirically estimating this variance at each location, the robot not only can estimate how much objects in its world 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.

Acknowledgments

This work was supported by the National Science Foundation (NSF) grants IIS INT2 0808767, and ECCS 0622229 Infomax Neural Networks for Real-Time Learning and Control.

References

[1] R. S. Sutton, "Verification, the key to AI," http://webdocs.cs.ualberta.ca/~sutton/IncIdeas/KeytoAI.html, November 2001.

[2] D. 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] T. K. Marks, J. R. Hershey, and J. R. Movellan, "Tracking motion, deformation, and texture using conditionally gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, February 2010.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[5] B. D. Lucas and T. Kanade, "An iterative image registration technique with application to stereo vision." in *Proceedings of Image understanding workshop*, 1981, pp. 121–130.

[6] J.-R. Duhamel, C. L. Colby, and M. E. Goldberg, "The updating of the representation of visual space in parietal cortex by intended eye-movements," *Science*, vol. 255, no. 5040, pp. 90–92, January 1992.