

Learning Acceptable Windows of Contingency

Kevin Gold and Brian Scassellati

Department of Computer Science

Yale University

New Haven, CT, USA

Email: kevin.gold@yale.edu, scaz@cs.yale.edu

Abstract—By learning a range of possible times over which the effect of an action can take place, a robot can reason more effectively about causal and contingent relationships in the world. However, learning these time windows in a noisy environment where random events interfere can pose a challenge. We present an algorithm for learning the interval $[t_{1_{min}}, t_{1_{max}}]$ of possible times during which a response to an action can take place, and implement the model on a physical robot for the domains of visual self-recognition and auditory social-partner recognition. The environment model that we use to justify our error bounds assumes that natural environments generate Poisson distributions of random events at all scales. From this assumption, we derive a linear-time algorithm, which we call *Poisson threshold learning*, for finding a threshold T that provides an arbitrarily small rate of background events $\lambda(T)$ if such a threshold exists for the specified error rate. We can then use this rate to calculate an expected number of false positives in our sample data and discard them. We implement the principles of our method using a motion detection module as our input stream in the visual domain, and sampled audio energy in the auditory domain. In this way, we find time windows for self-generated motion, self-generated audio, and verbal social responses. We also present data on the distributions of these events, showing that while our self-generated action had a normal distribution, the social events were better modeled by a Poisson process. Finally, we present several applications for which such simple classifiers could potentially prove useful, such as mirror self-recognition and learning the meanings of the words “I” and “you.”

I. INTRODUCTION

The identification of self-generated sensory information and the identification of social partners are both primarily problems of causality. If a motor command reliably causes a certain response in the visual field, a robot may correctly assume that it is seeing itself move. Similarly, social partner identification is not simply a matter of finding faces in the visual field; a person is not interacting with the robot unless she is *responding* to the robot – again, a judgment about causality.

To reason about causality, it is not enough to look at the order of events; one must also take into account when those events occurred. If one sends a letter in the mail and receives a letter from the recipient the very same day, it probably isn’t a response. The various nodes running in parallel on a modern robot architecture must contend with exactly this problem; motor responses and visual or audio feedback may not be immediate, and assuming that they are can lead to mistakes. The round trip time between command and response may be quite large, for instance, in the case of a robot being controlled by a rack of computers

that is not physically close to the robot’s sensors and end-effectors.

Identifying a chain of causality does not necessarily require logical reasoning. Humans appear to have the ability to detect contingent responses from an early age. Infants at 12 months are willing to follow the “gaze” of a faceless object that emits beeps and flashes contingent upon the infant’s own behavior, but will not follow the object’s gaze if the beeps and flashes are emitted at random [1]. Even 6-to-12-week-old infants become agitated if a closed-circuit video feed of their mothers interacting with them is replaced with a non-contingent recording [2]. Experiments with adults show that the illusion of animacy conferred by an object acting without visible cause is largely perceptual and automatic [3], suggesting that these decisions are fairly low-level, and therefore potentially easy to emulate.

From a machine-learning point of view, the learning of the window of acceptable times for an event is also very tractable. The problem of learning the minimum and maximum times between a cause and a particular effect is essentially that of learning an interval $[a, b]$ on the real number line. It is straightforward to prove that the Vapnik-Chervonenkis dimension, or VC-dimension, of this problem is 2, and that therefore the number of examples required to learn this interval with probability at least $1 - \delta$ and error at most ϵ is at most $\frac{13}{\epsilon} (2 \ln \frac{1}{\epsilon} + \ln \frac{1}{\delta})$ by a theorem in [4]. This result holds regardless of the distribution of the data, which is useful for dealing with potentially unpredictable events such as the actions of a human being. In a noiseless environment, once this many training examples have been presented, events falling outside the time window can be confidently discarded as not belonging to the class in question – whether the domain is self-generated motion or social responses.

The problem is that in the real world, we are not provided with clear positive and negative examples, as we would be in an ideal theoretical environment. Instead, we are presented with streams of real-valued sensor data, with higher values that may reflect either a contingent reaction to the robot’s actions, unrelated events occurring in the environment, or just sensor noise. If we merely take the minimum and maximum of the observed values, we run a high risk of choosing events generated by the environment as our extrema. To solve this problem, we have developed a method of adaptively setting a threshold on the input channel to guarantee a low rate of irrelevant threshold events. The method, which we call *PoissonThresholdLearning*,

relies on the assumption that the onsets of environmental events are largely uncorrelated with each other to produce a model that generates a sensor threshold with an arbitrarily low rate of background noise. As long as the events of interest produce sensor values that are above this threshold, the model can successfully learn a time window for detecting those events contingently by discarding the expected number of false positives. Moreover, the algorithm for setting the threshold runs in time linear with the sample size, and is optimal given the Poisson assumptions described above.

This does not completely solve the problem of positively identifying instances of the class, however. The problem is that while all events of class C may occur within a specific time window t , this does not logically imply that all events within time window t belong to class C . (This problem of “affirming the consequent” is common when applying simple property detectors, such as using color to judge whether an image contains skin.) Whether this is a problem in practice depends on the size of the time window. If the domain is self-generated motion, then we may hope that the responses are quick enough and reliable enough to make the probability of other events occurring within the window very small. If the domain is highly unpredictable, as it is in identifying social responses, then we may either bring in information from other sensors or attempt to reason probabilistically using Bayes’ Rule.

Whether these solutions work in practice is an empirical question – one that we shall address in this paper for the domains of self-recognition and social partner identification. In our other work, we have shown that a robot can learn to match the words “I” and “you” to the properties of “speaker” and “listener” by watching a game of catch in which the two participants comment on the action [5]. Here, we study the properties we hope to associate with the robot’s first-person understanding of these terms. By learning the grounded meanings of these terms, we hope to create robots that are better able to produce and understand social behavior.

Self-identification in [6] was performed by cross-correlating the repetitive motion of the robot’s hand with proprioceptive feedback from the motors, identifying the zero-crossings of each. This method of using motion feedback is similar to our approach in the visual domain. Cohen et al. [7] used Allen’s six possible relationships between fluents [8], instead of the specific time windows, to characterize the effects of their robot’s actions. Other recent approaches for learning about the self have included statistically reasoning about invariant visual properties [9] and mapping visual feedback to tactile feedback [10]. Much less work has been done on learning to detect social contingency, but [11] presented a robot programmed to do so; the probabilities it used were calculated using a supercomputer that ran a dynamic programming algorithm on human-collected experimental data. Finally, our work follows the general paradigm of a robot learning about its environment with very little preprogrammed knowledge – a methodology first proposed by Turing [12] and currently

known as “autonomous mental development” [13], “epigenetic robotics” [14], or “developmental robotics” (e.g., this issue of *Connection Science*).

We have implemented our system for detecting self-generated feedback in both the visual and auditory domains on Nico, a humanoid robot that has the same physical proportions as a human infant. (Some of the work presented here for the visual domain first appeared in [15], as well as in the 2005 AAAI Spring Symposium on developmental robotics.) In the auditory domain, we have extended the system to handle the detection of social responses as well. In addition to implementing the system of time windows outlined above, we also collected data on the resulting response time distributions, to determine whether a Bayesian comparison of these distributions could produce reliable indications of whether the events were self-generated, socially contingent, or neither. We describe our results below.

II. THE TIME WINDOW MODEL

Before describing the physical robot, we will first describe our general method for learning contingency time windows.

Assume that the robot is attempting to learn via a noisy sensor S in a complicated environment E . Given a module M that issues a command to an actuator at time 0, the onset of the feedback to the module M occurs at some time given by an unknown probability distribution $P(t)$. We wish to find $\{[t_{1_{min}}, t_{1_{max}}] : P(t < t_{1_{min}} \cup t > t_{1_{max}}) < \epsilon_1\}$ for some small ϵ_1 . In other words, we want to find the interval over which we can reasonably expect self-generated feedback to return to the module issuing the command.

If we were in a noiseless environment, we could simply note that this is a learning problem with a Vapnik-Chervonenkis dimension of 2, since any collection of two points cannot be inconsistently labeled, but a dataset of 3 exists that is impossible to consistently label – the case of two positive examples on either side of a negative example. (The details of VC dimension are not terribly important to our argument, but see [16] for a full explanation.) This would entail, by a theorem in [4], that $\frac{13}{\epsilon} (2 \ln \frac{1}{\epsilon} + \ln \frac{1}{\delta})$ trials are sufficient to learn the correct window with probability at least $(1 - \delta)$ and error at most ϵ – a number polynomial in $1/\epsilon$ and $1/\delta$. (Such a problem is said to be “PAC-learnable”, where PAC stands for “probably approximately correct” [16].) However, our sensor does not detect infallible Booleans of event/non-event, but some set of numbers $\{(s_1, \dots, s_n) : s_i \in \mathbb{R}\}$, with only the vague guarantee that during an interesting event, at least one of the s_i will be “large.” (In the case of visual data, this “sensor” may actually be a vision-processing module.) In addition, the environment and the sensor’s own noise occasionally produce high values for the s_i that are at least as high as those produced by self-generated action. To learn the timing of its self-feedback, it seems as if the robot must first distinguish between its own actions and sensory data generated by noise and the

environment. The reader may well wonder, then, how is this to become a method of self-identification!

The answer lies in how we model the environment. If we assume that the likelihood of an event occurring within a time interval of length t is the same probability p regardless of when we take our measurement, and that these events are stochastically independent of each other, then taking the limit as $t \rightarrow 0$ results in the Poisson distribution:

$$p(k; \lambda t) = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \quad (1)$$

where $p(k; \lambda t)$ is the probability of k events occurring within time t , given an event density of λ . In particular, $p(0; \lambda t) = e^{-\lambda t}$, and therefore the probability of at least one event happening within a time interval t is $1 - e^{-\lambda t}$. Because of its relatively easy to achieve prerequisites, the Poisson distribution has been observed in many natural domains, including radioactive disintegrations, flying bomb hits on London during World War II, chromosome interchanges in cells, wrong numbers, and distributions of bacteria on petri plates [17].

To successfully apply the assumption of stochastic independence, we must first handle the fact that natural events are likely to occur over time, and thus will produce values for the s_i that are potentially correlated with each other. For this reason, we will only be interested in the *onsets* of these natural events, and assume that events can be tracked across their durations.

We define events in terms of a threshold T : an event takes place at time t if at least one value of s_i has risen above T , and it does not belong to an already tracked event. Then the event density becomes a function of this threshold. We assume that $\lambda(T)$ is nonincreasing, with higher sensor values no denser than lower values.

We would like a threshold T that sets the probability of an environmental event occurring between our self-initiated action and the true sensory feedback to be less than some small number ϵ . (This ϵ is not to be confused with the ϵ mentioned in the discussion of the PAC-learning model.) Using our Poisson model of the environment and an estimate t_0 of the true time delay, this gives us the equation:

$$1 - e^{-\lambda(T)t_0} < \epsilon \quad (2)$$

Rearranging the terms to find $\lambda(T)$ gives:

$$\lambda(T) < \frac{-\ln(1 - \epsilon)}{t_0} \quad (3)$$

In any sample of N sensor readings from S taken over t seconds, we expect $Nt\lambda(T)$ readings to have an event above the threshold T . This suggests the following linear time algorithm for finding an appropriate threshold, which we call *Poisson Threshold Learning*:

$$\begin{aligned} (S_0, S_1, \dots, S_N) &= \text{SAMPLE}(N, S, t) \\ M &= (\max(S_0), \max(S_1), \dots, \max(S_N)) \\ T &= \text{SELECT}(N - \lfloor \frac{-N \ln(1 - \epsilon)}{t_0 t} \rfloor, M) \end{aligned}$$

Here, $\text{SAMPLE}(N, S, t)$ returns N samples of the value sets returned by the sensor S over t seconds. Since a sensor set exceeds the threshold iff its maximum value exceeds the threshold, we need only the maximum value for each sensor set. (This can be done online during sampling, to reduce memory requirements.) On this reduced data set, we call $\text{SELECT}(i, M)$: a linear time algorithm described in [18] for finding the i th smallest element of a set without sorting. Here, it guarantees that only $N\lambda(T)$ values exceed the threshold. Because every operation presented here is linear time, the algorithm runs in $O(N + n)$ time, where N is the number of samples and n is the number of values in each sample.

Once the environment has been sampled and the threshold calculated, the robot can begin to act. For N trial movements, the system finds the length of time between sending a command to the actuator and sensing a value that exceeds the sensory threshold. Once this data has been collected, the system discards the $\lfloor 1 - e^{-\lambda(T)t_{1min}} \rfloor N$ smallest values from t_1 , since these are the expected number of false positive samples. The maximum and minimum of the remaining values become $[t_{1min}, t_{1max}]$.

Waiting between steps ensures that the robot does not attempt to act if the environment has suddenly changed; presumably, an environment that once had a reasonable λ will eventually quiet down again. Finally, we discard the $\lfloor 1 - e^{-\lambda(T)t_{1min}} \rfloor N$ environmentally generated events that we expect to occur between sending a command and t_{1min} to obtain a closer estimate of the true window.

Detection of a social response is similar, except that the timing is measured from the offset of a self-generated event. Learning the social window $[t_{2min}, t_{2max}]$ therefore requires both identifying and tracking a self-generated event with some success. This implies that at least some self-learning must occur before social learning. In a real environment, we would also require that social learning only take place in the presence of some additional cue that signalled the presence of a social partner, such as a forward-looking face. We did not, however, require this condition to be fulfilled in our tests.

III. THE ROBOT PLATFORM

Nico is a humanoid robot built to serve as a testbed for models of infant learning (Fig.1). Nico currently has a single arm with 6 degrees of freedom (DOFs), while its head-neck assembly has an additional 7 DOFs.

Four miniature CCD cameras provide streaming video in wide and narrow fields of view for each of Nico's eyes; for simplicity, we will describe the path of a single image as it passes through the image processing pipeline. A frame grabber operating at 29-30Hz receives a 320x240 image from one of the cameras. Once lens distortion is removed using a fixed lookup table, the image is passed to a motion module, which finds the absolute difference between the grayscales of the current image and the previous one. This difference image is smoothed and thresholded with an adjustable threshold T_m and passed to a pre-attentive vision module. There, large regions of thresholded motion

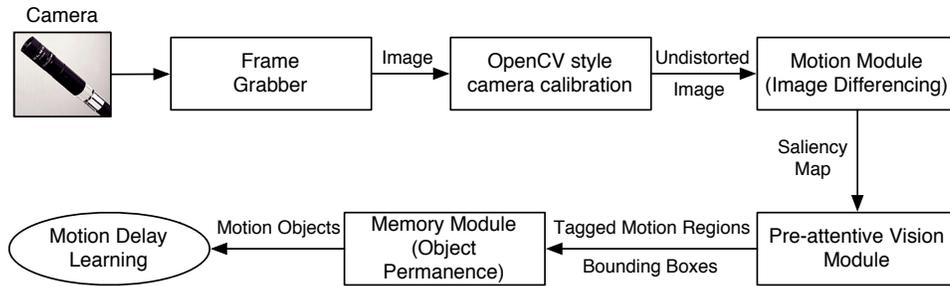


Fig. 2. The visual preprocessing pipeline.



Fig. 1. The robot, Nico. Nico is an upper-torso robot with a 6-DOF arm (left) and a 7-DOF head. Its four CCD cameras are located on either side of the bottom of Nico’s “forehead” cylinder (top, left).



Fig. 3. A bounding box around Nico’s arm indicates that its motion was identified as self-generated using the t_1 time window.

IV. SELF-RECOGNITION IN THE VISUAL DOMAIN

A. Methodology

Since the data from this portion of the paper were collected before our discovery of the Poisson threshold learning algorithm, T was set manually to provide a low $\lambda(T)$. An estimate of $\epsilon = 0.05$ was used to discard values after learning, but that was not based on a principled estimate of the underlying density $\lambda(T)$. (We intend to correct this and rerun our experiments in time for final publication.)

A series of 40 random poses was generated for Nico’s arm. Nico would then wait to ensure that the background rate of new bounding boxes was acceptably low; in practice, this meant waiting until there was no detected motion for several seconds. Nico then iterated through the poses, calculating the delay in its actions t_1 as described above by measuring the time from sending the motor command to receiving the first untracked motion bounding box. After each action, Nico returned briefly to monitoring the background rate of bounding boxes to ensure that it had not significantly changed. Finally, at the end of the trials, two outlying values were dropped (since $\epsilon N = 2$) and the remaining minimum and maximum were designated $[t_{1_{min}}, t_{1_{max}}]$. The resulting window could then be used to filter motion bounding boxes for self-generated motion (Fig. 3).

B. Results and Analysis

A subsequent 60 movement test run using the learned window resulted in only 3 movements that did not receive

are joined using a region-growing algorithm, which creates bounding box objects for each region. These events are then passed on to a tracking module before finally reaching our learning algorithm, at a frame rate of roughly 25Hz. Fig. 2 provides an overview of the vision pipeline.

In the audio domain, Nico uses a two-channel microphone connected to a Sound Blaster Live card for audio recording. Data was sampled at 44100Hz on each channel, and was sampled again to provide the raw energy of each 100ms segment as input to the learning module. Because of the instability of the Sound Blaster Live drivers on QNX, sound production was performed by a Creative CT5880 card. The resulting sound was issued through normal PC speakers.

Each module ran on one of 16 processors running the QNX Neutrino real-time operating system. The nodes communicated over a 100Mbit switch, with each module providing its output to the other modules via shared memory in a concurrency-safe fashion.

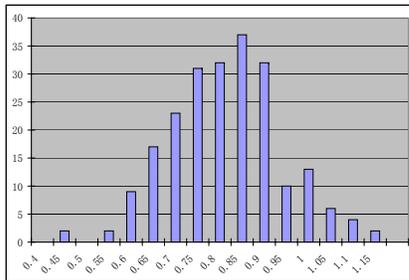


Fig. 4. Distribution of times elapsed between sending a random motor command to the robot’s arm and the perception of the resultant motion after the visual image has passed through the whole processing pipeline. The robot iterated through ten randomly selected motions 22 times.

the self-generated label – exactly the 95% accuracy for which we aimed. The learning thus took much less time than the theoretical upper bound described in the introduction, despite the relative variability of the response times (Fig. 4).

Under our normally placid laboratory conditions, it was difficult to assess how well this system would perform in a busier natural environment. Therefore, to test the limits of the system, we asked an independent subject to try to fool the robot by attempting to move when it did. If any part of the subject’s thresholded motion fell within the time window, it counted as a false positive, even if there was already some bounding box with the self-label. Despite this demanding adversarial setup, the widely variable t_1 times, and the relative simplicity of the method, the robot falsely labeled the subject’s movements only about half the time (33/60 movements).

Since our data shows that the self-feedback times form a quite well-behaved normal distribution 4, it is presumably possible to implement a Bayesian decision procedure for whether an event was more likely to come from the learned normal distribution for the self or the estimated Poisson distribution of environmentally generated events. Such an implementation would be potentially useful and straightforward to implement, but it is beyond the scope of the current paper.

V. SELF AND SOCIAL RECOGNITION IN THE AUDITORY DOMAIN

A. Methodology

Nico’s lack of text-to-speech software, combined with the fact that it doesn’t currently have much to say, required some creativity in its production of social-response eliciting actions. To this end, we used a selection of sound files of the character R2D2 from the film *Star Wars*: a character known for its emotive – if inscrutable – beeps, chirps, and whistles. A background threshold was set using the threshold algorithm described earlier with 100 samples, ϵ set to 0.05, and the conservative guess t_0 set to 2 seconds.

First, the robot went through a brief exploratory session alone in which it learned the t_1 for each sound. Unlike the

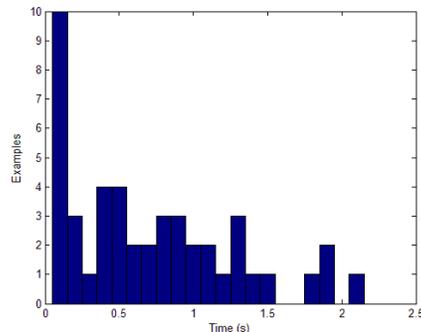


Fig. 5. The distribution of detected social responses for the robot. It appears to follow a negative exponential function, corresponding to the idea that humans generally respond socially as quickly as they can. Background Poisson events follow a negative exponential distribution with a longer tail, making the upper bound of the time window important and the lower bound mostly irrelevant in this domain.

motor-motion loop, this loop was highly reliable, with all but one of the 30 responses falling within 200 ms of the average. The algorithm then excluded that response from the window by virtue of falling within the ϵ lowest values. The long time required by QNX to interpret .wav files resulted in a t_1 time window of [1.3998s, 1.7017s] for the auditory domain.

Two independent subjects were then instructed to respond to these sounds as if Nico were actually engaging them in conversation. The exchanges continued for 45 utterances per subject. To avoid unnecessarily frustrating the subjects and reducing the social plausibility of the scenario, Nico did not wait for extended silence between utterances, but initiated sounds after only a second of detected silence. If the time delay detected for t_1 did not match fall within the expected time window, Nico ignored that data for future processing. (Since the transformed input was one-dimensional, there was no way in this simple space to keep track of what sound was whose if the two voices overlapped.) Otherwise, the detected sound was followed to its conclusion, and the first sound after that was assumed to be social for the purposes of learning. ϵ was again chosen to be 0.05 and t_0 was set to 2.

B. Results and Analysis

The resulting distribution of social response times is shown in Figure 5. Contrary to our expectations, the response times of socially interacting individuals are not normally distributed at all. Instead, the probability distribution function appears to be better modeled by a negative exponential.

In retrospect, there are good reasons to think that the social interaction distribution would have small or no time delays. While the turnaround time seems very fast compared to human processing speed, it is probably uncommon to have to wait until the very end of an utterance to decide what to say; we are good at predicting where a conversation is going, and the final syllable is usually unlikely to change our minds. Moreover, the decision of what to say next may have a roughly Poisson distribution

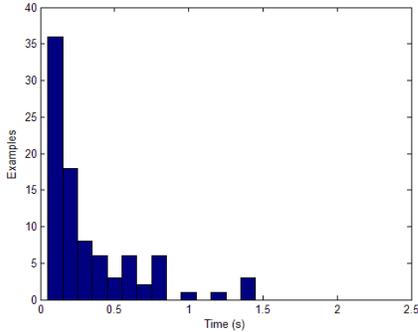


Fig. 6. Delays measured during conversations between the experimenter and the two social interaction subjects. Originally taken as potential data for a Bayesian estimator of the probability of social interaction, the data suggests that real social interactions have negative binomial or Poisson-like delay distributions.

over the length of the first speaker’s utterance; we would then expect a negative exponential to model the waiting time for this event. Assuming that the listener waits until the first speaker is done, this would result in a slightly top-heavy negative distribution.

If this model of “idea arrival” seems contrived, we can also argue from the conditions of the Poisson distribution. A Poisson distribution results from the assumption that an event is no more likely during one interval than another. Assuming that the probability that a person responds within a time interval Δt is non-increasing over time – that is, that the social partner does not become more likely to answer as time passes – then the probability density of the social response function is *at most* the negative exponential given by the Poisson process with λ_{max} , the probability density near time 0.

Both of these arguments assume that humans are good enough at predicting the end-content of speech directed at them to allow immediate responses. But what if this was only trivially true because of the crudeness of our robot’s social repertoire? Given a sequence of meaningless beeps and whistles, it was entirely possible, or even probable, that they began to ignore the “content” of Nico’s speech.

To provide suggestive (but not conclusive) evidence to rebut this claim, we present data originally collected to provide a “non-interacting” condition for any future Bayesian models. After testing in the interactive condition for each subject, each subject was engaged in conversation with the experimenter. Meanwhile, Nico was restarted in its learning mode, collecting the delays between the first string of detected sound after sending its vocalization command and the second. (The data presented here includes the cases where $t_1 < t_{1min}$, which Nico would normally reject.) Given the long t_1 delay for the vocalization, the measured delay was highly likely to be a delay in the conversation itself. Figure 6 shows a very clear negative exponential distribution for the combined data of the two subjects.

Just as in the case of self-recognition, a Bayesian decision algorithm could certainly be built to compare an event onset time to the social distribution and the environmental

distribution, to decide which was a more probable source for the event. However, such an algorithm would be useful only insofar as it discarded events that occurred too late – a function already performed by our time windows – and would be potentially harmful if it discarded social events that occurred within the acceptable interval.

VI. APPLICATIONS

One strong advantage of the method presented here for self-recognition is that it makes very few assumptions about the behavior, appearance, and kinematics of the robot. Instead of requiring a specific behavior for finding the self, as in [6], these self-recognition and social modules can easily piggyback on top of behaviors performed for some other purpose, or for no purpose except random exploration. The lack of assumptions about the robot’s appearance is also useful in case that appearance changes – for example, if the robot were to wear a glove. On the other hand, if the association of a particular color with the “self” property were to prove useful, this method can be used to bootstrap an expectation for that color.

But why not use a kinematic model? One argument for having a low-level detector for contingent motion is that it may help the robot quickly adapt to handling new tools and new situations. But the most intriguing answer may be the “mirror test” [19]. In brief, the ability to recognize oneself in the mirror has long been used (possibly without much good justification) as a gold standard for that vague notion called “self-awareness.” While recognizing the robot’s own static image could allow the robot to, for instance, identify a picture of itself, the primary distinction between that case and the mirror test – the one that would allow the robot to understand that the image was not merely a representation of itself, but an instantaneous image of itself *in the present* – is that of temporal contingency. If a robot can only ever learn one kinematic model for its arm, and rejects all other contingent events as “not self,” then it is doomed to fail the “mirror test.” (Note that we do not claim that Nico has “passed” the mirror test – currently a vague and subjective goal for a humanoid robot – but only that it satisfies a necessary condition for any reasonable definition of the test (Fig. 7).

In the social domain, the most obvious application of the social time window is to detect potential social partners that are not within the field of view. A potentially more interesting application is that this Boolean feature corresponds well to the semantic content of the word “you.” Elsewhere, we have shown that a machine observing others’ conversations can learn that “I” refers to the speaker and “you” refers to the listener [5]. To move to first-person usage, the robot will need properties grounded in its own experience to which those words can be anchored. We hypothesize that this is possible using the properties described in this paper.

VII. CONCLUSIONS

We have made novel contributions in three major areas with this paper.

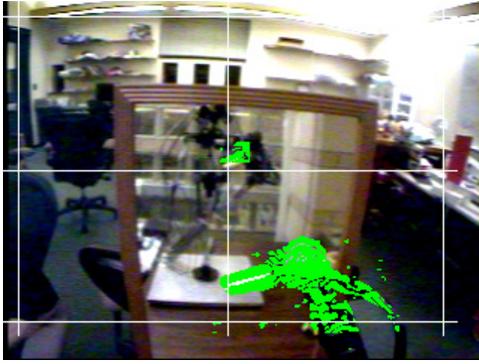


Fig. 7. The robot identifies the self-generated motion of its reflection. The shot is from one of Nico’s wide-angle cameras; Nico’s reflection (center) is moving its hand as Nico moves a real arm in the foreground (bottom right). Because the motion occurs within the learned time interval $[t_{1_{min}}, t_{1_{max}}]$, the motion is tagged as potentially self-generated (highlights).

First, we have provided a novel framework for learning contingent causal chains in noisy environments and arbitrary domains. Given a relatively innocuous set of assumptions about the environment and the nature of the sensory data, we showed that we can find a threshold that reduces the error in the possible time windows $[t_{1_{min}}, t_{1_{max}}]$ to an arbitrarily low value in linear time. We have also described briefly why we expect this number of samples to have an upper bound that is polynomial regardless of the distribution, using principles of PAC-learning [4].

Second, we have applied this framework to the problem of detecting the robot’s self-generated feedback in the environment. We have shown that the number of samples required to achieve an error rate ϵ on the true positives is much smaller in practice than the upper bound suggested by the VC dimension of the problem [4]. We also show that the distribution of the self-generated motion is quite clearly Gaussian, allowing us to substitute a probabilistic model for the Boolean attribute model given here. We nevertheless believe that real robotic systems should not use the learned time window as a sole determiner of self-identification, but as a filter to eliminate those actions that are not self-generated, or as a cue to investigate responses further.

Third, we have applied this framework to the problem of detecting contingent social responses, and have presented empirical data that suggests that these follow a negative exponential distribution similar to the waiting times for a Poisson process. We have argued that our time window process therefore captures the most critical distinction that a Bayesian decision process would make in deciding between the two distributions – namely, the time after which to give up on a social response.

But much work still lies ahead in this area. Our data on social delays is suggestive, not conclusive, and more psychological data is necessary to decide whether that distribution is well-modeled by a Poisson distribution. The utility of employing Bayesian methods in the time window domain has yet to be determined. Poisson threshold

learning has yet to see very many field tests, since its invention occurred after much of the data presented here was collected. We also have yet to prove how many samples are needed to bound the error of the $\lambda(T)$ estimation, and have not yet formally proven that the problem of learning real-valued intervals in our noisy environment is learnable in polynomial time. How reasonable is the assumption of uncorrelated event onset times in the real world? How should we sample the environment to best preserve this assumption, while still taking linear time to calculate the threshold? Are these time windows more useful as Boolean attributes or probabilities? And are these the correct properties to associate with the words “I” and “You”?

We and our robot still have much to learn.

ACKNOWLEDGMENTS

The authors would like to thank Liz Darbie, Frederick Shic, and Elizabeth Kim for their volunteering as test subjects, and the attendees of the AAAI 2005 Spring Symposium on Developmental Robotics for their helpful comments. Thanks also to Philip Michel, who collaborated with us on the early self-recognition work [15], and to Ganghua Sun and Andrew Lovett, who coded several of the modules mentioned in this paper.

REFERENCES

- [1] S. C. Johnson, “Whose gaze will infants follow? features that elicit gaze-following in 12-month-olds.” *Developmental Science*, vol. 1, pp. 233–238, 1998.
- [2] J. Nadel, I. Carchon, C. Kervella, D. Marcelli, and D. Réserbat-Plantey, “Expectancies for social contingency in 2-month-olds.” *Developmental Science*, vol. 2, no. 2, pp. 164–173, 1999.
- [3] B. J. Scholl and P. D. Tremoulet, “Perceptual causality and animacy,” *Trends in Cognitive Sciences*, vol. 4, no. 8, pp. 299–309, 2000.
- [4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, “Learnability and the vapnik-chervonenkis dimension,” *Journal of the Association for Computing Machinery*, vol. 36, no. 4, pp. 929–965, 1989.
- [5] K. Gold and B. Scassellati, “Using context and sensory data to learn first and second person pronouns,” in *Human-Robot Interaction*, under review.
- [6] P. Fitzpatrick, “From first contact to close encounters: a developmentally deep perceptual system for a humanoid robot,” Ph.D. dissertation, MIT, 2003.
- [7] P. R. Cohen, C. Sutton, and B. Burns, “Learning effects of robot actions using temporal associations,” in *Proceedings of the 2nd International Conference on Development and Learning*, 2002.
- [8] J. F. Allen, “An interval based representation of temporal knowledge,” in *International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., 1981, pp. 221–226.
- [9] Y. Yoshikawa, K. Hosoda, and M. Asada, “Cross-anchoring for binding tactile and visual sensations via unique association through self-perception,” in *Proceedings of the Fourth International Conference on Learning and Development*, 2004.
- [10] Y. Yoshikawa, Y. Tsuji, K. Hosoda, and M. Asada, “Is it my body? body extraction from uninterpreted sensory data based on the invariance of multiple sensory attributes,” in *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2004.
- [11] J. Movellan, “Infomax control as a model of real time behavior: Theory and application to the detection of social contingency,” UCSD MPLab, Tech. Rep., 2005.
- [12] A. Turing, “Computing machinery and intelligence,” *Mind*, vol. 49, pp. 433–460, 1950.
- [13] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, “Autonomous mental development by robots and animals,” *Science*, vol. 291, no. 5504, pp. 599–600, 2000.

- [14] J. Zlatev and C. Balkenius, "Introduction: Why "epigenetic robotics"?", in *First International Workshop on Epigenetic Robotics*, vol. 85, 2001, pp. 1–4.
- [15] P. Michel, K. Gold, and B. Scassellati, "Motion-based self-recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.
- [16] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
- [17] W. Feller, *An Introduction to Probability Theory and its Applications, Volume I*, 3rd ed. Wiley, 1968.
- [18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press, 2001.
- [19] G. Gallup, J. Anderson, and D. Shillito, "The mirror test," in *The Cognitive Animal: Empirical and Theoretical Perspectives on Animal Cognition*, M. Bekoff, C. Allen, and G. Burghardt, Eds. MIT Press, 2002.