Νάρκισσος
(Narcissus)

Kevin Godby and **Jesse Lane** and **Tanasha Taylor**
Iowa State University
Ames, Iowa 50014
{godbyk,jalane,tanashat}@iastate.edu

Abstract

In Greek mythology Narcissus was a youth who became infatuated with his own reflection after gazing into a pool of water. Unlike Narcissus, humans can recognize themselves when confronted with their own reflection. However, like Narcissus, many animals and most of today's robots cannot accomplish this simple feat of self recognition. We present here preliminary work in developing a system that we hope will eventually be able to distinguish between itself and others using vision and motion.

Introduction

As robots become more sophisticated and pervasive, they will be forced to operate in more dynamic and social environments. The ability to predict the behavior of other agents in these complex environments will become an advantage to such robots (and the people who interact with them). In order to develop a theory of mind to account for the intents, beliefs, and motivations of other individuals, a robot needs to be able to distinguish between another entity and itself. One proposed method of learning the difference between self and other is to use contingency, the time dependence of perception and action.

Watson (1994) suggested contingency as a method used by infants when learning to detect self. He outlined four general methods for detecting contingency: contiguity, temporal correlation, conditional probability, and causal implication. Below are brief descriptions of each method.

Contiguity The notion of contiguity simply states that "if a contingent stimulus occurs shortly in time following a behavior, then subsequent behavior of the subject will be altered as a function of the reinforcement value of the stimulus" (Watson 1994).

Temporal Correlation Contingency may be detected as the correlation between the rewards received shortly after performing a particular behavior. It is reinforced if lesser amounts of reward are received for lesser amounts of behavior.

Conditional Probability Conditional probability keeps track of instances in which the behavior occurs and the stimulus does not, versus instances when the stimulus occurs but the behavior does not.

Causal Implication Proposed by Bower (1989), causal implication suggests that the agent "observes the world with a natural inclination to formulate the potential cause–effect relations that would be *logically* consistent with the observed instances of behavior and stimulus events."

For our experiment, we chose to implement Watson's *conditional probability* method of contingency detection.

Previous Work

Research in self-recognition has primarily been an exercise in performing experiments that attempt to determine if a human in a particular developmental stage or an animal can recognize itself (Rochat & Striano 2002; Reiss & Marino 2001; Gallup 1970; Gallup, Anderson, & Shillito 2002). Probably the most famous experimental method for self-recognition is the mirror test (Gallup 1970; Gallup, Anderson, & Shillito 2002). In the mirror test the subject is marked (usually under anesthesia) in such a way that the mark is visible only when the subject looks in a mirror. If the subject attempts to explore the marking on the reflection in the mirror, then there is no self-recognition. On the other hand, if the subject explores the marked area on its own body, then there is a strong case for the ability to recognize self.

Recently, work has begun to develop robots which also have the ability to self-recognize. Our work was largely influenced by one such effort, the Nico project at Yale University (Michel, Gold, & Scassellati 2004; Gold & Scassellati 2005). The Nico group used contingency detection as a method for robotic self-detection. They chose to implement contingency by using contiguity. The robot Nico randomly moved its arm from pose to pose while remembering the minimum and maximum amount of time elapsed between a motor command and the perception of movement in the visual field. Then, in the detection stage, movements were classified as self if they fell within the learned window following a motor command. The results were quite good, with the robot detecting itself rather robustly; however, it suffered a good deal of degradation in performance when presented with an anticipatory distractor. We will revisit anticipatory and other distractors later in the paper.

Methodology

Our Robot Platform

Our robot Narcissus is a three-link, planar, robotic arm which is used in conjunction with a Firewire camera. The arm is constructed using three common hobby servos and three aluminum arm segments. We are using the Phidget servo controller which interfaces to a PC via a USB cable. For processing the video stream obtained from the camera we are using the OpenCV library developed by Intel and freely available at `http://sourceforge.net/projects/opencvlibrary`. Narcissus uses a single Firewire camera mounted above the arm to keep the arm within the visual field of the camera. By having a stationary camera, we avoid having to deal with egomotion. All vision processing and motor control is accomplished by a single 3 GHz Intel processor running the Linux operating system.

Watson's outline for the conditional probability method of calculating contingency presupposes that the agent can distinguish between separate objects in the environment as well as tell that a given time has passed between a self-initiated action and the perception of movement of an object. In order to accomplish this we used some standard methods to segment objects from the scene using color and motion.

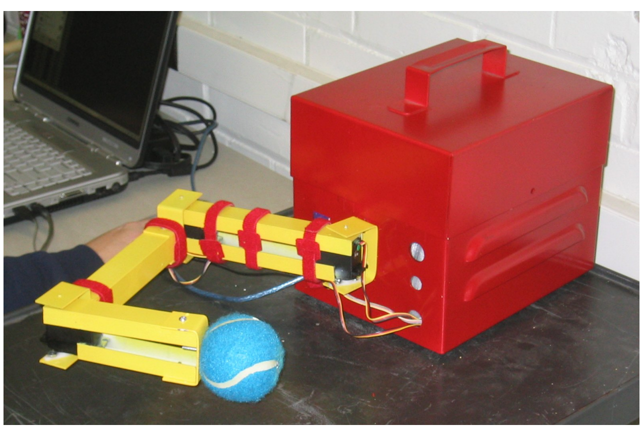

Figure 1: Narcissus

Visual System For the objects of interest we chose brightly colored tennis balls since they are relatively easy to segment from the scene using color. We are able to select an object to be tracked and the resulting hue histogram is used to calculate a back-projection, that is a mask where pixels with a high probability of being the desired color have a value close to 1.0 in the interval $[0, 1]$. It is important to note that the color segmentation is not used in any way to distinguish one object as self versus another—we are only using it to pick objects out of the background. Each object is given a unique identifier based on its color so that the number of perceptions of that particular object can be calculated.

We use the vision system to detect when a labeled object has moved and note the identifier of the moving object and the time that the movement occurred. Our method for detecting motion returns a list of bounding boxes for areas in the image that contain movement as determined by frame differencing. These bounding boxes can move a great deal from frame to frame as well as overlap considerably. Therefore we use the color data to determine which object a bounding box refers to and impose some very simple time continuity constraints so that multiple bounding boxes per moving object per frame are not labeled as separate perception events. We also track multiple bounding boxes for the same object through consecutive frames. We assume that if an object is detected moving again within a very short period of time, then we are dealing with the same movement event. In this work, we allowed a full two seconds to expire before considering movements to be considered a new perceptual event. This time interval large in our case because our robot took some time to settle into a stable pose due to the inexpensive hardware.

Also, we were unable to obtain a wider angle lens for our camera so some poses caused the end marker to temporarily leave the frame, although every pose ended with the marker within the scene.

The visual subsystem feeds into the training and detection modules as shown below.

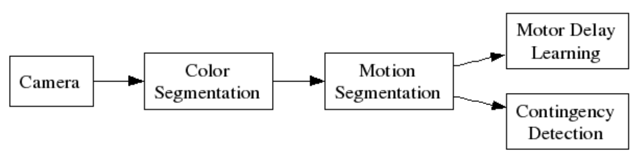

Figure 2: Narcissus' processing pipeline.

Conditional Probability

We use the conditional probability for detecting contingency because it should provide better protection against false positives and false negatives.

The *sufficiency index* is the probability of a stimulus given some specified time following a behavior:

$$P(S(t)|B(t')) = \frac{P(B(t')|S(t)) \cdot P(S(t))}{P(B(t'))} \quad (1)$$

where $B(t')$ is the behavior at time t', $S(t)$ is the stimulus at time t. The sufficiency index helps to protect against false negatives because a high sufficiency index occurs when a stimulus consistently follows a behavior. Thus, if you move your hand in front of your face, you *expect* to see your hand in front of your face.

The *necessity index* is the probability of a behavior given some time preceding a stimulus:

$$P(B(t')|S(t)) = \frac{P(S(t)|B(t')) \cdot P(B(t'))}{P(S(t))} \qquad (2)$$

again, where $B(t')$ is the behavior at time t' and $S(t)$ is the stimulus at time t. The necessity index helps to prevent false positives because a high necessity index is calculated when a behavior consistently precedes a stimulus. To use our example above, if you suddenly see your hand in front of your face, you expect that you performed the "move hand in front of face" action very recently.

By combining the two indices, a robust contingency method can be obtained. Watson proposes that if both of these indices are near or equal to 1.0, then contingency has been detected. In this work we were able to calculate these probabilities using information gained from our robot and the environment.

Training Action–Perception Correlation

Since Watson assumes that the agent engaged in contingency detection can determine intervals between action and perception in addition to distinguishing objects from each other, we had to train our robot to do the same. The robot has a small set of five possible poses that it can randomly cycle through. There is an extended pose as well as two symmetric Z-poses and two symmetric U-poses.

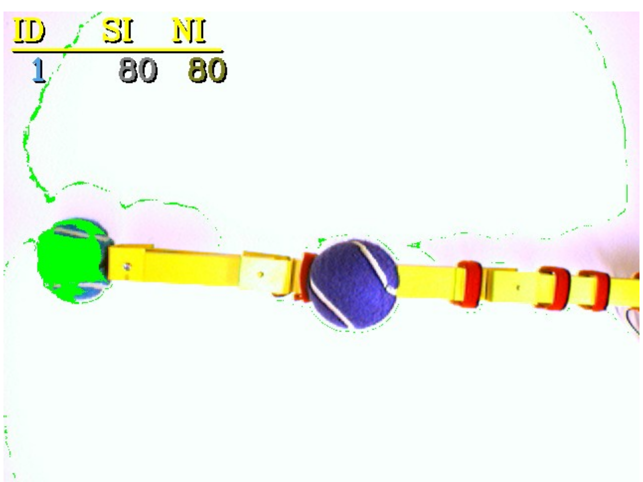

Figure 3: Extended pose.

During training the robot randomly moves from pose to pose recording the time that motor commands are issued and the time at which movement is first detected. There are no distractors in the training mode so that we may assume that all movements are made by the robot. We then calculate and record the mean and the standard deviation for the time intervals between action and perception to be used in the detection stage.

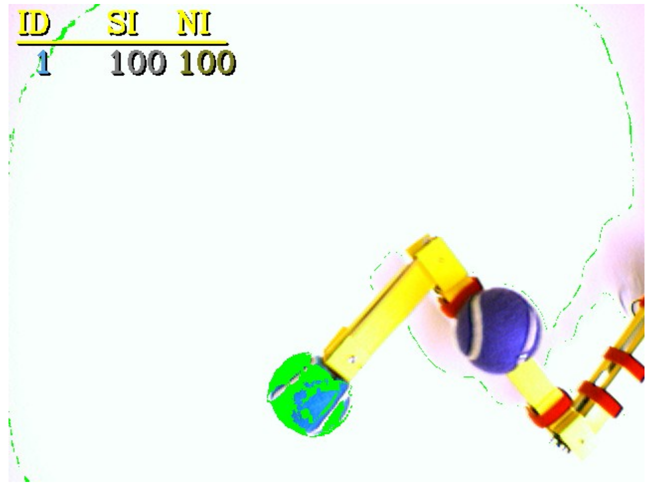

Figure 4: Z-pose.

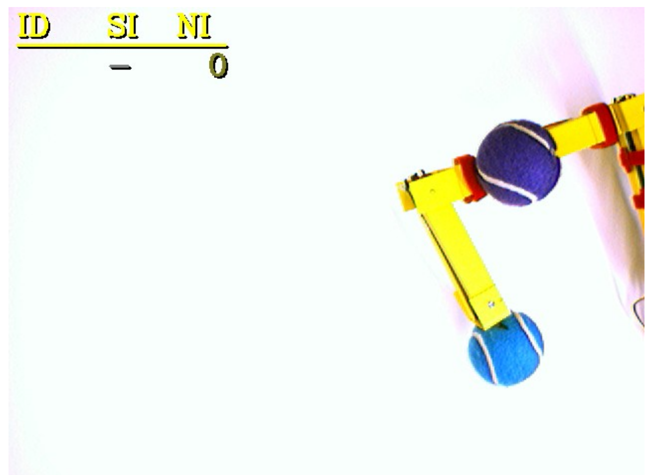

Figure 5: U-pose.

Detection of Self

As stated, one application of a contingency algorithm is the detection of self. Our robot Narcissus accomplishes this feat using the necessity and sufficiency indices mentioned above. During the detection stage, the robot can be moved manually (by pushing a button that initiates a motor command) or randomly as in the training stage. When an action is initiated, the time is recorded and the action event is put into a queue. Similarly, when a movement is perceived, the time is recorded along with the object identifier (obtained from the color data) and the perception is also placed in a queue. During each cycle of the robot's main execution loop, the queues are traversed and each action and perception are compared to see if they correlate. Correlation is determined by calculating the time interval between the action and perception and determining if it lies within three standard deviations from the mean calculated in the training stage.

$$|x - \mu| < 3\sigma \tag{3}$$

Under the assumption that the time intervals are Gaussian, then using this criteria for determining correlation, 99.73% of the time intervals that fall within the trained distribution. This criterion is only useful if the distribution is "tight" enough (*i.e.*, the standard deviation is not large). It was experimentally determined that the trained distribution indeed has these properties.

Each object tracked has a counter for the number of correlated action–perception pairs, as well as the total number of perceptions of that object. There is also a global counter for the number of actions initiated. Using these counters the sufficiency and necessity indices are calculated as follows.

$$SI_i = \frac{\#\text{ correlated action–perception pairs}}{\#\text{ total perceptions of object } i} \tag{4}$$

$$NI_i = \frac{\#\text{ correlated action–perception pairs}}{\#\text{ total actions initiated}} \tag{5}$$

Detection of self has been successful if both of the indices are above a threshold, 0.8 in this work. We then color the pixels of the objects in the image that have been determined to be self green as in figures 3 and 4.

Results

Our results are quite preliminary but demonstrate the feasibility of the method. We ran four experiments with all but the fourth experiment having three runs of approximately twenty arm pose changes per run. The four configurations used were one color (blue) with no distractors, two colors with no distractors, two colors with a semi-random distractor, and two colors with an anticipatory distractor.

The distractors are other colored objects that are moved in the visual frame to determine the performance of the algorithm when there are other moving objects that could be classified as self. The semi-random distractor was a red ball on the end of a pole that one of the investigators moved about in a random fashion. The anticipatory distractor was the same red ball, but this time the investigator tried to anticipate the movement of the robot and move the distractor at the same time. Below are graphs of the sufficiency and necessity indices over time during the first run of each experiment. The anticipatory distractor experiment only had one run due to time constraints.

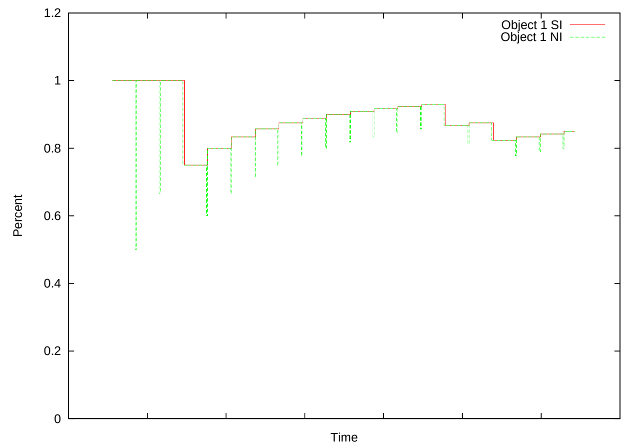

Figure 6: One color maker, no distractors. Run #1.

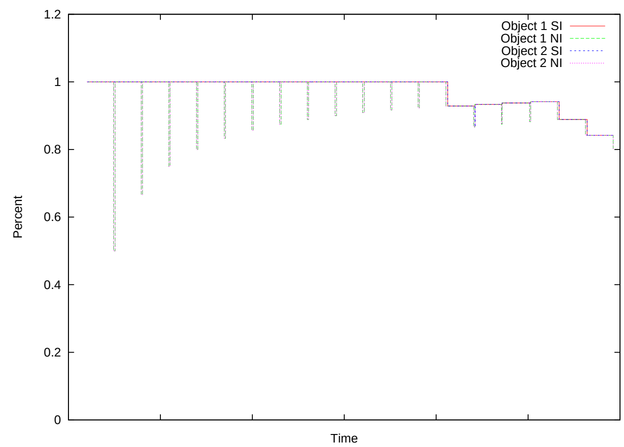

Figure 7: Two color markers, no distractors. Run #1.

Discussion

Our results were very satisfying. For instance, we see that in figure 6 the blue marker was identified as self for almost the entire run (note the temporary drops in the indices are due to the action counter being incremented and then waiting for resulting perceptions). We see that the time interval for the fourth pose change fell outside the correlation criteria so both indices dropped. However, for each subsequent correct correlation both indices increased until the blue marker was again detected as self.

In figure 7 we see the results for two colors (blue and purple) with no distractors. Again we see that both markers are successfully determined to be self for most of the run. At the end of the run, the performance began to drop, but we believe that this is a result of lag in the system due to high load rather than the algorithm. More work will have to be done to determine this for sure.

Figure 8 shows two colors (again blue and purple)

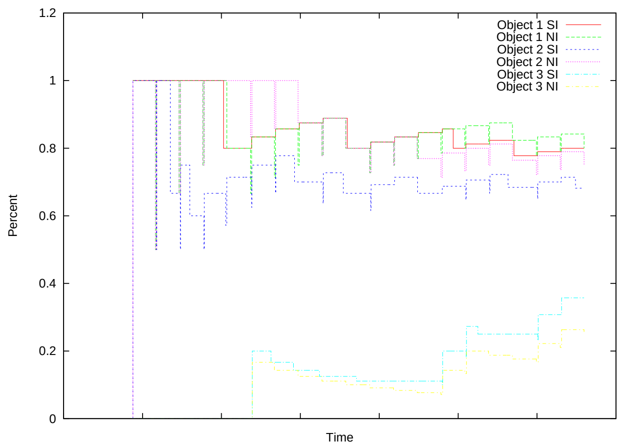

Figure 8: Two color markers, semi-random distractor. Run #1.

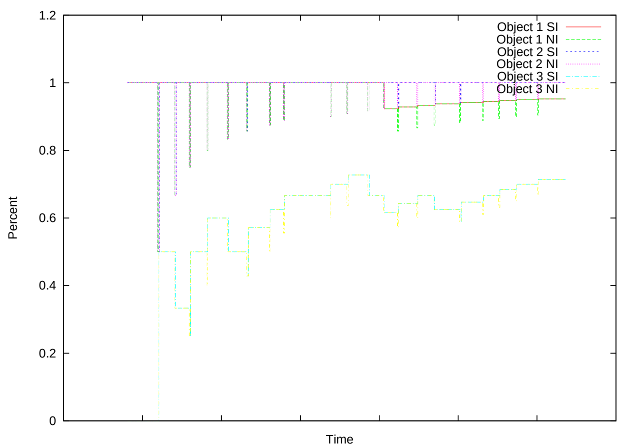

Figure 9: Two color markers, anticipatory distractors. Run #1.

along with a red semi-random distractor. There were some difficulties with this run and more data will have to be taken with this configuration. Again the performance of self-detection drops about half way through the run, although it seems to be recovering marginally toward the end. This run is important for two reasons. First, the performance losses do not appear to be directly related to the distractor. We see that the distractor periodically moves at the same time as the robot arm and thus has correlated action–perception pairs that get counted toward its sufficiency and necessity indices. However, the performance of the blue marker does not fall until a few pose changes later. It is suspected that the performance of the purple marker is so low because the small set of random poses were poorly chosen. As can be seen in figures 3 and 4, there are possible pose changes between a U-pose and a Z-pose that can leave the purple marker relatively unmoved. This in turn reduces the necessity index since an action was initiated but no movement was seen. The second reason this run was important is that we see the indices for the distractor decay as it fails to move at the same time as the arm. Therefore, an object that is not able to correlate itself strongly with the movements of the arm is not likely to be classified as self.

Finally, in figure 9 we see the only run done with two colors and an anticipatory distractor. Here the red distractor was purposely moved to attempt to correlate its movement with the initiated actions. We can see that the anticipation of the distractor was improving over the run with the distractor nearing the threshold to be considered self. However, the performance of the classifier on the two markers that were actually attached to the robot did not decline due to the actions of the distractor. These results are quite natural when we consider that the motor map in humans is quite plastic. It is useful for us to consider an object that moves when and where we move to be a temporary extension of self. We also expect that the classification of another object as self should decay as it no longer moves when we move.

As stated at the beginning of this section, these results are quite preliminary and are not conclusive. More data needs to be taken to determine more quantitatively the performance of this algorithm as well as its comparison to other contingency-based self-detection algorithms.

Future Work

There are many directions we would like to take this work, from just completing what has been started here to expanding the work to developing a theory of mind for robotic agents. Here we will cover the less ambitious ideas for future work.

We would like to move Narcissus to a higher-end robot platform. We chose the simple construction using hobby servos and aluminum because it was easy, cheap, and we already had knowledge of how to interface quickly with the hardware. However, the platform is not stable and not easily reproduced. Also, due to the load on the computer and the very simple velocity controller employed, the movements of the robot were very jerky. We have a number of commercial robot arms in our lab that we will be working on interfacing to and using in our follow-up experiments.

We would like to take a lot more data. The data taken here cannot be considered a large enough set to draw any conclusions. Each run of approximately twenty pose changes took five minutes to execute. Also, the data is very sensitive to initial conditions. For instance, a miscorrelation at the beginning of a run results in many more steps where the robot does not successfully classify objects as self. One way to mitigate this dependence on initial conditions is to do many long runs and then average the results.

Another direction that we would like to explore is to more rigorously test the distractor methods. Having a human investigator try to anticipate the movements of the robot or act semi-randomly is a very subjective way to test how the algorithm breaks. We would like to set up two identical robots and set them so that they each take up exactly half of the visual frame. Then we can have one robot initiate movements while the other moves in predetermined intervals before and after so that the effects of distraction can be more rigorously quantified.

The conditional probability method of contingency detection is only one of the four methods outlined by Watson. We have seen that previous work used the contiguity method. We would like to implement all four methods on one platform and compare them. Using the two-robot configuration along with all four methods should provide a comprehensive analysis of contingency detection.

Finally, we would like to explore more sophisticated methods of object permanence and detection. The methods used here were rather ad hoc and we believe that many of instances of poor performance were due to issues other than the algorithm. We find it also somewhat unsatisfying that we must segment objects from the background in order for our contingency algorithm to work. We would like to explore "fuzzier" implementations that work without the need to have discrete objects and action–perception pairs.

Acknowledgments

We would like to thank Dr. Alexander Stoytchev for the many inspirations developed in his developmental robotics course as well as the OpenCV vision code that he donated (little of which is in its original form).

References

Bower, T. G. R. 1989. *The rational infant: Learning in infancy*. New York: W. H. Freeman.

Gallup, G.; Anderson, J.; and Shillito, D. 2002. *The Cognitive Animal: Empirical and Theoretical Perspectives on Animal Cognition*. Cambridge, Mass: MIT Press.

Gallup, G. 1970. Chimpanzees: self-recognition. *Science* 167(914):86–87.

Gold, K., and Scassellati, B. 2005. Learning about the self and others through contingency. American Association for Artificial Intelligence.

Michel, P.; Gold, K.; and Scassellati, B. 2004. Motion-based robotic self-recognition. In *IEEE-RAS/RSJ International Conference on Humanoid Robots*. Sendai, Japan: IEEE.

Reiss, D., and Marino, L. 2001. Mirror self-recognition in the bottlenose dolphin: A case of cognitive convergence. In *Proceedings of the National Academy of Sciences (PNAS)*, volume 98, 5937–5942.

Rochat, P., and Striano, T. 2002. Who's in the mirror? self-other discrimination in specular images by four- and nine-month-old infants. *Child Development* 73(1):35–46.

Watson, J. S. 1994. Detection of self: The perfect algorithm. In Parker, S.; Mitchell, R.; and Boccia, M., eds., *Self-Awareness in Animals and Humans: Developmental Perspectives*. Cambridge University Press. chapter 8, 131–148.