

# Robot Self-Recognition Using Conditional Probability–Based Contingency

Kevin M. Godby and Jesse A. Lane

Iowa State University  
1620 Howe Hall  
Ames, Iowa 50011  
(515) 294-3438

{godbyk, jalane}@iastate.edu  
<http://www.vrac.iastate.edu/~godbyk/narcissus/>

## Introduction

As robots become more sophisticated and pervasive, they will be forced to operate in more dynamic and social environments. 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.

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

## Previous Work

Our work was largely influenced by the Nico project at Yale University (Gold & Scassellati 2005). The Nico group used contingency detection by contiguity as a method for robotic self-detection. The robot randomly moved its arm while recording the minimum and maximum amount of time elapsed between a motor command and the perception of resulting movement. Then, in the detection stage, movements were classified as self if they fell within the learned window following a motor command. The robot detected itself rather robustly but it suffered a good deal of degradation in performance when presented with an anticipatory distractor. Anticipatory distractors will be described later in this paper.

## Our Robot Platform

Our robot Narcissus (shown in figure 1) is a three-link, planar, robotic arm which is used in conjunction with a Firewire camera. Narcissus uses a single camera mounted above the arm to keep the arm within the visual field of the camera.

Watson’s outline for the conditional probability method of calculating contingency presupposes that (a) the agent can distinguish between separate objects in the environment and (b) tell that a given time has passed between a self-initiated

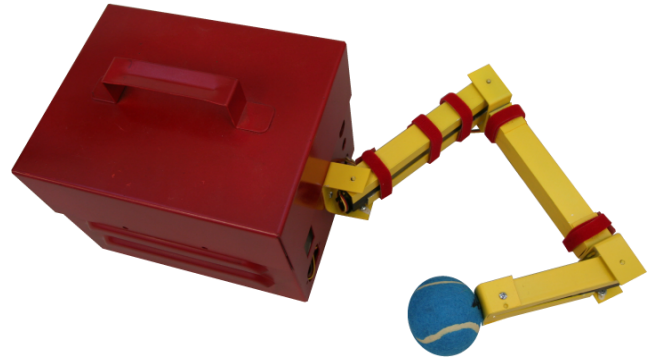


Figure 1: Narcissus is a three-link, planar, robotic arm. The blue tennis ball on the end of the arm is a color marker.

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.

## The Algorithm

The robot tracks objects and tries to determine if the objects are part of itself or not. *Objects* are colored markers in the scene. Some of these colored markers are attached to the robot, others are not attached. Unattached color markers are known as *distractors*. The robot arm can initiate an *action*—a command to move the arm to a new randomly-chosen pose. The robot records the timestamp the action was sent. When the robot detects that an object has started moving, it notes the timestamp and the object as a *perception*.

## Training

The training mode collects time intervals between actions and perceptions. In training mode, a new action is initiated every 15 seconds and the timestamp is recorded. All perceptions that occur between this action and the next are considered to originate from the action. A timestamp is recorded for the first perception that occurs after the action. The interval between the action and the perception is calculated and recorded.

Training mode is stopped after a predetermined number of actions. The standard deviation and mean are calculated

over the recorded action–perception time intervals to be used in the detection mode.

### Conditional Probability and Contingency Indices

Watson’s conditional probability method of detecting contingency uses two indices: the sufficiency index and the necessity index. An action and perception are *correlated* if they both occur within a short period of time.

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

$$SI = \frac{\text{number of correlated action–perception pairs}}{\text{number of perceptions of the object}} \quad (1)$$

For example, 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:

$$NI = \frac{\text{number of correlated action–perception pairs}}{\text{total number of actions initiated}} \quad (2)$$

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 detection method can be obtained. Watson proposes that if both of these indices are near or equal to 1.0, then contingency has been detected.

### Detection

Every 15 seconds an action is initiated and the timestamp is recorded in a list,  $A$ . After each video frame is received, any new perceptions are recorded in a list,  $P$ , and the sufficiency and necessity indices are calculated.

To calculate the indices, we compare each recorded action timestamp to each recorded perception timestamp. An action and perception are considered to be correlated if the time interval between the action and perception ( $x$ ) lies within three standard deviations ( $3\sigma$ ) from the mean ( $\mu$ ) calculated in the training stage.

$$|x - \mu| < 3\sigma \quad (3)$$

The correlated action–perception pair is added to a set,  $C$ .

For each object, the sufficiency index is the number of elements in  $C$  that correspond to the object divided by the total number of perceptions in  $P$ . The necessity index is the number of elements in  $C$  that correspond to the object divided by the total number of actions in  $A$ .

If the sufficiency and necessity indices are both greater than the threshold (0.8 in this work), then the object is labeled as self.

### Results

Narcissus performed three runs using two different-colored markers attached to the robot. Each run consisted of 20 actions. The results of these runs are shown in the table below.



Figure 2: The green pixels indicate which color markers the robot considers “self.” Narcissus recognizes the red anticipatory distractor as “other.”

The robot also completed a single run with two different-color markers attached to itself and one anticipatory distractor. The distractor was a third colored marker moved around by the experimenter. The experimenter tried to time the movements of the distractor with the movements of the robot in an attempt to get the robot to consider the distractor part of itself. Despite the presence of the distractor (see figure 2), the robot correctly labeled the two color markers attached to its body as “self” during all 20 poses.

Run	Object 1	Object 2
1	100%	100%
2	100%	90%
3	85%	85%

Table 1: Results of three runs with two color markers attached.

### Further Information

For further information about our robot Narcissus, the algorithm pseudo-code, source code, collected data, photos, MPEG movies, and an extended version of this paper, please visit our website at <http://www.vrac.iastate.edu/~godbyk/narcissus/>.

### Acknowledgments

We would like to thank Alex Stoytchev for his discussions and guidance and Jake Ingman for his photography skills.

### References

- [Gold & Scassellati 2005] Gold, K., and Scassellati, B. 2005. Learning about the self and others through contingency. American Association for Artificial Intelligence.
- [Watson 1994] 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.