Learning to Identify the Controllability of Containers and their Contents using an Extension of Self-Detection

Final Report for CprE 585X

Shane Griffith

Abstract - Robotic control of the contents of a container has challenged roboticists for a number of years. Traditionally, roboticists have created complex control algorithms that are engineered to demands of the system. It may be possible, however, for a robot to *learn* about the controllability of containers and their contents. Toward this, more work is needed to show that a robot can learn to identify what it can control.

This paper showed how the same self-detection algorithms that can identify self could be used to identify the controllability containers and their contents. In this experiment the robot interacted with 10 different containers and 5 different blocks. The robot observed the movement patterns of objects as it waved them around. Mutual information was calculated between pairs of objects that moved together, and the output was used to generate a sequence of controllability graphs. The result was a new feature representation for what the robot controls, which can be used to describe the evolution of an activity over time.

Introduction

Robotic control of the contents of containers has challenged roboticists for a number of years. The slosh-free control of liquids inside containers is a particularly active area of research [1][2][3]. Many papers have proposed complex control algorithms for addressing this problem. However, these algorithms are heavily engineered to the demands of a specific system; they are not designed to work under different operating conditions. For example, the parameters for the size of the container and the exact contents of the container have to be specified beforehand in order to compute an engineered solution. It is surprising that research toward identifying the controllability of arbitrarily sized containers and their contents is not yet forthcoming, given that a large amount of research has already addressed robotic control of many specific containers.

Clearly, an algorithm *does* exist for identifying the controllability of many different types of containers and their contents, because some children demonstrate it every chance they get (see Fig. 1). However, the algorithm that these children use remains unknown. What is known is that children undergo a long progression of learning in order to perfect their container–manipulation abilities. Infants play with containers incessantly in order to learn about them. It is through their play and exploration that they first learn what is controllable and then later gain control over a container's contents.

a) b)

Fig. 1: Cookie monsters. a) Retrieving a jar of cookies from the cupboard. b) Fetching a freshly poured glass of milk from the table.

Similarly, it may be possible for robots that play with and explore objects to learn to predict the controllability of containers and their contents. Indeed, a few papers have shown that a behavior–babbling robot can learn to identify itself as a controllable object [4][5][6][7]. The robots in these papers learned how to control themselves once they detected themselves. Some of the robots also learned about the controllability of different objects [6][7]. Little work, however, has addressed how self-detection methods can allow a robot about the controllability of containers and the objects that are inside them.

This paper described how a robot could begin to learn about the controllability of containers and their contents, which is based on an extension of self-detection. The robot (see the cover page) performed a sequence of exploratory behaviors on 10 containers and 5 blocks. The robot also observed a human imitate the same behaviors that it performed. Color tracking was used to identify the positions of the robot (or the human), the object, and the block during each trial. Mutual information was calculated between pairs of objects that moved together, and the output was used to generate a sequence of controllability graphs. The result was a new feature representation to identify what the robot controls, which can also be used to describe the evolution of controllability over the duration of an activity (see Fig. 2).

The new representation was able to identify similar controllability graphs for different containers, blocks, and agents that used them. The representation was also able to identify that the robot had indirect control of a block inside a container when the robot grasped the container. The representation persisted for small amounts of time when no movement occurred, which meant, for example, that the robot `knew' it could control objects at times when it was stationary. Furthermore, patterns of control that the robot had over the objects were shown to match patterns of control that the human had over the objects. The drawbacks of the representation are discussed and several directions for future work are given to expand on this approach.

Related Work

Developmental Psychology

Containers are one of the first abstract object categories that infants learn [8]. Their single distinguishing feature—their concave shape—makes them one of the simplest kinds of objects. However, it takes infants years of play and exploration before they master how to detect and use containers [9]. During this time, the progression of container learning is intermittent [10]. Infants can go months before they learn to attend to any additional variables that are important for containment.

Infants begin to understand how objects move around inside containers at about 2.5 months. Although they cannot yet manipulate containers, they have many opportunities during the day to observe their caretakers using them. Infants know that an object inside a container will move with the container when the container is moved [11]. However, they do not yet know how two objects can get in that spatial relationship with one another.

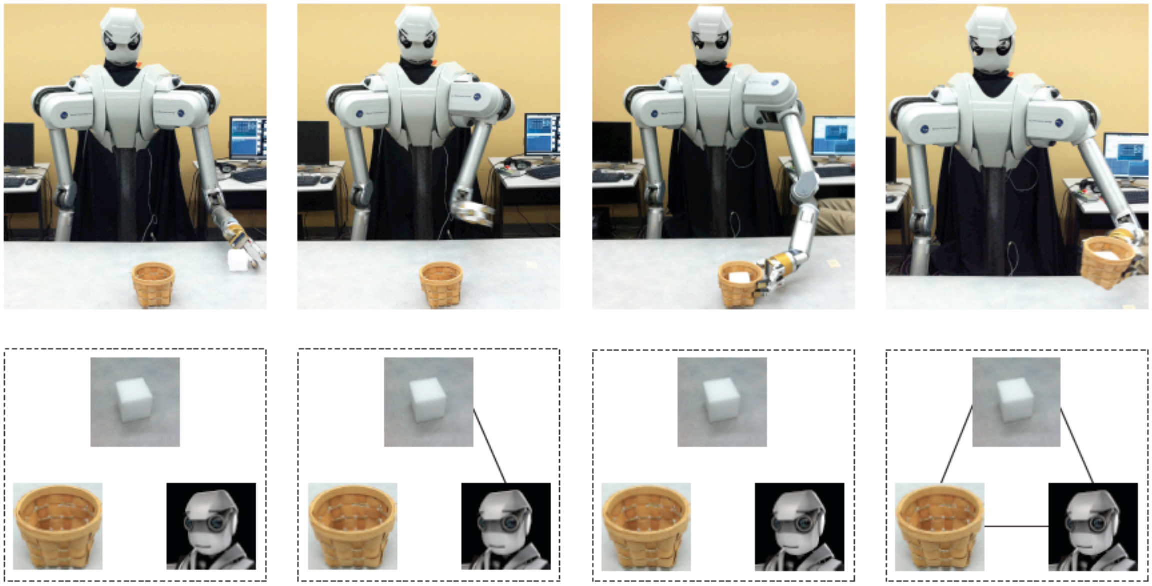

Fig. 2: An example of how the feature representation presented in this paper could be used to describe the evolution of an activity. In this example, the robot used a container to transport a block. The robot grasped a block, dropped it into a container, and then transported the container during the sequence. Edges between pairs of objects in the graphs show that the robot is confident that the two objects will move together. The graph also describes what the robot controls when edges are formed between objects and the robot. The sequence of controllability graphs is one way to describe the activity.

Infants begin to learn how their own body affects the movement of blocks inside containers at around 9 months of age. Nine-month-old infants are fascinated with inserting things into containers. First they try inserting their hands into containers [12]. Later they try inserting blocks into containers, which they subsequently shake. Their insert behavior peaks when infants are around 15 months old [12], which suggests that infants have spent months learning how their own movement affects the movement of objects inside containers.

In this paper the robot learned to identify the controllability of containers and their contents by exploring containers like infants do. The robot observed the movement patterns of objects as it waved the objects around. It also observed the movement patterns of objects as a human waved the objects around. The robot learned to identify the controllability of containers and their contents using an extension of self-detection.

<u>Robotics</u>

Little research has addressed the controllability of the contents of containers. Research that uses similar ideas to the ones proposed here have, however, shown how a robot can learn to detect and control the tip of tools, which is useful for keeping containers in an upright orientation. Kemp and Edsinger [13] showed how a robot could autonomously learn to detect and control the tip of a tool, which the robot used to control the open end of a bottle and a cup (the robot also interacted with a brush). The robot grasped each object and waved it around in order to identify the tip. The tip of the object was estimated from the points in the image sequence that had the highest optical flow. Their method works with arbitrarily sized containers (and other tools).

Also essential to container manipulation is the ability to control two objects at the same time. People routinely have to insert an object into a container (e.g., while preparing a drink that should be stirred). A robot can help people do this if it has the fine-tune control that is required to insert objects into containers. Edsinger and Kemp [14] showed how a robot could bimanually manipulate an object and a container in order to insert the object into the container. They presented a control system with which a robot manipulated five containers and five tools. The robot cued a person to hand it two objects, grasped both objects and then inserted one into the container, and afterword placed the cup on a shelf. When given a spoon, the robot stirred the contents of the cup before placing the cup on the shelf. The success of their control system derived from three themes of tool use: *cooperative manipulation*, *task relevant features*, and *let the body do the thinking*.

The marble maze game is one way to measure a robot's skill at controlling the movement of an object inside a container. The goal of a marble maze is to tilt a flat, square container in various directions in order to move a marble from one end of the maze to the other, while keeping the marble from falling into holes in the bottom of the container. Bentivegna et al. [15] created a learning framework that a humanoid robot used to learn to solve this task. The robot first learned what primitives (small units of behavior) it should use to order to solve the task by observing a human. As the robot practiced the game in more trials, it became better at choosing the right primitive to use in different situations and it learned how to modify the parameters of the primitive in order to achieve the desired outcome more often.

Before a robot can learn about the controllability of a container and its contents it first has to learn that containers with contents produce different outcomes than objects without contents. Griffith *et al.* [16] showed that an upper--torso robot was able to make this distinction by observing the movement patterns of a block and different objects as it interacted with them. The robot grasped a block, dropped it above an object, and pushed the object. The robot learned co-movement, separate movement, and noise outcomes from the different movement patterns that it observed. It perfectly separated containers from non-containers using the frequency with which different outcomes occurred with each object. Their follow up work [17] used a similar learning framework to show that a robot can categorize containers from non-containers by the different acoustic outcomes that are produced.

This paper asks how a robot can begin to learn to identify the controllability of containers and their contents. It is based on the assumption that robot learning should be grounded in its behaviors and the outcomes that they produce [18][19]. Without behavior–grounded knowledge, what the robot learns would not verifiable by the robot; it would have no way to test, verify, and correct its knowledge autonomously [18][19]. Thus, this paper used a developmental learning strategy, which is an important strategy toward creating a learning algorithm that can ultimately handle large changes in the robot's operating environment, and the containers that it has to control.

Experimental Setup

<u>Robot</u>

All experiments were performed with the upper-torso humanoid robot shown on the cover page. The robot was built with two 7-DOF Whole Arm Manipulators (WAMs) by Barrett Technology, each equipped with the Barrett Hand as its end effector. The WAMs are mounted in a configuration similar to that of human arms. They are controlled in real time from a Linux PC at 500 Hz over a CAN bus interface. The robot is also equipped with two cameras (Quickcams from Logitech). The cameras capture 640x480 color images at 30 fps. Rubber fingers were stretched over the each of the three fingers on the robot's left hand in order to augment the robot's grip of the objects that it manipulated.

<u>Objects</u>

A set of 10 objects was used for the experiments (see Fig. 3). The objects are containers in one orientation and become non-containers when flipped over. They were selected to have a variety of shapes and material properties. They were also all about the same height, which helped keep the blocks from bouncing out when dropped inside.

A set of 5 blocks was also used in the experiments (see Fig. 3). The 5 blocks varied in shape and material properties. The blocks were small enough to fit inside each of the containers. They were big enough to be grasped by the robot.

<u>Behaviors</u>

The robot performed a sequence of exploratory behaviors in order learn about the controllability of different objects and their contents (see Fig. 4). The robot first grasped a block, waved it around, and then dropped it above an object. Next, the robot grasped the object, waved it around, and then dropped it. Afterword, the robot watched as a human imitated the same sequence of behaviors that the robot performed. The robot performed one interaction trial for each block–object combination. Thus, a total of 20 objects * 5 blocks = 100 trials were performed by the robot. The robot performed a total of 5 *movements per wave behavior* * 2 *wave behaviors* *100 *trials* = 1000 waving motions. The sequence of behaviors performed during a trial are explained below:

> *Prepare Experiment* – Before each trial a block and an object were placed at marked locations on the table. Double--sided tape was used to mark the position of the block on the table and to keep it from rolling out of place once set there. The starting position of the block and the object were within the robot's visual field to ensure that they were visible during at least part of the trial.
>
> *Grasp Block* – The robot's first exploratory behavior was to grasp the block. The robot moved its arm into the field to the position of the block and then grasped it.

Fig. 3: The objects and the blocks used in the experiments. The set of 10 containers was selected to have a variety of shapes and material properties. All were roughly the same height. The non-containers are the same ten containers flipped upside down. The set of 5 blocks also varied in shape and material properties.

Wave Block – The robot moved the block closer to its camera and waved it back and forth after grasping it. Each back-and-forth motion of the robot consisted of a single jerky movement behavior. A total of 5 jerky movements were performed during this behavior.

Drop Block – The robot dropped the block above the object after waving the block around. The block was dropped in a uniform space of about 10 cm^2 above the object. The size of the drop zone was empirically set to be a little larger than the biggest container. The blocks fell into containers about 80% of the time.

Grasp Object – The robot grasped the object after dropping the block above it. In some trials one or more of the fingers failed to close during the behavior, which prevented the robot from having a firm grip of the object. In these trials, the experimenter manually reopened the hand, repositioned the object, and closed the hand before the robot moved on.

Wave Object – The robot brought the object closer to its camera and waved it back and forth after it grasped the object. The robot performed the same wave motions with the object that it performed with the block. This behavior was different, however, because sometimes the robot was holding a container and a block was inside it.

Drop Object – The robot dropped the object after waving it around. The object was dropped in place.

Observe the Human – The robot observed a human perform the same sequence of behaviors on the objects after it dropped the object. The human imitated the actions of the robot. First, the block and the object were placed at separate locations in the center of the table. After the experiment was prepared, the human grasped the block, waved it around, and then dropped it above the object. Next, the human grasped the object, waved it around, and then dropped it. The human usually dropped the block into the container during trials when the robot dropped the block into a container. The human usually missed when the robot missed.

Methodology

<u>Data Collection</u>

Data was captured from the robot's visual sensory modality for whole the duration of each trial. A sequence of 640x480 color images was captured at roughly 20 fps. Nanosecond timestamps were saved for each image. The times that the robot performed each movement during was also recorded.

<u>Movement Detection</u>

Objects in the robot's visual field were tracked using color in order to extract movement detection sequences. To get a position of the robot, an object, or a block from an image, a color range was specified for it and the pixels within that color range were used to compute an image mask. The largest connected component was found in the resulting mask and its center position was used as its position. Movement was detected when the position changed by more than a threshold d over a short temporal window [t', t"]. The values for d and [t', t"] were empirically determined. In this case, movement was detected when the position changed by more than $d = 10$ pixels between two consecutive

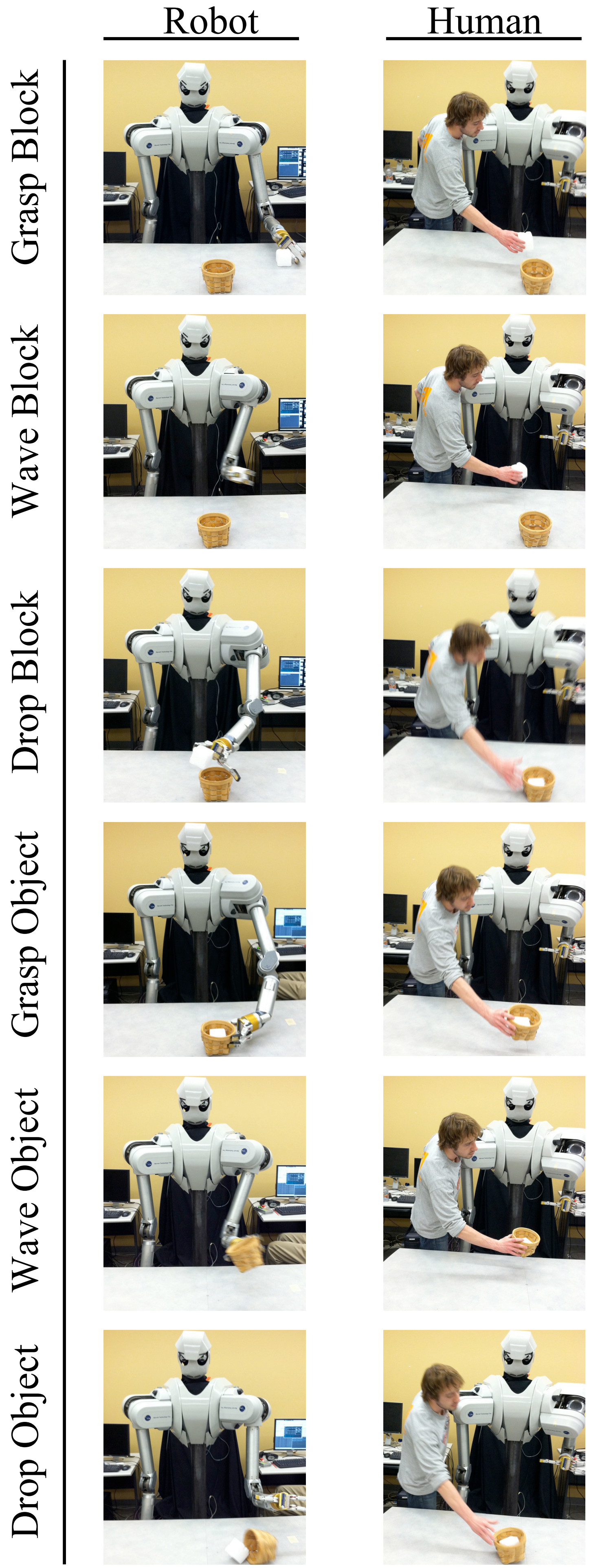

Fig. 4: The sequence of exploratory behaviors performed by the robot and the imitations of the robot performed by a human. See the text for more details.

frames. The result of this process was a movement detection sequence for the robot (or the human), an object, and a block for each trial (see Fig. 5).

The output of the movement detection step for trial T is a binary sequence $B_T = B_T^1, \ldots, B_T^l$, whose binary values specify movement/no-movement. Binary sequences were recorded for the robot, an object, and a block for the duration of time that the robot interacted with the objects. Binary sequences were also recorded for the human, an object, and a block for the duration of time that the robot observed the human interact with the objects.

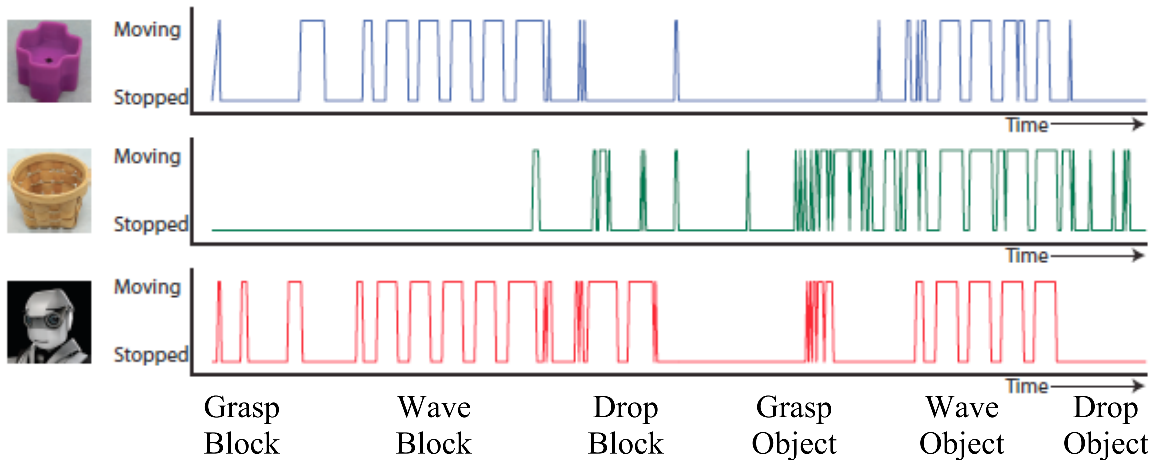

Fig. 5: Movement detected for the purple block, the wicker basket, and the robot as the robot interacted with the block and the object during one trial. The sequence shows that the block moved with the container when the container was waved around. The block was inside the container during this part of the trial. A separate movement detection sequence for the purple block, the wicker basket, and the human was generated for the part of the trial when the human imitated the behaviors of the robot.

<u>Identifying the Controllability of Objects using an Extension of Self-Detection</u>

The robot identified the controllability of objects using an extension of self-detection. By measuring the uncertainty in the delay between the times that the robot issued a movement command and the times that a pixel moved in the visual field, the robot can identify what pixels belong to its body and what pixels do not. Similarly, by measuring the mutual information with which two objects move together in the robot's visual field, the robot can identify what objects affect each other's movement and what objects do not. The robot can calculate the mutual information between its own movement and the movement of objects in order to identify what objects it controls. A simple graph can be used to represent these co-movement and controllability relationships.

A *controllability graph* was used to represent what the robot controlled (see Fig. 6). Nodes in the graph are the objects tracked by the robot. An edge is created between two nodes in the graph when the robot is confident that the movement of one of the objects

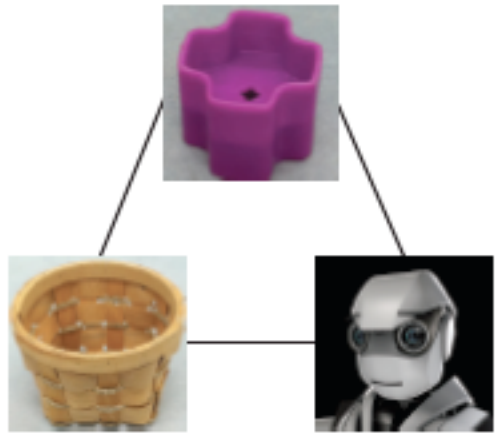

Fig. 6: A controllability graph. Nodes represent the purple block, the wicker basket, and the robot. Edges between pairs of nodes show that they move together. The edge between the robot and the block and the edge between the robot and the wicker basket show that the robot controls them. The robot was able to create this graph after it dropped a block into a container and then moved the container.

provides information about the movement of the other object, i.e., that they move together, and vice versa. An edge between the robot and an object means that the robot has control of the object. A sequence of controllability graphs illustrates the evolution of what the robot can control as it interacts with objects over time.

The mutual information and the confidence values were computed from the movement detection data for each pair of objects and used to generate controllability graphs, as shown in Figs. 7 and 8. Take the block, B, and the container, C, for instance. The relationship between the movement of a block, quantified by a variable B, and the movement of a container, quantified by a variable C, is used for co-movement detection. The mutual information $I(B; C)$ quantifies the amount of information shared between the two variables. The criterion for co-movement:

Co-Movement if $I(B; C) > 0$
Separate Movement if $I(B; C) = 0$

Since the precise distribution of the variables is not known but only the data is available, an edge between objects is considered to be a statistical hypothesis test where the null hypothesis is *Separate Movement*. In practice, exponential convergence is observed for the p-value due to statistical properties of entropy estimation.

The mutual information, I(B;C), and the confidence value, Pr(DATA|I=0), were computed from the movement detection data, $B_{TB} = B_{TB}^1, ..., B_{TB}^l$ and $B_{TC} = B_{TC}^1, ..., B_{TC}^l$. A histogram of movement detection data was generated over a three-second-long sliding window for the sequences $B_{TX} = B_{TX}^1, ..., B_{TX}^l$ and $B_{TY} = B_{TY}^1, ..., B_{TY}^l$. From this the mutual information, I(B;C), can be computed using the equations below. The confidence value Pr(DATA|I=0) is computed from the mutual information, I(B;C) using the X^2 distribution. An edge is defined between the block and the object in the graph when 1 - Pr(DATA|I=0) >= 0.99.

Container \ Block	0	1
0	N_{00}	N_{01}
1	N_{10}	N_{11}

N_{00} = The number of times that neither object moved.

N_{01} = The number of times that only the block moved.

N_{10} = The number of times that only the container moved.

N_{11} = The number of times that both objects moved.

The probability estimates for the different combinations of movement detection data.

$$\hat{p}_{00} = \frac{N_{00}}{N}$$
$$\hat{p}_{01} = \frac{N_{01}}{N}$$
$$\hat{p}_{10} = \frac{N_{10}}{N}$$
$$\hat{p}_{11} = \frac{N_{11}}{N}$$

The marginal probability distributions for the block and the container.

$$\hat{p}_B = \frac{N_{01}+N_{11}}{N}$$
$$\hat{p}_{\bar{B}} = \frac{N_{00}+N_{10}}{N}$$
$$\hat{p}_C = \frac{N_{10}+N_{11}}{N}$$
$$\hat{p}_{\bar{C}} = \frac{N_{00}+N_{01}}{N}$$

The mutual information between the block and the container.

$$\hat{I}(B;C) = \hat{H}(B) + \hat{H}(C) - \hat{H}(B,C)$$

The estimated Shannon entropies.

$$\hat{H}(B) = -(\hat{p}_B log_2 \hat{p}_B + \hat{p}_{\bar{B}} log_2 \hat{p}_{\bar{B}})$$
$$\hat{H}(C) = -(\hat{p}_C log_2 \hat{p}_C + \hat{p}_{\bar{C}} log_2 \hat{p}_{\bar{C}})$$
$$\hat{H}(B,C) = -(\hat{p}_{00} log_2 \hat{p}_{00} + \hat{p}_{01} log_2 \hat{p}_{01} + \hat{p}_{10} log_2 \hat{p}_{10} + \hat{p}_{11} log_2 \hat{p}_{11})$$

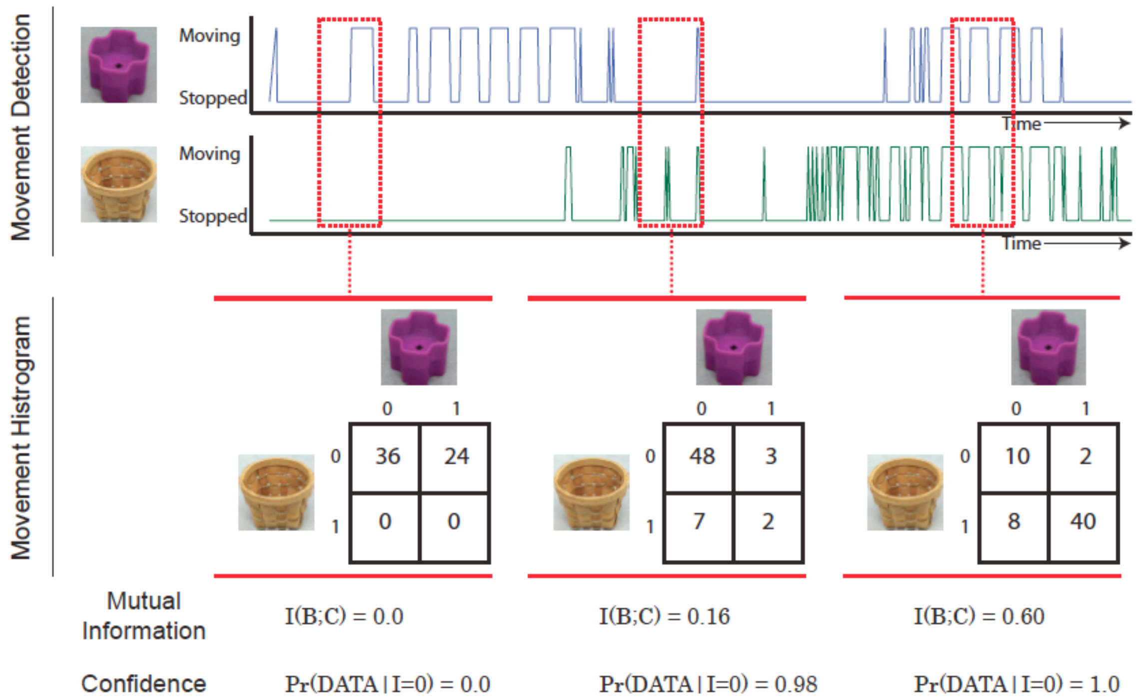

Fig. 7: The process of extracting mutual information and confidence values for a pair of objects. The movement detection data within a three-second-long sliding window is used to compute a histogram, from which mutual information and confidence values were calculated. Three different histograms are shown. The first histogram captures movement data from a snippet of time when the robot was waving the block around and the object was in place. The data in the second histogram is for when the robot dropped the block. The data in the third histogram is for when the robot waved the container around and the block was inside it. The mutual information and confidence values increased as the block and container moved together more often.

Results

The results of identifying the controllability of containers and their contents are shown in Fig. 9. The figure shows that the representation used in this was paper was able to identify what the robot controlled. Similar sequences of controllability graphs were generated for different blocks, objects, and agents. The sequence of controllability graphs for containers for many of the trials resembled the two examples depicted in the figure (with varying degrees of noise). The same was true for trials with the non-containers.

The representation was able to show that the robot had indirect control of the block. Although the block was inside the container during some trials, the robot's waving motion with the container affected the movement of the block. The controllability graph easily captured this relationship. The indirect control of the block was similar to the direct control of the block, which the robot had when it grasped the block. However, the

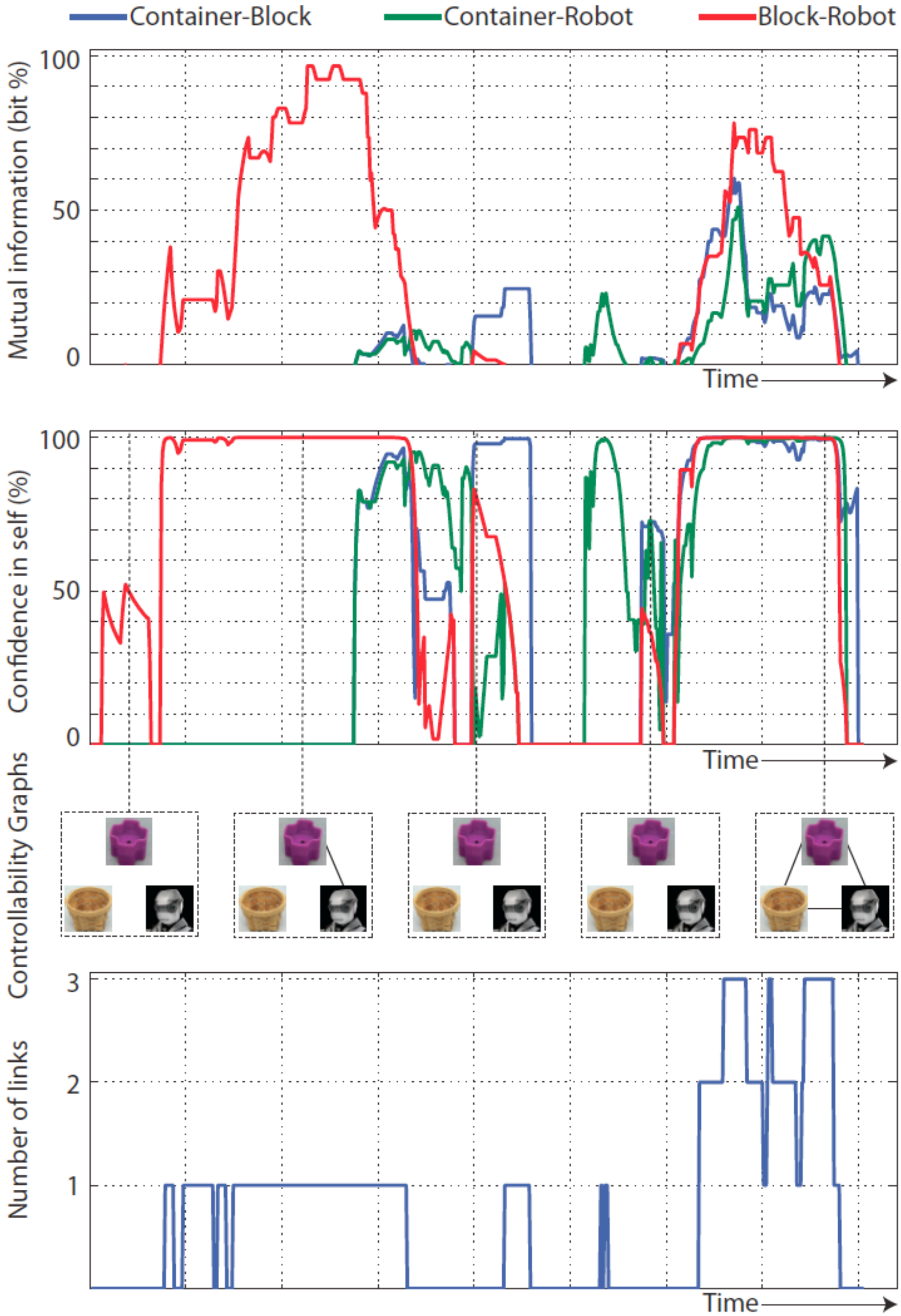

Fig. 8: Mutual information, confidence, and controllability graphs for one interaction trial performed by the robot. The graphs are the end result of the movement sequence analysis (see Fig. 5) on each of the three pairs of objects: the container—block pair; the container—robot pair, and the block—robot pair. A link between a pair of objects is created if the confidence value for that pair is 1. The result is a sequence of controllability graphs, which show what the robot controls at different times during the trial. A similar sequence of controllability graphs can be created for the human. The last graph shows the number of links in the controllability graphs, and is used for comparison purposes only.

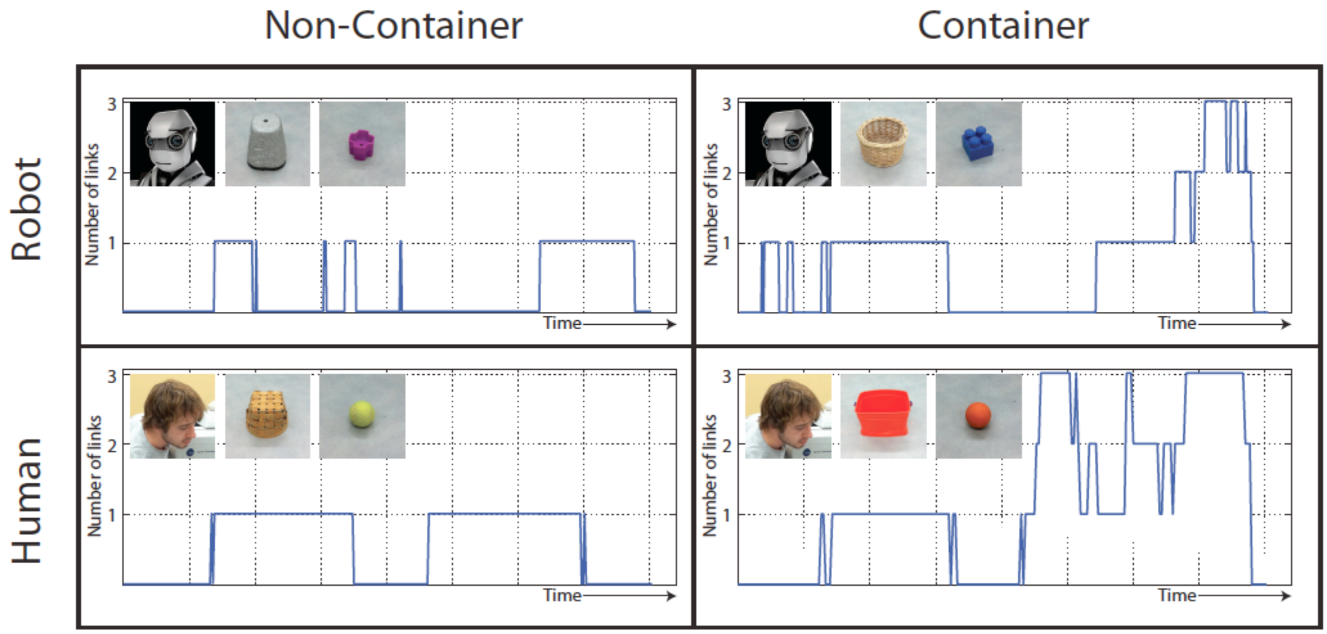

Fig. 9: A comparison of sequences of controllability for different agents, objects, and blocks. The two sequences generated with the non-containers, and the two sequences generated with the containers, are similar even though the agents, the objects, and the blocks used to create them are different. All four sequences are noisy due to errors in the tracking data.

representation predicted that the robot had control of the block faster when the block was directly grasped, as opposed to when it was inside a container.

Another advantage of this representation is that the links between pairs of objects in the sequence of controllability graphs persisted for a short time even when the objects were not moving (see Fig. 10). A simpler model of using co-movement/separate movement features between objects would only capture the functional relationship between objects from frame to frame. However, the representation described in this paper persisted for up to three seconds during periods of inactivity, because the sliding window was three seconds long. For example, the controllability graph remained steady in most cases when the robot came to short stops as it moved back and forth during the *wave* behavior (the loss of links in the controllability graphs during the behavior were caused by errors in tracking; the lack of movement had a smaller effect). Thus, in this feature representation it took the robot three seconds of inactivity to forget that a link exists between two objects.

The three second window also meant that it took the robot three seconds to form a link. For example, most of the time a link was not formed between a block and a container immediately after the block was dropped into the container. The block and the container simply did not move around long enough for a link to form between them in the controllability graph. This meant that the robot is not able to identify the link of co-movement between the block and the container after the drop block behavior. However, it may be possible for a robot to learn to predict this controllability. That is, once a block is dropped inside a container, it may be possible for the robot to predict a link in the controllability graph.

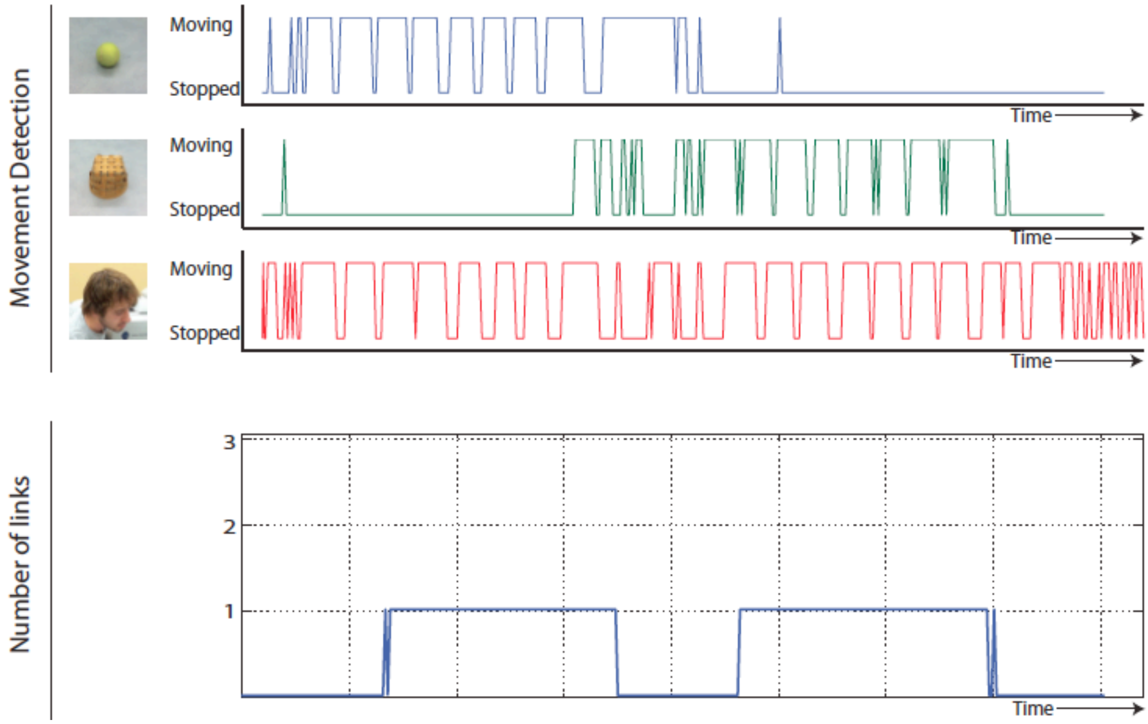

Fig. 10: The movement detection data and the sequence of the number of links between the objects for one trial performed by a human. The comparison shows that the controllability graph representation persists during short periods of time when the robot is stationary. The representation can persist for up to three seconds. This means that the noise in the controllability graph representation is due to tracking errors.

Some behaviors changed what the robot was able to control and some behaviors did not, which is implicitly shown in the graphs. The controllability graph was always affected when the robot opened or closed its hand. That is, the behaviors performed by the hand almost always signaled a transition in what the robot controlled (e.g., dropping the object into a container or grasping a container). Changing direction of movement during the wave behavior was a less reliable indicator that links would appear or disappear in the robot's controllability graph. This information is useful for learning what behaviors the robot has to perform in order to change what it can control. The robot could also learn that the controllability graphs will remain the same when it does not perform one of these behaviors, even past the three-second window of inactivity.

The results also show that a lot of noise was present in the controllability representation. The number of links in the graph frequently fluctuated from 1 to 3 as the during the wave object behavior. The strongest link was between the robot and the block. The weakest links were between the object and the robot and between the object and the block. This suggests that the robot lost control of the container during the behavior. However, the robot actually did have good control of the object. Tracking errors caused the reduced accuracy. The block usually occluded a large portion of the container when it was inside, which caused errors in tracking the object. These errors were present even when color-based tracking readily identified regions of the image that belonged to the object.

The overall quality of the representation was most negatively affected by the use of color tracking. The white block was frequently mistaken for the shiny metal robot. Some of the wicker baskets had the same RGB color range as the human arm. Other combinations of object and blocks were also difficult to tell apart. The object tracking method was usually too basic to support the representation described in this paper.

The controllability graphs for each trial would become more stable if more sophisticated tracking methods were used. One way to improve the results is to perform color tracking using a different color space, e.g., HSV, which is more robust to changes in lighting. A different object-tracking method works by identifying clusters of texture-based features of the objects. Alternatively, an even more sophisticated tracker could use SIFT features. Any improvements to the tracking model would probably significantly improve the controllability graph representation extracted for many of the trials.

Conclusion and Future Work

This paper presented a novel approach for learning to identify objects that the robot can control. Learning to identify what the robot controls derives from an extension of an uncertainty-driven self-detection algorithm. The approach enabled the robot to identify the controllability of containers and their contents. The method produced consistent results when tested with different containers, blocks, and agents. The method was also resilient to small periods of time when no movement occurred.

This paper also described a novel method for representing an activity. A *controllability graph* was defined, whose nodes and edges describe co-movement between objects and what the robot can control. The pattern of controllability for a sequence of behaviors was described using a sequence of controllability graphs. The representation for the robot's activities and the representation extracted for the human's activities were qualitatively similar.

One direction for future work is to extend the current approach in order to make the controllability graph more scalable to larger sets of objects. In the current implementation, mutual information is computed between every pair of objects that the robot is tracking. That is, computation time grows exponentially for every object that the robot is tracking. The scalability can be improved, however, by only computing the mutual information between pairs of objects that are adjacent to each other.

Another direction for future work is show how the robot can use the controllability graph representation to learn how to can gain control the contents of containers. The proposed research only seeks to show that the robot can identify what objects are controllable based on their co-movement patterns with other objects. A robot could use what it learned in this study to start to try and improve its control over the blocks inside containers. If successful, the robot could be evaluated against other approaches (e.g., [15]) using the marble maze game.

Future work could also explore the robot's ability to autonomously learn several parameters that determine the controllability of blocks inside containers. For example, how tightly fitting the block is with the container determines the amount of controllability the robot will have over the block. Perhaps the robot can learn to predict controllability based on the size of the block. Another variable tat the robot could learn is controllability vs. jerkiness during movements. The more jerky the robot's movement, the less the block will appear to be moving with the container and the robot. A robot could learn that, given a block inside a container, the best strategy for manipulation is the strategy that has the least jerky movements. If a robot can learn about these variables using our algorithm, then it would be useful for solving the slosh-free motion of containers task, even with arbitrarily sized containers.

Acknowledgements

I must acknowledge Vlad Sukhoy for his contribution of the uncertainty driven self-detection methodology and algorithms, which I used to generate the controllability graphs. He was also instrumental in setting up and running the experiments.

References

[1] J. Feddema, C. Dohrmann, G. Parker, R. Robinett, V. Romero, and D. Schmitt. "Control for slosh-free motion of an open container." IEEE Control Systems Magazine. v. 17. no. 1. pp. 29-36. 1997.

[2] V.J. Romero and M.S. Ingber, "A Numerical Model for 2-D Sloshing of Pseudo-Viscous Liquids in Horizontally Accelerated Rectangular Containers", *Proc. 17th International Conference on Boundary Elements*, pp. 1995.

[3] Tzamtzi, M.P.; Koumboulis, F.N.; Kouvakas, N.D.; , "A two stage robot control for liquid transfer," *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on* , vol., no., pp.1324-1333, 25-28 Sept. 2007

[4] K. Gold and B. Scassellati. "Learning about the self and others through contingency." 2005 AAAI Spring Symposium "Developmental Robotics." Stanford, CA. Mar. 2005.

[5] Stoytchev, A., "Self-Detection in Robots: A Method Based on Detecting Temporal Contingencies", Robotica, volume 29, pp. 1-21, 2011. Cambridge University Press 2011.

[6] Giorgio Metta and Paul Fitzpatrick. Early integration of vision and manipulation. Adaptive Behavior, 11:2, pp. 109-128, June 2003.

[7] Kemp, Charles C. and Edsinger, Aaron. "What Can I Control?: The Development of Visual Categories for a Robot's Body and the World that it Influences". Proceedings of the Fifth International Conference on Development and Learning, Special Session on Autonomous Mental Development. 2006.

[8] M. Casasola, L. Cohen, and E. Chiarello. "Six-month-old infants' categorization of containment spatial relations." Child Development. vol. 74. no. 3. pp. 679-693. 2003.

[9] Crain, W. Theories of development: Concepts and applications. Chapter: Piaget's Cognitive-Developmental Theory, pp. 110-146. Prentice Hall, Upper Saddle Ricer, NJ. 2009.

[10] R. Baillargeon. "How do infants learn about the physical world?" Current Direction in Psychological Science. vol. 3. no. 5. pp. 133-140. 1994.

[11] Hespos, S. and Baillargeon, R. "Reasoning about containment events in very young infants." *Cognition*. v. 78. no. 3. pp. 207-245. 2001.

[12] R. Largo and J. Howard. "Developmental progression in play behavior of children between nine and thirty months: II: Spontaneous play and language development." Developmental Medicine and Child Neurology. vol. 21. no. 4. pp. 492-503. 1979.

[13] Kemp, Charles C. and Edsinger, Aaron. "Robot Manipulation of Human Tools: Autonomous Detection and Control of Task Relevant Features". Proceedings of the Fifth International Conference on Development and Learning, Special Session on Classifying Activities in Manual Tasks. 2006.

[14] Edsinger, Aaron and Kemp, Charles. "Two Arms are Better than One: Designing Robots that Assist People in Everyday Manual Tasks". Proceedings of the IEEE International Conference on Advanced Robotics (ICAR). 2007.

[15] D. Bentivegna, C. Atkeson, and G. Cheng. "Learning Tasks from observation and practice." *Robotics and Autonomous Systems.* v. 47. no. 2. pp. 163-169. 2004.

[16] Griffith, S., Sinapov, J., Miller, M., and Stoytchev, A. "Toward interactive learning of object categories by a robot: A case study with container and non-container objects." In Proc. of the 8th IEEE Intl. Conf. on Development and Learning (ICDL), Shanghai, China. 2009.

[17] Griffith, S., Sinapov, J., Sukhoy, V., and Stoytchev, A. "How to separate containers from non-containers? a behavior-grounded approach to acoustic object categorization." In Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), Anchorage, AK. 2010.

[18] R. Sutton. "Verification, the Key to AI." on-line essay. Available at: http://www.cs.ualberta.ca/ sutton/IncIdeas/KeytoAI.html

[19] A. Stoytchev, “Some basic principles of developmental robotics,” IEEE Transactions on Autonomous Mental Development, vol. 1, no. 2, pp. 122–130. 2009.