Separating containers from non-containers:

A framework for learning behavior–grounded object categories

by

Shane Griffith

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-Majors: Computer Engineering and Human Computer Interaction

Program of Study Committee:
Alexander Stoytchev, Major Professor
Nicola Elia
Manimaran Govindarasu

Iowa State University

Ames, Iowa

2011

DEDICATION

To my Dad, my Mom, and my siblings Steve, Nathan, Amanda, and Alex.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I must acknowledge my advisor Alexander Stoytchev and my colleagues Jivko Sinapov, Vladimir Sukhoy, and Matt Miller from the Developmental Robotics Lab for their contributions to the research presented in this thesis. They have helped me brainstorm ideas for experiments, conduct experiments, write code to analyze data, and publish research papers. I must also acknowledge the National Science Foundation, the Department of Electrical and Computer Engineering at Iowa State University, and the Virtual Reality and Applications Center at Iowa State University for providing me with generous financial support. The research presented in this thesis was partially supported by the National Science Foundation Graduate Research Fellowship (NSF Grant No. 0751279).

ABSTRACT

Many tasks that humans perform on a daily basis require the use of a container. For example, tool boxes are used to store tools, carafes are used to serve beverages, and hampers are used to collect dirty clothes. One long term goal for the field of robotics is to create robots that can help people perform similar tasks. Yet, robots currently lack the ability to detect and use most containers. In order for a robot to have these capabilities, it must first form an object category for containers.

This thesis describes a computational framework for learning a behavior–grounded object category for containers. The framework was motivated by the developmental progression of container learning in humans. The robot learns the category representation by interacting with objects and observing the resulting outcomes. It also learns a visual model for containers using the category labels from its behavior–grounded object category. This allows the robot to identify the category of a novel object using either interaction or passive observation.

There are two main contributions of this thesis. The first contribution is the new behavior–grounded computational framework for learning object categories. The second contribution is that the visual model of an object category is acquired in the last step of this learning framework, after the robot has interacted with the objects. This is contrary to traditional approaches to object category learning, in which the visual model is learned first before the robot has even had the chance to touch the object. Because the visual model is learned in the last step, the robot can ascribe to a novel object the functional properties of its visually identified object category.

CHAPTER 1. INTRODUCTION

1.1 The Use of Containers in Animals and Robots

The ability to categorize objects is a key milestone in the development of many animals (Rosch, 1978). One basic object category that many different animals can identify is that of containers. Containers are characterized by a distinct set of perceptual and functional features, which also make them identifiable. For example, containers are concave, which allows them to hold other objects. Also, an object placed inside a container typically moves with the container when the container is moved. Animals can identify these and other properties of containers by interacting with different objects and observing the resulting outcomes.

Some animals can distinguish between containers and non-containers at birth. For example, hermit crabs can use containers without having to spend time learning about them. They have been "programmed" by natural selection to find shells that they use for shelter and to display dominance (Vance, 1972). On a sandy beach, a hermit crab investigates all dark blobs in its visual field in order to find a specific type of shell that has the right texture, the right size, and no holes (Bertness, 1980). The advantage of having a suitable shell is important enough for a hermit crab to start a fight with another hermit crab in order to steal its shell (Vance, 1972). What hermit crabs know about containers as an object type is limited to shells, but there is no reason for them to have more general knowledge of containers.

In contrast, humans have to spend years learning how to manipulate containers before they can use them effectively. The developmental progression of container learning in humans starts around 2.5 months of age (Hespos and Baillargeon, 2001a) and continues past the age of 6 (see Crain (1999), chap. 6). Initially, human infants learn from their observations of the sounds and the movement patterns of containers and their contents (Horst et al., 2005; Robinson

and Sloutsky, 2004). Once they have the physical capability to manipulate objects, they can learn even more about them through play and exploration (Power, 2000). What humans come to know about containers as an object type is grounded in their behavioral and perceptual repertoire, which allows them to detect and use novel containers.

Humans spend a significant amount of time learning about containers because their environments are full of them. For example, tupperware is used to preserve food, bags are used to transport belongings, and filing cabinets are used to organize documents. These containers make it possible for humans to complete many daily tasks that would otherwise be difficult or complicated. Thus, spending a significant amount of time and effort learning about containers during childhood ultimately reduces the total amount of time spent learning about their properties in adulthood.

One possible application of robots is to help eliminate arduous work for humans. Robots, however, currently lack the ability to detect and use containers in a flexible way. There are robots that can manipulate containers using inflexible programs (see, for example, Rusu et al. (2009) and Okada et al. (2009)), but little learning takes place in those systems. For example, an object may be detected as a container by matching a laser scan with a pre-specified 3D shape. Similarly, a task may be performed by executing a pre-specified behavioral routine. In terms of what these robots know about containers, however, they are more like hermit crabs than human infants, except that they are expected to work in dynamic human environments with a variety of containers.

The static programs currently used by robots have too many limitations, which make them of little use to humans in real human environments. A general purpose robot that must know its operating environment beforehand would have to be customized for the unique home or office setting that it is placed in, since no two human environments are exactly the same. Furthermore, any human environment probably has too many containers to simply program a robot with the ability to use all of them. Also, because human environments are constantly changing, continuous re-programming efforts would be required in order to maintain the usefulness of the robot. This last point is especially important: because a human environment can change, the

robot would probably end up committing terrible manipulation blunders that would render it ineffective, even if the inflexible programs written for it were entirely accurate at one point in time.

Thus, robots that *learn* how to identify containers and how to use them would provide the most utility in home and work environments. Creating robots that can learn an object category for containers is not straightforward, however, as it is not immediately obvious what the category label means. There are many different ways to describe an object that is labeled as a 'container' (Sally, 2005). For example, a short, concave object may be called a 'container' by some humans, but not by others (Sally, 2005). Furthermore, because a human has a different sensorimotor repertoire than a robot, what a human calls a container may not be a 'container' to a robot. Thus, a robot's object category for 'container' should be grounded in its own behavioral and perceptual repertoire.

1.2 Research Questions

The main research question addressed in this thesis is the following:

How can a robot learn a behavior–grounded object category for containers?

In order to answer the main research question, five subsidiary research questions are investigated. They are described below.

1. Can a robot identify containers from non-containers by interacting with them?

Theories in psychology and cognitive science have proposed that active interaction with objects is necessary to form object categories that capture the functional properties of the objects (Mandler, 2004). This may explain why humans and many animals use active behavioral exploration to learn about and to classify novel objects (Lorenz, 1996; Power, 2000). Similar behavior–grounded approaches have proven quite useful in robotics as well (Fitzpatrick et al., 2003; Stoytchev, 2005). The advantage of using behaviors to ground information about objects is that the robot can autonomously test, verify, and correct its own knowledge representation without human intervention (Sutton, 2001; Stoytchev, 2009). Similarly, it may be possible for

a robot to learn object categories for containers and non-containers by actively interacting with these two kinds of objects.

2. Can a robot use movement detection as a way to ground object categories?

One way to describe a container is that an object inside a container moves with it, whereas objects beside it do not. This property may be one embodied definition of containers that a robot can easily learn. In fact, several studies in psychology have relied on this phenomenon to determine infants' knowledge of containers (Hespos and Spelke, 2007; Hespos and Baillargeon, 2001b; Leslie and DasGupta, 1991). Studies in psychology have also shown that infants can learn object categories by observing the co-movement patterns of two different objects (Horst et al., 2005; Robinson and Sloutsky, 2004). The functional properties of objects can be captured by the co-movement patterns that they produce, which a robot may be able to observe and learn from in order to form object categories.

3. Can a robot use auditory feedback as a way to ground object categories?

Humans ground object knowledge using multiple modalities, e.g., touch and hearing, in addition to vision. It may also be possible for a robot to learn an object category for containers using sensory modalities that are different from vision. In fact, a growing body of empirical studies in embodied acoustic object recognition supports this view (Krotkov et al., 1997; Richmond and Pai, 2000; Torres-Jara et al., 2005; Sinapov et al., 2009; Sinapov and Stoytchev, 2009). These studies have shown that the sounds produced while probing an object and other forms of simple contact are sufficient for a robot to identify the material type from which the object is made of. A robot can become better at object recognition as it performs more exploratory behaviors on an object (Sinapov et al., 2009). Further work is necessary, however, to determine if a robot can use similar acoustic models to form object categories.

4. Can a robot with an extensive behavioral and perceptual repertoire form a single meaningful categorization for a set of objects?

Humans and many animals have extensive behavioral and perceptual repertoires, which they use to learn about objects (Lorenz, 1996). Presumably, a robot with multiple behaviors

and multiple sensory modalities could also use them to learn about objects. Assuming that an object categorization can be formed using information from a single behavior–modality combination, this question investigates whether multiple such categorizations can be combined into a single, unified one. In other words this question investigates how to combine different categorizations for a set of objects derived from different behavior–modality contexts.

5. Can the representation for movement detection be improved for object categorization?

Movement detection is a good way for a robot to capture the functional properties of objects for the purposes of forming object categories. Research question 2 investigates one such representation that captures the co-movement patterns between two objects. That representation, however, is limited because it does not identify the precise times during an interaction when the two objects begin to co-move. This research question investigates whether it is possible to improve the representation of movement dependencies between two different objects by adding this temporal information.

1.3 Contributions

There are two main contributions of this thesis. The first contribution is the new framework for learning behavior–grounded object categories. The second contribution is that the visual model is learned in the last step of the framework. In this thesis, the object categories learned by the robot are grounded in its interactions with objects and the resulting sounds and movement patterns that it observes. The visual model of an object category is based on the resulting object category labels derived from the robot's own experience with the objects. Because the visual model is derived from behavior–grounded object category labels, the robot can predict the functional properties of a novel object without the need for human–provided labels. Together, these contributions mark the beginning of a new approach to object category learning by robots.

1.4 Overview

The rest of this thesis is organized as follows. Chapter 2 reviews the related work in robotics and developmental psychology, which motivated this research. Chapter 3 describes the upper–torso humanoid robot that was used to conduct the experiments. Research questions 1 and 2 are addressed in Chapter 4, which shows that a robot can learn object categories by interacting with objects and observing their co-movement patterns. Research questions 3 and 4 are addressed in Chapter 5, which shows how a robot with multiple behaviors and multiple sensory modalities (audio and vision) can learn a single, unified categorization for a set of objects. Research question 5 is addressed in Chapter 6, which improves the robot's representation for co-movement detection. Finally, Chapter 7 summarizes the thesis and offers some possible directions for future work.

CHAPTER 2. RELATED WORK

This chapter reviews the related work on object categorization in both robots and infants. This chapter also covers the related work on container recognition and manipulation. It also summarizes the developmental progression of learning about containers in infants.

2.1 Object Categorization in Robotics

2.1.1 Implicit Object Categorization

Previous research has shown how robots can interactively identify the functional properties of objects, which is a first step toward identifying explicit object categories. Pfeifer and Scheier (1997) were among the first to address this problem. They programmed a mobile robot with an ability to learn how to move differently–sized objects for the purpose of cleaning its environment. The robot learned that it could carry small objects and push medium–sized objects. It ignored the large objects that it could not push or carry, which allowed it to learn faster. Thus, the robot implicitly categorized the objects by their *movability*.

Metta and Fitzpatrick (2003) found that the tasks of object segmentation and recognition could be made easier if the robot is allowed to push the objects with its arm. The detected movement after the robot's arm hit an object was used to delineate the object and construct a model for recognition. The robot used this procedure to interact with 4 different objects and to implicitly categorize them by their *rollability*. Complex internal models were avoided because "the environment can be probed and re-probed as needed" (Lungarella et al., 2003).

Ugur et al. (2007) showed how a mobile robot could learn about the traversability of objects in a simulated environment. The robot attempted to traverse an area that had randomly dispersed spheres, cylinders, and cubes. It learned which objects could be pushed aside (spheres

and cylinders in lying orientations), and which could not (cubes and cylinders in upright orientations).

Learning the similarity between objects is another type of implicit object categorization. Sinapov and Stoytchev (2008) showed how a robot could describe different tools using a hierarchical taxonomy of outcomes. The robot constructed outcome taxonomies for 6 different stick-shaped tools based on its interactions with them and used the outcome taxonomies to measure the similarity between the tools. Montesano et al. (2008) introduced a system that a robot could use to learn relationships between its actions, the perceptual properties of objects, and the observed effects. The system was evaluated with data from interactions with differently–sized spheres and cubes.

2.1.2 Learning Object Categories

The problem of interactively learning object categories is receiving increasing attention in robotics. Nakamura et al. (2007) introduced an unsupervised approach to multimodal object categorization, in which objects were categorized by the similarity of their perceptual features. A robot interacted with 40 different objects, which included 8 different categories of children's toys. The robot squeezed objects to observe hardness, viewed objects from different angles to obtain visual appearance features, and shook objects to capture acoustic properties. Results showed that when all three modalities were used the object categories formed by the robot closely resembled human–provided ones. Further results showed that visual appearance information could be used to infer the hardness of a novel object, but not its acoustic properties (Nakamura et al., 2007).

Sahai et al. (2009) showed how a humanoid robot could learn object categories based on their writability. A humanoid robot scribbled with 12 different objects on 12 different surfaces. The robot categorized objects by the frequency with which each object left a mark on a surface. Also, it categorized surfaces by the frequency with which each surface preserved the traces left by each object. The resulting categorizations separated the objects and the surfaces that provided the most utility in robotic writing tasks from those that provided the least.

Object categories can also be grounded in the robot's object recognition models. For example, Sinapov et al. (2009) demonstrated that acoustic object recognition is feasible even with a large set of objects and when multiple behaviors are performed. The robot listened to the acoustic outcomes produced by 36 objects as it grasped, shook, dropped, pushed, and tapped them. Individually, some behaviors were more useful for acoustic object recognition than others. As the robot performed more behaviors on an object, however, the recognition accuracy approached 99%. In a follow up study (Sinapov and Stoytchev, 2009), the robot was able to categorize the objects based on their material type and whether or not they had contents inside them.

Sinapov et al. (2011) also showed how a vibrotactile sensory modality could be used for the purposes of object categorization. The vibrotactile sensor was constructed using a three-axis accelerometer that was attached to one of the robot's three fingers. The robot scratched 20 different surfaces in 5 different ways as it captured data from the accelerometer in order to learn a vibrotactile recognition model for each surface. The robot discriminated between the different surfaces with an accuracy of about 45-65% when it was allowed to scratch them only once. That accuracy jumped to 80% when the robot used the vibrotactile data from all 5 scratching behaviors. The robot also learned meaningful categories of the surfaces, which were, again, grounded in the robot's recognition models.

After a robot has learned several object categories it can quickly infer the properties of a novel object by recognizing its category. The category membership of an object, however, may not be clearly defined as the object may be similar to objects in many different categories. Sinapov and Stoytchev (2011) addressed this problem with a framework for object category recognition. The task of the robot was to choose one of 6 possible categories for an object given the interaction history and predefined labels for a set of 25 other objects. The robot captured proprioceptive and auditory feedback as it performed 5 different exploratory behaviors on the novel object. Relational features were used to compare the perceptual properties of the novel object with those of the objects in the robot's interaction history. The category of the novel object was inferred using the category labels for the objects that were most similar to it.

2.2 Container Manipulation in Robotics

The related work on container manipulation in robotics can be divided into three broad categories: container detection, container use, and container control. These three areas are described in more detail below.

2.2.1 Detecting Containers

Traditionally, the object detection and manipulation problems in robotics have been solved concurrently by generating 3D models of the objects and then applying complex control algorithms that use these 3D models (see, for example, Rusu et al. (2009) and Okada et al. (2009)). These approaches are limited, however, because they suffer from the limitations of the human–defined 3D visual representation (Brooks, 1991). This subsection summarizes two articles that address the problem of container detection, but avoid using complex 3D models for the objects.

Saxena et al. (2008) proposed a supervised learning algorithm for the task of identifying good places to grasp on novel objects. The algorithm was trained using hand–labeled synthetic 3D objects. For each object a human marked the best places to grasp the object. Visual classifiers based on convolution masks were trained off-line from this labeled data. The robot was able to use the trained visual classifiers to find suitable grasp points for some dishes, which it unloaded from a dishwasher. The visual classifiers also detected good places to grasp on the rims of several bowls that were placed in an office setting. Because the classifiers were trained from human labeled data, however, the algorithm frequently identified good places to grasp on the handle of coffee cups, which were not actually graspable by the robot.

A similar learning methodology for identifying good places to grasp on objects was introduced by Montesano and Lopes (2009). Instead of using features from hand–labeled synthetic objects, however, their learning algorithm was trained using grasp points identified by the robot during its exploration. Also, because real world data was used, much less data was available to train the learning algorithm compared with (Saxena et al., 2008). Their results showed that the grasp success of the robot varied based on the type of the object it was dealing with.

Containers were used in their experiments as well and the algorithm was able to identify the rim as a potential grasp point.

2.2.2 Using Containers

One long–term goal for the field of robotics is to create robots that can assist people with daily tasks (Kemp et al., 2007). Because containers are used in many tasks that humans perform, household robots would have to learn how to use containers. Several research teams have already started to address this problem. For example, the primary tasks of the humanoid robot created by Okada et al. (2009) include: washing dishes, pouring drinks, and fetching dishes from a cupboard. Unfortunately, the operation of the robot is limited to a static environment and all of its behaviors are preprogrammed. Although Okada et al. (2009) have demonstrated that it is immediately possible to solve some container manipulation problems, many researchers are focusing on using different strategies for solving these tasks.

Nguyen and Kemp (2008) showed that some of the challenges that assistive robots face can be simplified using environment augmentation strategies designed to help service dogs. For example, dogs are better at manipulating door handles, cupboards, and appliances when towels are affixed to them. Similarly, some robots are also much better at grasping towels than handles that are designed for human hands. In their work, a mobile robot could almost always open a cupboard, a filing cabinet, and a microwave when towels were attached to them.

The robot butler HERB has also been used to solve several challenging container manipulation tasks. Chang et al. (2010) showed that transporting an object becomes easier when HERB can reposition the object before grasping it. They evaluated their approach using three containers: a frying pan, an electric kettle, and a water jug. The robot was less taxed when it transported the objects using their approach.

Kemp and Edsinger (2006b) showed how a robot could learn to detect and control the tip of an object or a tool, which may be useful for keeping containers in an upright orientation when they are manipulated. The robot grasped an object and waved it around in order to detect its tip. The tip of the object was estimated from the points in the image sequence that had the

largest optical flow. The robot was able to control the open end of a bottle and a cup after it detected each of their tips.

2.2.3 Controlling Containers and Their Contents

The problem of controlling containers and their contents is an active area of research. For instance, several papers have formulated control algorithms for the slosh-free control of liquids inside containers (Feddema et al., 1997; Romero and Ingber, 1995; Tzamtzi et al., 2007). However, these algorithms are heavily engineered for 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 the solution. Little research has addressed how a robot can learn to control containers and their contents using its own experience.

The marble maze game is one way to measure how well a robot can learn to control 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 the holes on the bottom of the container. Bentivegna et al. (2004) introduced a learning framework for a humanoid robot to solve this task. The first task of the robot was to learn how to use its preprogrammed behavioral primitives to play the game by observing a human play the game. The second task of the robot was to improve its performance by practicing the game on its own. The robot became better at choosing the right primitive to use with more practice. The robot also learned how to modify the parameters of its behavioral primitives in order to control the marble more reliably.

The ability to control two objects at the same time is also essential for container manipulation. Edsinger and Kemp (2007) showed how a robot could bimanually manipulate an object and a container at the same time in order to insert the object into the container. The robot cued a person to hand it two objects, grasped both objects and then inserted one into the container. In another demo, when given a spoon, the robot stirred the contents of a cup before placing the cup on a shelf.

2.3 Object Categorization in Infants

The object categorization framework described in this thesis was motivated by work in Developmental Psychology, which attempts to explain how infants perform categorization tasks. Psychologists have found that if infants are presented with a set of objects in which several of the objects have a common functional property, then the infants will categorize the objects based on this property (Horst et al., 2005). In this context, infants categorize the objects by the sounds that they make or by their visual movement patterns, and not by static perceptual properties like object shape or color (Horst et al., 2005; Robinson and Sloutsky, 2004).

Infants may categorize objects in this way because they learn from the events that capture their attention (Schmidt, 1995). For example, an object that makes noises will automatically draw their attention (Butterworth and Castillo, 1976). Events that violate their expectations (e.g., an unexpected movement pattern) also capture their attention (Baillargeon, 2004). Spelke argues that from birth infants can predict the movement patterns of objects and form expectations about their trajectories (Spelke and Kinzler, 2007). Infants know that there is no action without contact, that two objects cannot merge into one, that one object cannot split into multiple objects, etc. (Spelke and Kinzler, 2007).

Typically, the expectations of infants seem to agree with the laws of real-world physics, but there are some exceptions. Needham et al. (2006) found that when 7.5-month-old infants see a key-ring with keys, they perceive two distinct objects and thus predict that the key-ring and the keys will *move separately* when the key-ring is moved. More experienced 8.5-month-old infants, however, expect that the key-ring and the keys will move together because they have seen and heard the two 'distinct' objects move together many times (Needham et al., 2006). A similar shift in expectations has been observed while studying infants' knowledge of containers: infants come to expect that an object inside a container will move with the container when the container is moved (Hespos and Spelke, 2007; Hespos and Baillargeon, 2001b; Leslie and DasGupta, 1991).

Together, these findings suggest that there is a gradual process of object category learning, in which object category representations are progressively grounded in different actions and

their outcomes. Indeed, it is believed that infants first represent "what actions can be done on objects of certain kinds" (Rochat and Striano, 1998), before they incorporate the object's visual shape into their representations. This may imply that behaviors and their outcomes form the bases of infants' initial concepts. Baillargeon has shown that only after infants have formed an "initial concept" do they begin to incorporate variables into their representations that serve to refine their predictions (Baillargeon, 1994). Passively observable object properties such as shape are learned gradually over time if they consistently appear with members of a category (Hespos and Baillargeon, 2001b; Wang and Baillargeon, 2008; Hespos and Baillargeon, 2006; Baillargeon and DeVos, 1991; Aguiar and Baillargeon, 1998; Sitskorn and Smitsman, 1995).

The fact that infants gradually improve their object category representations as they gain more experience supports Cohen's hypothesis that there is an information processing mechanism underlying object categorization (Cohen, 2003). One information processing mechanism known to be used by humans of all ages is the detection of the frequency of occurrence of a stimulus (Hasher and Zacks, 1984). Humans implicitly extract the frequency information for a variety of naturally occurring phenomena (Hasher and Zacks, 1984). It is reasonable to assume that infants may also use frequency information to separate objects into categories.

Neuroscientists have suggested that the representation of an abstract object category involves a multiregional activation of the brain, which reaches many different sensory areas (Simmons and Barsalou, 2003; Damasio, 1989). These multimodal representations are formed in high-level convergence zones in the hierarchical organization of the brain (Barsalou et al., 2003; Simmons and Barsalou, 2003; Damasio, 1989). At these convergence zones, fragments of data from multiple modalities are bound together if they occur coincidentally or sequentially in space and time (Simmons and Barsalou, 2003; Damasio, 1989). The resulting multimodal representation encodes how objects in the category sound, move, look, feel, etc.

After years of exploration, children learn many different object categories, which they represent using a hierarchical structure (Rosch, 1978). Their three–level hierarchy encodes superordinate, basic, and subordinate object categories. Infants first learn to categorize objects at the basic level (e.g., containers, hammers, and screwdrivers). With more experience, in-

fants learn superordinate–level categories, which are abstract groupings of basic–level categories (e.g., tools). Last of all, infants learn subordinate–level categories, which define small differences between objects at the basic–level (e.g., cups, bowls, and baskets are all examples of subordinate–level container categories). The order in which different levels of object categories are learned demonstrates that basic–level object categories are the easiest for infants to learn, followed by superordinate and then subordinate categories.

Infants have a separate developmental learning progression for each object category that they form. The knowledge that they acquire for one category is not necessarily generalized to other categories. For instance, infants learn to attend to the height variable in occlusion events at 3.5 months, but in containment events they don't attend to the height variable until 7.5 months (Hespos and Baillargeon, 2001a). Similar knowledge delays have been found for many other variables and categories (Baillargeon, 1994).

Infants first learn categories for the simplest kinds of objects and events. The simpler the object or event, the easier it is for them to learn about it. For example, occlusion is easier for infants to learn about than containment (Hespos and Baillargeon, 2006), which is why infants learn to incorporate the height variable into their perceptual repertoire for occlusion before containment. The height variable is included even later in other, more complicated, event categories. For example, the height variable is incorporated into the covering event category by 12 months (Wang et al., 2005).

Finally, it is worth mentioning that infants only retain the categories that are emphasized by their native language. Before language learning, infants are sensitive to many different categories. However, only a subset of these concepts are reinforced by their language, and these are the categories that they retain (Bowerman and Choi, 2003). For example, infants raised in English–speaking homes retain a category for containment spatial relationships, whereas infants raised in Korean–speaking homes retain a category for tight–fit and loose–fit spatial relationships. It is possible, however, for them to relearn the categories that are not emphasized by their native language, e.g., an English–speaking adult can relearn to distinguish between tight-fit and loose-fit spatial relationships (Bowerman and Choi, 2003).

2.4 The Developmental Progression of Container Learning in Infants

Containers are one of the first abstract object categories that infants learn about (Casasola et al., 2003). Their single distinguishing feature—their concave shape—makes them one of the simplest kinds of objects. However, it takes years of play and exploration before infants can master how to detect and use containers (Crain, 1999). During this time, the progression of container learning is intermittent (see Table 2.1). Months can go by before infants learn to attend to any additional variables that are important for containment.

Infants begin to understand how objects move around when they are placed inside containers at about 2.5 months of age. 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 (Hespos and Baillargeon, 2001a), which could form the basis for the 'initial concept' of containers (Baillargeon, 1994). However, they do not yet know how two objects can get into that spatial relationship with one another.

Infants incorporate more variables into their knowledge about containers as they grow older. The first variables that they acquire are perceptual. For example, infants learn to attend to the width of objects in containment events when they are approximately 5.5 months old (Hespos and Spelke, 2004). Infants at around 6 months of age have learned that containment is possible only with concave–shaped objects, which signifies the formation of an abstract concept of containers (Casasola et al., 2003). By about 7.5 months, they learn to attend to the height of objects in containment events. Many more perceptual variables associated with containers are learned over the course of development (see Table 2.1).

Infants begin to learn how to use 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 (Largo and Howard, 1979a). Later, they try inserting blocks into containers and then try to shake the containers. Their insertion behaviors peak when they are around 15 months old (Largo and Howard, 1979a), which suggests that infants have spent a period of several months (on and off) learning how their own movements affect the movements of objects

The Developmental Progression of Container Learning in Humans		
Age	Type of Knowledge	Source
2.5 months	Infants can distinguish between occlusion and containment events. They understand that an object inside a container can move with it. They also know that an object cannot be lowered into a closed container.	(Hespos and Baillargeon, 2001a)
3 months	Infants can distinguish between a concave and a convex hemisphere.	(Bonniec, 1985)
5 months	Infants are sensitive to the width of objects in containment events. They can predict the movement patterns of an object that is lowered into a tight-fitting or a loose-fitting container.	(Hespos and Spelke, 2004)
6 months	Infants can identify novel objects as containers, which indicates that they have formed an abstract categorical representation of containment.	(Casasola et al., 2003)
7.5 months	Infants are sensitive to the height of objects in containment events.	(Hespos and Baillargeon, 2001a)
8 months	Infants begin to understand how their hands can be inserted into containers.	(Bonniec, 1985)
8.5 months	Infants understand that wide, compressible objects can be lowered into narrow containers.	(Aguiar and Baillargeon, 1998)
9 months	Infants begin to understand how objects can be inserted into containers.	(Bonniec, 1985)
10 months	Infants are sensitive to the transparency of objects in containment events.	(Luo and Baillargeon, 2005)
12 months	Infants expect to find things inside right-side-up containers, but not inside upside-down ones.	(Freeman et al., 1980)
15 months	This is the peak age for container play. At this age infants also start saying the word 'into.'	(Largo and Howard, 1979a)
18 months	Infants understand that only an open container, but not a closed one, can contain liquid.	(Caron et al., 1988)
20 months	Infants are sensitive to the bottoms of containers in containment events.	(MacLean and Schuler, 1989; Caron et al., 1988)
6 years	Children understand that two differently shaped containers with the same volume can hold the same amount of liquid.	(see Crain (1999), chap. 6 on Piaget's Theory)

Table 2.1 A comprehensive list of what humans know about containers at different stages of their development.

inside containers.

Table 2.1 indicates that infants learn about containers in small increments, one container property at a time. For instance, they have learned to attend to width at 5 months (Hespos and Spelke, 2004), height at 7.5 months (Hespos and Baillargeon, 2001a), compressibility of objects at 8.5 months (Aguiar and Baillargeon, 1998), transparency at 10 months (Luo and Baillargeon, 2005), etc. Individually, each of these properties may seem trivial or uninteresting. In combination, however, these variables help infants refine their predictions of containment events. As more and more variables are learned, infants can predict the outcome of containment events more and more accurately.

Table 2.1 also shows the order in which these variables are learned during the developmental progression of the container object category. For example, infants may attend to the dimensions of objects (e.g., width and height) before they attend to the material properties of objects (e.g., compressibility and transparency). They may also learn how discrete rigid objects interact before they learn how fluids interact. These differences suggest that some properties of containers are easier to learn about than others.

Perhaps the most striking data in Table 2.1 is that infants are able to learn an abstract representation of containment events by 6 months (Casasola et al., 2003), which occurs well before they can manipulate containers themselves when they are 9 months old (Bonniec, 1985). Infants probably observe a large number of containment events every day and, apparently, what they observe is enough for them to learn to distinguish containers from non-containers. The sounds and the visual movement patterns of containers as they are being manipulated by someone else are their primary sources of information. Other sources of information, e.g., direct tactile and proprioceptive feedback, are probably not necessary, at least initially.

As table 2.1 shows, there are many different things that a robot may have to learn about containers before it is able to detect and use them in a general way. This thesis, however, investigates only how a robot can learn an object category for containers, which is among the first aspects of container learning in human infants. The robot learns object categories using the same type of sensory data that human infants use to form an object category of containers, i.e.,

from the sounds and the movement patterns of objects, but not from proprioceptive and tactile feedback (see Chapters 4, 5, and 6). Furthermore, in a recent study we demonstrated that the robot can learn about containers when a human, not the robot, interacts with the objects and generates the necessary movement patterns (Sukhoy et al., 2011). Thus, like infants, the robot can learn an object category for containers either when the interactions with objects are performed by itself or by a human. In spite of that, the focus of this thesis is on how a robot can learn an object category for containers by interacting with the objects itself, and leaves learning from observation for future work.

CHAPTER 3. EXPERIMENTAL PLATFORM

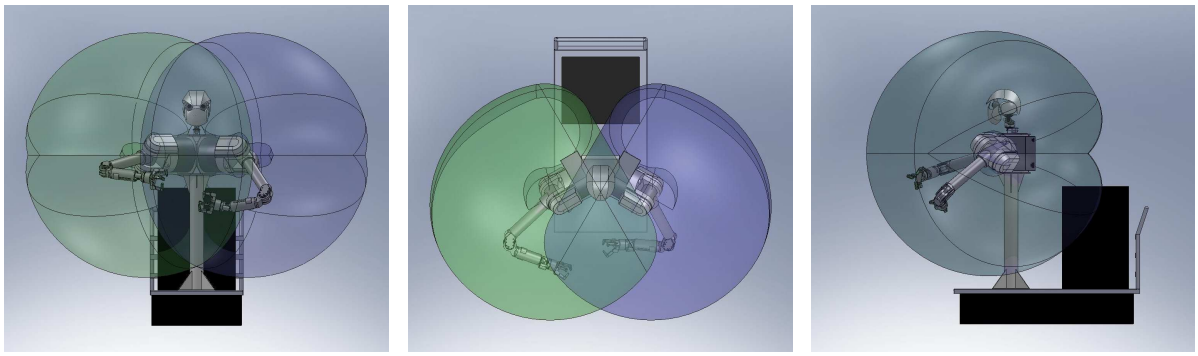
This chapter describes the robot platform that was used to conduct all of the experiments described in this thesis.

3.1 The Upper–Torso Humanoid Robot

The upper–torso humanoid robot used in this thesis is composed of two Barrett Whole Arm Manipulators (WAMs), which are mounted in a human–like configuration (see Fig. 3.1). This configuration allows the robot to use some of the same interaction strategies that infants use to learn about objects. For example, an infant sitting in a high–chair learns by interacting with objects placed on the table in front of it. Similarly, the robot can interact with objects placed on the table in front of it.

The range of motion of the robot's arms is depicted in Fig. 3.2. Each WAM arm can move around a hemisphere–shaped region of space. Bimanual manipulation is possible where the two hemispheres overlap.

Figure 3.1 The upper–torso humanoid robot used in this thesis, shown here moving a container. Objects were placed on the table before each trial by the experimenter.



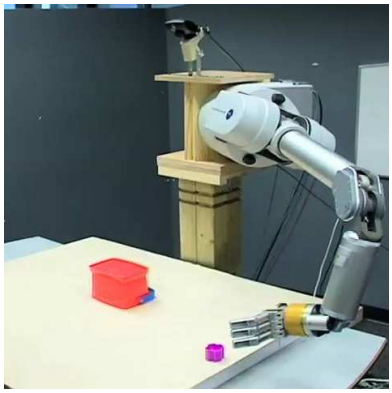
Front View Top View Side View

Figure 3.2 CAD drawings that depict the range of motion of both WAMs from the front, the top, and the side of the robot. Bimanual manipulation is possible where the two hemispheres overlap. These images were drawn by Steven Lischer.

The hardware structure of the robot evolved over time (see Fig. 3.3). For the first experiment (described in Chapter 4), only one WAM was mounted on a wooden platform, which was a temporary fixture that served as a mock-up version for the final metal platform. The structure of the robot was finalized by the time the following two experiments were undertaken (described in Chapters 5 and 6). The overall configuration of the robot has since remained the same, but many other aspects of the robot's software architecture are still evolving.

The upper–torso humanoid robot was designed to be very sturdy and to look visually appealing. The sturdiness of the robot derives from a hollow metal cylindrical stand. The top end of the stand is attached to an angled mounting bracket, to which the two arms are bolted. The bottom end of the stand is attached to a metal cart. The robot also has an articulated functional head. Plastic covers made using a 3D printer were added to the robot's chest and head to improve its visual appearance.

The metal cart that serves as the base of the robot is also used to hold the electronic components of the robot. Cords from the two WAMs and the head are fed down the hollow metal stand. From the base of the stand the cords are routed to various electronics placed in a cabinet on the back of the cart. The cabinet includes power boxes for the two WAMs and Hands, two computers that control the robot and process sensory data, and audio components for capturing sound.



a) September 2008

b) September 2009

Figure 3.3 The evolution of the robot: a) the form of the robot at the time of the first experiment (described in Chapter 4); b) the form of the robot at the time of the following two experiments (described in Chapters 5 and 6).

3.2 The WAM

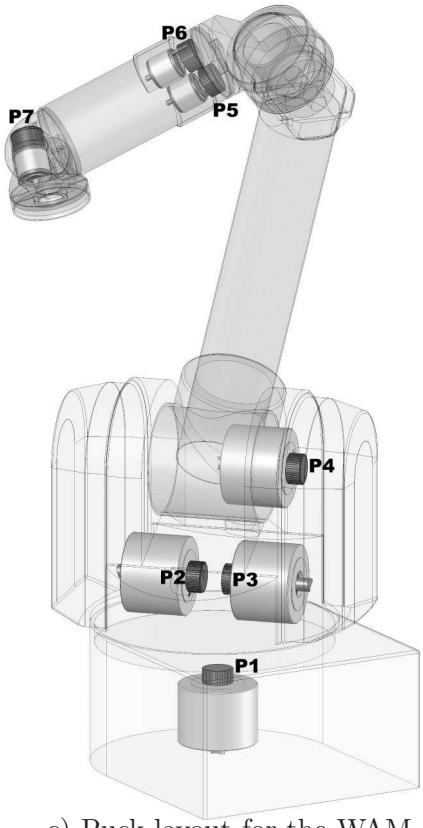
The Barrett WAM is different from most robotic arms because of its open–loop backdrivability. An arm with good backdrivability can both sense the force applied to it and apply joint torques at the same time. Each joint in the WAM arm is driven by a low inertia and low friction cable-and-cylinder transmission system, which gives the robot its good backdrivability (see Fig. 3.4.a). The WAM has seven joints: four joints in the shoulder and three joints in the wrist. Each joint is controlled by a servo–controller located at the joint (see Fig. 3.4.b and 3.4.c). Because of their miniature size, the servo–controllers are called pucks. When an external force is applied to the arm, the torque required to keep the arm in its current position changes, but the pucks sense this change in torque and modify the arm position appropriately.

Many different arm positions can be reached by the WAM. Its effective sphere of reach is approximately 1 meter when all seven joints are fully extended. At shorter distances, the position of the end effector can be maintained even though the arm can move through a variety of poses. This is known as *kinematic redundancy* and is possible because the WAM has seven joints (Rooks, 2006). Kinematic redundancy is advantageous when the robot has to manipulate an object that is placed in a cluttered environment.

The WAM arm is especially suitable for manipulating objects when fitted with the Barrett Hand as its end effector. The three–finger Barrett Hand can grasp objects in a variety of ways.

a) A cable-and-cylinder drive

b) A puck servo–controller



c) Puck layout for the WAM

Figure 3.4 The hardware that distinguishes the Barrett WAM from other articulated–arm robots. a) cable-and-cylinder drive for one of the joints. b) close-up of the puck servo–controller. One puck is used for each of the seven joints of the WAM. c) CAD drawing showing the positions of all pucks and motors in the WAM. The pictures were adapted from (Rooks, 2006).

The outer two fingers can swivel around the palm to provide different grasp styles, including power grasps, precision grasps, and weak grasps. Force sensors on each finger ensure that the amount of force applied during a grasp is just strong enough to squeeze an object without damaging it.

The WAM arm coupled with a Barrett Hand weighs 28 kilograms. It can lift and hold a 3 kilogram object and move it at speeds up to 3 meters per second. The robot can lift heavier loads, but that would reduce the lifetime of the cables used to drive the robot.

Figure 3.5 The robot's vision system: a) a closeup of the 3D camera; b) a color image of a red non-container captured by the camera when mounted on the robot; c) the depth image corresponding to b).

3.3 The Robot's Sensory Modalities

The experiments presented in this thesis used three sensory modalities: video, audio, and proprioception. The robot's visual system includes two cameras (Quickcams from Logitech), which are mounted in the robot's head. Each camera captures 640x480 color images at 15 fps.

The robot's visual system also includes a 3D camera—a ZCam manufactured by 3DV Systems (ZCam, 2008). The ZCam captures 320x240 depth images and 640x480 color images. The relative resolution of the depth images is accurate to ±1-2 cm. The depth images are calculated by first pulsing infrared light in two frequencies and then detecting and processing the reflected pulses of light. Figure 3.5 shows a close up of the ZCam and its field of view when mounted on the robot.

The robot's auditory system consists of two Audio-Technica U853AW hanging microphones (see Fig. 3.6.a), which are mounted in the robot's head. The microphones' output is fed through two ART Tube MP Studio Microphone pre-amplifiers (see Fig. 3.6.b). The signal from the amplifiers is fed to a Lexicon Alpha bus-powered interface (see Fig. 3.6.b), which transmits

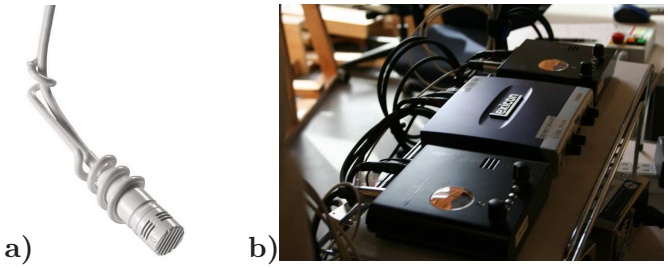


Figure 3.6 The robot's auditory system: a) one of the two microphones (an Audio-Technica U853AW Hanging Microphone); b) the two pre-amplifiers (ART Tube MP Studio Microphone pre-amplifiers) and the bus–powered interface (a Lexicon Alpha bus-powered interface) shown in their current setup on the back of the cart.

the signal to a Linux PC over USB. Sound was captured using the Java Sound API at 44.1 KHz over a 16-bit mono channel.

The robot's proprioception stream consists of the joint torque and the joint position data from the two WAM arms. Each of the seven pucks in the two WAMs encodes a joint torque and a joint position value. Thus, the proprioceptive feedback consists of 28 values at any given point in time. The proprioception data for the robot was captured at 500 Hz using the robot's low-level API.

CHAPTER 4. SEPARATING CONTAINERS FROM NON-CONTAINERS USING VISION*

Object categorization is one of the most fundamental processes in human infant development (Cohen, 2003). Yet, there has been little work in the field of robotics that addresses object categorization from a developmental point of view (Fitzpatrick et al., 2008). Traditionally, object categorization methods have been vision based (Sutton et al., 1994). However, these disembodied approaches are missing a vital link, as they leave no way for a robot to verify the correctness of a category that is assigned to an object. Instead, the robot's representation of object categories should be grounded in its behavioral and perceptual repertoire (Sutton, 2001; Stoytchev, 2006).

This chapter proposes an embodied approach to object categorization that allows a robot to *ground* object category learning in its sensorimotor experience. More specifically, the robot's task is to detect two classes of objects: containers and non-containers. In the proposed framework, interaction and movement detection are used to ground the robot's perception of these two object categories. First, the robot forms a set of outcome classes from the detected movement patterns during its interactions with different objects (both containers and non-containers). Second, objects are grouped into object categories by the frequency that each outcome class occurs with each object. Third, a perceptual model is learned and used to generalize the discovered object categories.

The framework was tested on a container/non-container categorization task, in which the robot dropped a block above the object and then pushed the object. First, the robot identified

*This chapter is a paper that was presented at the 2009 IEEE International Conference on Development and Learning (ICDL) (Griffith et al., 2009).

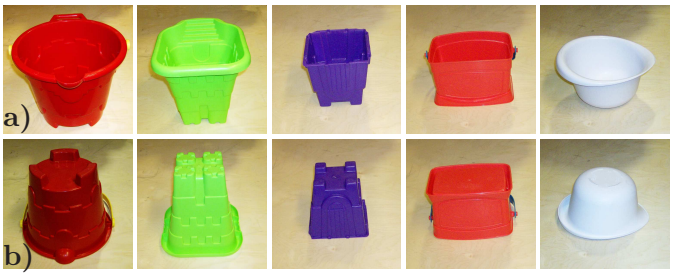


Figure 4.1 The objects used in the experiments: a) the five containers: big red bucket, big green bucket, small purple bucket, small red bucket, small white bowl; b) these containers can easily become non-containers when flipped over.

three outcomes after interacting with the objects: co-movement outcomes, separate movement outcomes, and noisy outcomes. Second, the robot identified that co-movement outcomes occurred more often with containers than with non-containers and thus separated containers from non-containers using unsupervised clustering. Third, a perceptual model was learned and was shown to generalize well to novel objects. Our results indicate that the robot can use interaction as a way to detect the functional categories of objects in its environment.

4.1 Experimental Setup

4.1.1 Robot

The experiments in this chapter were performed with a Barrett WAM mounted on a wooden platform. The ZCam was also used, which captures 320x240 depth images in addition to color images. See chapter 3 for more details.

4.1.2 Objects

The robot interacted with different container and non-container objects that were placed on a table in front of it (see Fig. 4.1). The containers were selected to have a variety of shapes and sizes. Flipping the containers upside-down provided a simple way for the robot to learn about non-containers. Therefore, the robot interacted with 10 "different" objects, even though there were only 5 real objects. During each trial the robot grasped a small block and dropped it in the vicinity of the object placed in front of it. The object was then pushed by the robot and the movement patterns between the block and the object were observed.

4.1.3 Robot Behaviors

Four behaviors were performed during each trial: 1) *grasp* the block; 2) *position* the hand in the area above the object; 3) *drop* the block; and 4) *push* the object. A person placed the block and the object at specific locations before the start of each trial. Figure 4.2 shows a sequence of interactions for two separate trials. The four behaviors are described below.

Grasp Behavior: The robot grasped the block at the start of each trial. The grasp behavior required the robot to open its hand, move next to the block, and close its hand.

Position Behavior: The robot positioned its hand in the area above the object after grasping the block. Drop positions were uniformly selected from a 40cm×40cm area relative to the center of the object. The object was consistently placed in the same location.

Drop Behavior: The robot dropped the block once its hand was positioned in the area above the object. The block either fell into the object (except when the trial involved non-container objects), or fell beside it. In some cases the block rolled off the table (approximately 5% of 1000 trials). In these situations, a human experimenter placed the block at the location on the table where it rolled off.

Push Behavior: The robot pushed the object after dropping the block. The pushing direction was uniformly selected between two choices: *push-toward-self* or *push-toward-right-of-self*. The robot pushed the object for 10 cm with an open hand (see Fig. 4.2.d and 4.2.e).

4.2 Methodology

4.2.1 Data Collection

Experimental data was collected during the *push* behavior. This interaction was captured from the robot's 3-D camera as a sequence of 640x480 color images and 320x240 depth images recorded at roughly 20 fps. The *push* behavior lasted approximately 3.5 seconds for a single trial. A total of *roughly* $3.5 \times 20 = 70$ images were recorded per trial.

For each of the 10 objects shown in Fig. 4.1 the robot performed 100 interaction trials for a total of 1000 trials.

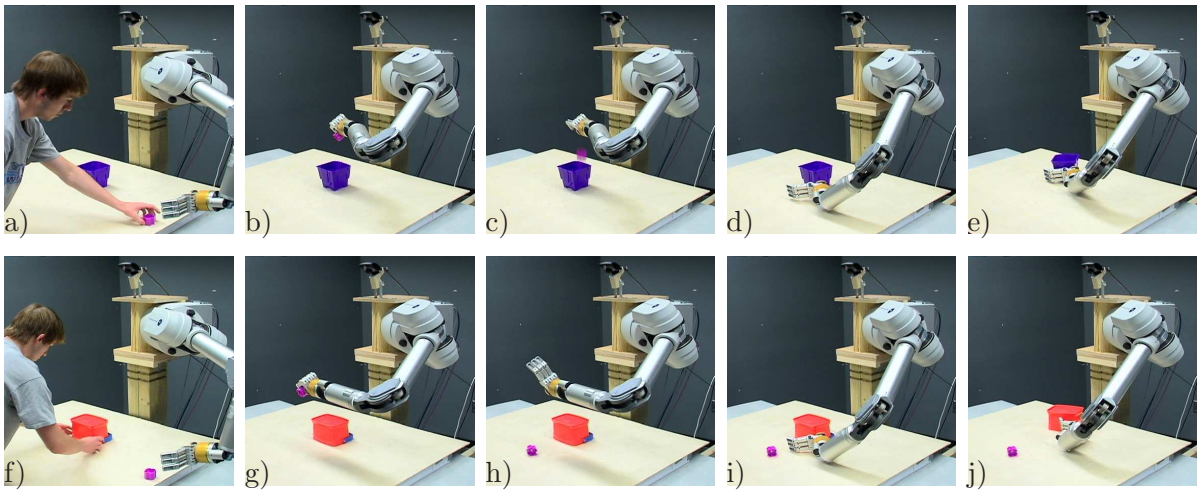


Figure 4.2 The sequence of robot behaviors for two separate trials: a) before each trial a human experimenter placed the block and the container at a marked location; b) the robot carried out each trial by grasping the block and positioning the hand in the area above the container; c) dropping the block; d) starting the *push* behavior; e) and ending the *push* behavior. f)-j) The same as a)-e) but for a non-container object.

4.2.2 Movement Detection

The robot processed the frames from the 3-D camera to detect movement and to track the positions of the block and the object. To locate each object, the color images were segmented based on the object's color and the coordinates of the largest blobs were calculated. The value for z was found at the corresponding $[x, y]$ position in the depth image. The last known position was used if the block or the object was occluded.

Movement was detected when the $[x, y, z]$ position of the block or the $[x, y, z]$ position of the object changed by more than a threshold, δ, over a short temporal window $[t', t'']$. The threshold, δ, was empirically set to 10 pixels per two consecutive frames. A box filter with a width of 5 was used to filter out noise in the movement detection data.

4.2.3 Acquiring Interaction Histories

Once a given trial, i, was executed, the robot constructed the triple (B_i, O_i, F_i), indicating that the behavior $B_i \in \mathcal{B}$ was used to interact with object $O_i \in \mathcal{O}$ and outcome vector F_i was observed. The behavior represented with B_i was either *push-toward-self* or *push-toward-right-*

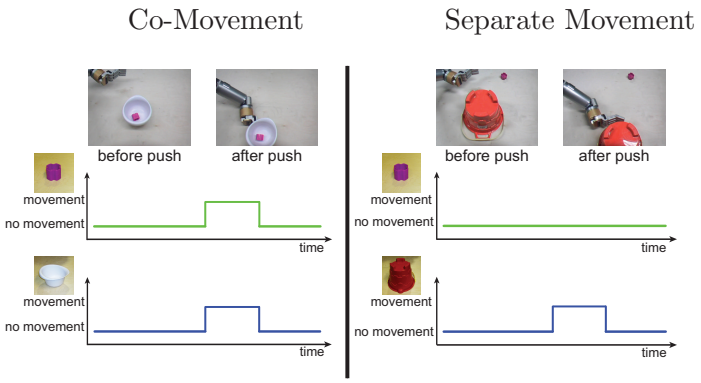


Figure 4.3 An example of co-movement (left) and separate movement (right). Co-movement outcomes occur when the block falls into a container. In this case, the block moves when the container moves. Separate movement outcomes occur when the block falls to the side of the container or during trials with non-containers. In these instances the movements of the two objects are not synchronized.

of-self. Also, $\mathcal{O} = \{O_1, \ldots, O_{10}\}$ denoted the set of objects (containers and non-containers) used in the experiments. Finally, each outcome was represented with the numerical feature vector $F_i \in \mathbb{R}^2$.

The outcome $F_i = [f_1^i, f_2^i]$ captured two observations: 1) whether the object O_i and the block moved at the same time, and 2) whether the object O_i and the block moved in the same direction. Hence, f_1^i equaled the number of time steps in which both the object and the block moved together divided by the number of time steps in which the object moved. In other words, the value of f_1^i will approach 1.0 if the object and the block move at the same time, but it will approach 0.0 if the object and the block do not move at the same time.

Additionally, the second outcome feature, f_2^i, was defined as

$$f_2^i = ||\Delta pos_i(object) - \Delta pos_i(block)||, \tag{4.1}$$

where $\Delta pos_i(object) \in \mathbb{R}^3$ and $\Delta pos_i(block) \in \mathbb{R}^3$ are equal to the detected change in position of the object and the block, respectively, while they are pushed during trial i. In other words, the value of f_2^i will approach 0.0 if the object and the block move in the same direction, but it will become arbitrarily large if the object and the block move in different directions. Both f_1^i and f_2^i are required in order to represent whether the block and the object move together or move separately (see Fig. 4.3).

4.2.4 Discovering Outcome Classes

Various co-movement patterns can be observed by acting on different objects in the environment. Outcome classes can be learned to represent these patterns. The robot's interaction history would change over time, gradually growing more robust to outliers. A variety of factors affect the number of possible outcome classes (e.g., number of perceptual observations). Let $\{F_i\}_{i=1}^{1000}$ be the set of observed outcomes after performing 100 interaction trials with each of the 10 objects. We used unsupervised clustering with X-means to categorize the outcomes, $\{F_i\}_{i=1}^{1000}$, into k classes, $C = \{c_1, \ldots, c_k\}$. X-means extends the standard K-means algorithm to estimate the correct number of clusters of the dataset (Pelleg and Moore, 2000). Section 4.3.1 describes the results.

4.2.5 Discovering Object Categories

Certain outcome classes are observed more often with some objects than with others. This difference can be used to form object categories. For example, compared to non-containers, a container will more often exhibit the co-movement outcome when a small block is dropped above it. Therefore, the robot can use its interaction history with objects to discover different object categories, which might be how infants go about achieving this task (Cohen, 2003).

Let us assume that the robot has observed a set of **outcome classes** $C = \{c_1, \ldots, c_k\}$ from its interactions with several objects, $\mathcal{O} = \{O_1, \ldots, O_{10}\}$. Let $H_i = [h_1^i, \ldots, h_k^i]$ define the interaction history for object i, such that h_j^i is the number of outcomes from outcome class c_j that were observed when interacting with the i^{th} object.

The interaction histories were normalized using zero mean and unit standard deviation. Let the normalized interaction history, Z_i, for interaction history H_i be defined as $Z_i = [z_1^i, \ldots, z_k^i]$, such that $z_j^i = (h_j^i - \mu_j)/(\sigma_j)$, where μ_j is the average number of observations of c_j, and σ_j is the standard deviation of observations of c_j. Through this formulation, the i^{th} object is described with the feature vector $Z_i = [z_1^i, \ldots, z_k^i]$.

To discover **object classes**, the robot clustered the feature vectors $Z_1, \ldots, Z_{10}$ (one for each of the 10 objects shown in Fig. 4.1) using the X-means clustering algorithm. Clusters

found by X-means were interpreted as object categories. X-means was chosen to learn both the individual *outcome classes* and *object classes* because: 1) it is an unsupervised clustering algorithm; and 2) it does not require the human programmer to know the number of clusters in advance. The results are described in section 4.3.2.

4.2.6 Categorizing Novel Objects

It is impractical for a robot to categorize all novel objects by interacting with them for a long time. However, the robot can interact with a few objects to form a behavior-grounded object category and then learn a generalizable perceptual model from these objects. This method allows a robot to quickly determine the category of a novel object.

The predictive model could classify novel objects once it is trained with automatically labeled images. In this case, the robot interacted with 10 objects, so 10 depth images were used to train the predictive model, as shown in Figure 4.4 (only one image of each object was necessary since the robot viewed objects from a single perspective). The labels assigned to the 10 images were automatically generated by X-means during the object categorization step. For each depth image, let $s^i \in \mathbb{R}^n$ be a set of perceptual features extracted by the robot. The robot learns a predictive model $\mathcal{M}(s^i) \rightarrow k_i$, where $k_i \in \{0, 1, \ldots, K\}$ is the predicted object category for the object described by features s_i, and K is the number of object categories detected by the X-means clustering algorithm.

The task, then, is to determine a set of visual features that can be used to discriminate between the learned clusters of objects. These objects have been grouped based on their functional features, *i.e.,* co-movement and non-co-movement. It is reasonable to assume that other features, like the shape of the objects, might be related to these functional properties, and therefore allow for the quick classification of novel objects into these categories. Presumably, as children manipulate objects and extract their functional features, they are also correlating visual features with their observations. Accordingly, the robot also attempted to build a perceptual model of containers by extracting relevant visual features and associating these features with the functional clusters.

To do this, the robot used the *sparse coding* feature extraction algorithm, which finds compact representations of unlabeled sensory stimuli. It has been shown that sparse coding extracts features similar to the receptive fields of biological neurons in the primary visual cortex (Olshausen and Field, 1996), which is why it was chosen for this framework. The algorithm learns a set of basis vectors such that each input stimulus can be approximated as a linear combination of these basis vectors. More precisely, given input vectors $x_i \in \mathbb{R}^m$, each input x_i is compactly represented using basis vectors $b_1, \ldots, b_n \in \mathbb{R}^m$ and a sparse vector of weights $s^i \in \mathbb{R}^n$ such that the original input $x_i \approx \sum_j b_j s^i_j$. The weights $s^i \in \mathbb{R}^n$ represent the compact features for the high-dimensional input image x_i. We used the algorithm and MATLAB implementation of Lee *et al.* (Lee et al., 2007) for learning the sparse coding representation.

The robot extracted 2 features (i.e., $n = 2$ in the formulation above) from the 10 objects used during the trials, as shown in Figure 4.5. The figure shows that the algorithm extracted a feature characteristic of container objects and a feature characteristic of non-container objects. Each input x_i consisted of a 30 x 30 depth image of the object, as shown in Figure 4.4.

Given a novel object, O^{test}, the robot extracted a 30 x 30 depth image of it, x_{test}, and found the feature weight vector $s^{test} \in \mathbb{R}^2$ such that $x_{test} \approx \sum_j b_j s^{test}_j$. The robot then used the Nearest Neighbor algorithm to find the training input x_i (a 30 x 30 depth image of one of the 10 training objects) such that the Euclidean distance between its sparse feature weight s^i and s^{test} is minimized. The robot subsequently categorizes the novel object (as either 'container' or 'non-container') with the same class label as the nearest neighbor training data point.

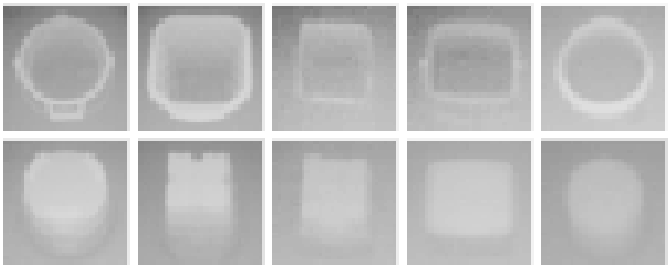


Figure 4.4 The 10 depth images of the objects used as input to the sparse coding algorithm. The 320x240 ZCam depth images were scaled down to 30x30 pixels before the algorithms generated sparse coding features from them.

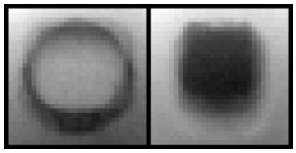


Figure 4.5 The two basis vectors that were computed as a result of the *sparse coding* algorithm. These visual features were later used to classify novel objects as 'containers' or 'non-containers.'

4.3 Results

4.3.1 Discovering Outcome Classes

Figure 4.6 shows the results of unsupervised clustering using X-means to group trials with similar outcome classes. The figure also shows the frequency with which each outcome class occurred for each container and non-container. X-means found three outcome classes among all of the trials: one cluster of co-movement events, one cluster of separate movement events, and a third cluster corresponding to noisy observations.

The first two outcome classes were expected. We found that the third outcome class had several causes. Sometimes the human experimenter was placing the block on the table after it fell off, sometimes the block was slowly rolling away from the container, and sometimes the movement detection noise was not completely filtered out. However, the fact that the robot formed a co-movement outcome class meant that it could find meaningful relationships among its observations. This result suggests that the robot could possibly categorize objects in a meaningful way.

4.3.2 Discovering Object Categories

The result of unsupervised clustering using X-means to categorize objects resulted in two object categories: one cluster with the five containers (Fig. 4.1 a) and another cluster with the five non-containers (Fig. 4.1 b).

This result shows that a robot can successfully acquire an experience-grounded concept of containers. In other words, this grounded knowledge of containers could be verified at any time

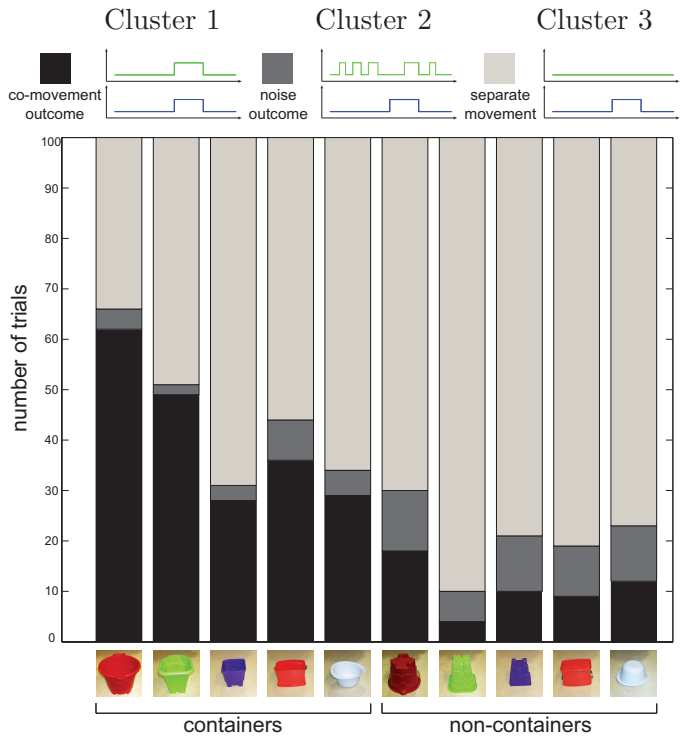


Figure 4.6 The result of unsupervised clustering using X-means to categorize outcomes. X-means found three outcome classes: co-movement (black), separate movement (light gray), and cases of noise (dark gray). The co-movement outcome occurred more often with containers compared to non-containers. Movement duration and movement vector features were extracted from the robot's detected movement data and used during the clustering procedure.

by re-probing the environment using the same sequence of interactions. But this also means that further experience with containers could enhance the robot's container categorization ability.

The result also supports the claim that co-movement patterns can provide the robot with an "initial concept" (Baillargeon, 1994) of containers when the interaction involved dropping a block from above and pushing the object. In this case, the functional properties of the objects were more salient than other variables that affected the outcome (e.g., size and shape).

4.3.3 Evaluation on Novel Objects

The robot was tested on how well it could detect the correct object category of 20 novel objects (see Fig. 4.7). The set of novel objects included 10 containers and 10 non-containers. Using the extracted visual features and the Nearest Neighbor classifier (see section 4.2.6), the

Figure 4.7 The result of using a Nearest Neighbor classifier to label novel objects as 'containers' or 'non-containers'. The flower pot (outlined in red) was the only misclassified object. Sparse coding features were extracted from the 10 training objects and used in the classification procedure.

robot was able to assign the correct object category to 19 out of 20 test objects. This implies that the robot not only has the ability to distinguish between the containers and non-containers that it interacts with, but it can also generalize its grounded representation of containers to novel objects that are only passively observed.

4.4 Summary

This chapter proposed a framework that a robot could use to successfully form simple object categories. The proposed approach is based on the principle that the robot should ground object categories in its own sensorimotor experience. The framework was tested on a container/non-container categorization task and performed well. First, the robot identified co-movement outcomes, separate movement outcomes, and noisy outcomes from the movement patterns of its interactions with objects. Second, the robot perfectly separated containers from non-containers using the pattern that co-movement outcomes occurred more often with containers than non-containers. Third, the robot used this separation to learn a perceptual model, which accurately detected the categories of 19 out of 20 novel objects.

These results demonstrate the feasibility of interaction-based approaches to object categorization. In other words, a robot can use interaction as a method to detect the functional categories of objects in its environment. Furthermore, a robot can also learn a perceptual model to detect the category of objects with which the robot has not interacted. Therefore, when the perceptual model is in question, the robot can interact with the object to determine the object category.

Numerous results in developmental psychology laid the groundwork for the framework presented in this chapter. Future work should continue to build on this foundation by relaxing some of the assumptions at the center of this approach. An obvious extension would be to find methods of interaction-based object categorization that go beyond co-movement detection. Another interesting extension would be to modify the current framework so that the robot learns category–specific interactions (e.g., dropping a block above an object and pushing the object) through imitation. The approach presented in this chapter should also be evaluated in a richer environment with more objects, behaviors, and more categories of objects.

Some of these extensions are addressed in the following two chapters.

CHAPTER 5. SEPARATING CONTAINERS FROM NON-CONTAINERS USING AUDIO AND VISION*

Object categorization is a fundamental skill that emerges early in the course of human infant development (Rakison and Oakes, 2003). From the moment infants begin to manipulate objects, they can identify differences between them in terms of the sensations that the objects produce (Power, 2000). As infants gain more control over their bodies, they begin to grasp, mouth, scratch, and bang objects in order to learn about them (Rochat, 1989). These exploratory behaviors and the sensations that they produce lay the foundations for forming many different object categories (Gibson, 1988).

Each object category that infants learn in this way is associated with a set of functional and perceptual properties (Rosch, 1978). For example, containers have the functional property that a block placed inside of a container will start to move when the container is moved. Containers also have the perceptual property that they look concave. Different object categories are represented by different collections of properties. Over time, infants' category representations become more diverse (Barsalou et al., 2003).

In contrast, the majority of object categorization systems in artificial intelligence and robotics are almost entirely image–based. Given a clear view of the object these disembodied classifiers can accurately categorize objects using visual appearance alone (Pinz, 2005). Because they do not use the robot's body, however, the functional properties of objects cannot be learned by these systems (Smith, 2005). Additional information sources are required for

*This chapter is a paper that will appear in the IEEE Transactions on Autonomous Mental Development (Griffith et al., 2011). This chapter combines methodology from our earlier work that appeared in ICDL 2009 (Griffith et al., 2009), ICRA 2010 (Griffith et al., 2010), and AAAI 2010 (Griffith and Stoytchev, 2010).

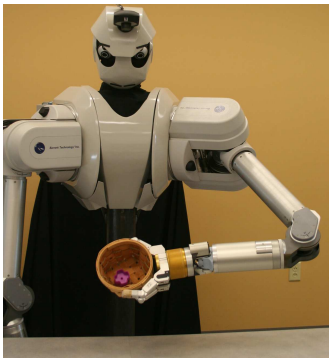


Figure 5.1 The upper-torso humanoid robot, shown here shaking one of the objects used in the experiments. The small plastic block inside the object produces auditory and visual events, which the robot can detect and use to categorize the object as a container.

learning object categories that capture something about the functional properties of objects.

The problem of learning object categories becomes more complex when multiple information sources are available. For example, consider a robot that has microphones and cameras, which record information streams while the robot interacts with objects. Objects that were traditionally categorized only by their visual appearance can now also be categorized by the sounds that they produce or by their movements as the robot performs different behaviors on them. To make things even more complicated, each behavior–modality combination results in a different object categorization. It is not straightforward to figure out which of these categorizations are more meaningful or if it is possible to combine them into a single categorization.

Research in developmental robotics has shown that robots can form meaningful behavior–grounded object categories using a single exploratory behavior and a single sensory modality (Griffith et al., 2009, 2010; Metta and Fitzpatrick, 2003). Because these categories are grounded in the robot's own behavior, the robot can test, verify, and correct that knowledge autonomously without human intervention (Sutton, 2001; Stoytchev, 2009). More work is needed, however, to show how a robot with an extensive behavioral and perceptual repertoire can reconcile the different object categorizations that result from each behavior–modality combination.

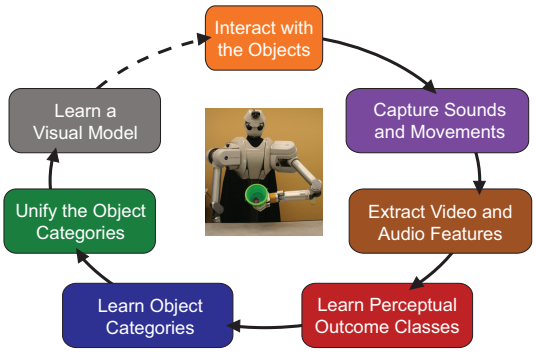


Figure 5.2 The framework used by the robot to learn object categories. First, the robot interacts with the objects and observes the outcomes that are produced. The extracted auditory and visual features are used to learn perceptual outcome classes. These are used to form object categories, one for each behavior–modality combination. The categories are unified using consensus clustering into a single category. Finally, a visual model is trained that can recognize the categories of novel objects. The dotted line is to show that the visual model could be used to guide and refine future interactions with objects.

This chapter introduces a computational framework that allows a robot to form a single behavior–grounded object categorization after it uses multiple exploratory behaviors to interact with objects and multiple sensory modalities to detect the outcomes that each behavior produces. Our robot (see Fig. 5.1) observed acoustic and visual outcomes from 6 different exploratory behaviors performed on 20 objects (containers and non-containers). Its task was to learn 12 different object categorizations (one for each behavior–modality combination), and then to unify these categorizations into a single one. In the end, the robot divided the objects into object categories that a human would call containers and non-containers. Furthermore, the robot was able to learn a visual model of the two categories and use this model to categorize novel objects.

Fig. 5.2 shows a high-level overview of the framework described in this chapter. First, the robot interacts with the objects and observes the sounds and the movement patterns that the objects produce. Perceptual features are learned from the raw sensory data, and feature extraction is performed. Next, the robot captures the different functional properties of the objects by clustering the extracted features into outcome classes. Object categories are learned by clustering the objects based on how often the different functional properties occur with each object. The object categorization step also includes a unification process, which unifies the

Figure 5.3 The objects used in the experiments. (**Containers**) The first two rows show the 10 container objects: wicker basket, metal trash can, potpourri basket, flower pot, bed riser, purple bucket, styrofoam bucket, car trash can, green bucket, and red bucket. (**Non-containers**) The second two rows show the same 10 objects as before but flipped upside down, which makes them non-containers for this particular robot with this particular set of behaviors.

object categories produced from multiple behaviors and sensory modalities. A visual model that can predict the categories of novel objects is learned in the last step. The visual classifier is trained using the object category labels produced by the unified clustering procedure. The visual classifier could also help guide and refine the robot's future interactions with objects.

5.1 Experimental Setup

5.1.1 Robot

The experiments described in this chapter were performed using the upper–torso humanoid robot shown in Fig. 5.1. Audio was captured using one of the microphones mounted in the robot's head. Video was captured using the ZCam, which was mounted on the top of the robot's head. See Chapter 3 for more details.

5.1.2 Objects

The robot interacted with a small plastic block and 10 different objects (shown in Fig. 5.3). Each of the 10 objects was a container in one orientation and a non-container when flipped over. Flipping the containers was an easy way for the robot to learn about non-containers while preserving the dimensions of the objects in the two categories.

The objects were selected to have a variety of shapes, sizes, and materials. Objects were tall, short, rectangular and round. They were made of plastic, metal, wicker, and foam. A few objects that were initially selected could not be used because they were too large to be grasped. Also, the aluminum fingers of the Barrett Hand did not create a firm grip with some objects, which was important for a large-scale experimental study like this one. Therefore, rubber fingers were stretched over each of the robot's three fingers to achieve more reliable grasps.

5.1.3 Robot Behaviors

The robot performed six behaviors during each trial: 1) *drop block*, 2) *grasp* object, 3) *move* object, 4) *shake* object, 5) *flip* object, and 6) *drop object*. Before the start of each trial a person placed the block and the object at specific locations. The robot grasped the block and positioned its hand in the area above the object before executing the six behaviors. Figure 5.4 shows the sequence of interactions for two separate trials (one with a container and one with a non-container). The individual behaviors are described below.

Drop Block Behavior: The height from which the robot dropped the block over the object was the same for all trials/objects. The drop positions were randomly selected from a 2D Gaussian distribution centered above the object in a plane parallel to the table. The standard deviation of this distribution was empirically set to be linearly proportional to the width (in pixels) of each object. Inverse kinematics was used to move the robot's hand to the drop position. Thus, the small block fell inside the container during approximately 70% of all trials with containers. During the other 30% of the trials with containers (and during trials with non-containers) the block fell on the table. In some cases the block rolled off the table

(approximately 5% of all trials). In these cases, the block was left off the table for the duration of the trial.

Dropping the block produced a lot of noise and large visual movements. During trials when the block fell into a container, however, the block moved less and made less noise.

Grasp Behavior: The robot grasped the object after dropping the block above it. Grasping the object produced little noise and only slightly moved the object. Thus, we expected that the categorizations resulting from this behavior would be somewhat less meaningful.

The robot failed to grasp the object in some cases (63 out of 2000 trials), which occurred when one of the robot's three fingers did not properly close. In these situations, a person monitoring the experiments recorded the error. All of the problematic trials were repeated after the initial round of experiments were completed.

Move Behavior: After grasping the object with its left hand, the robot moved the object toward the right side of its body. Moving the object produced little noise, as the object made little contact with the table and the block laid still either on the table or inside a container. So, we expected that the robot would only form meaningful categories based on its visual observations from this behavior.

Shake Behavior: The robot shook the object after moving it. Shaking took place well above the table to avoid banging the object into the table. Shaking the object caused a lot of movement and produced a lot of noise when the block was inside a container. During trials with non-containers, however, the behavior produced little noise and rarely caused co-movement between the block and the object. So, we expected that meaningful categorizations would be produced for this behavior.

Flip Behavior: The robot flipped the object over after shaking it. Flipping the object produced sounds only during trials in which the block was inside a container. During these trials, the block fell out of the container and crashed into the table. Thus, we expected that the robot would capture differences between containers and non-containers using this behavior.

Drop Object Behavior: The robot dropped the object after flipping it. Dropping the object always produced sounds and sometimes caused movement patterns between the block and the object. The acoustic outcomes and the visual movement patterns, however, were

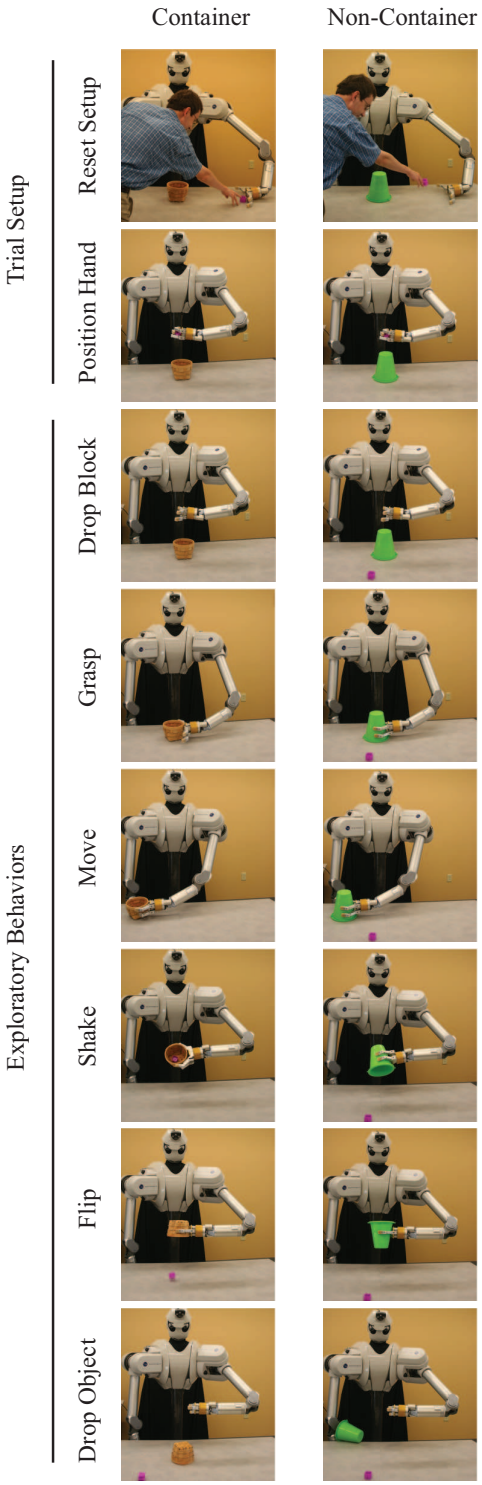


Figure 5.4 Snapshots from two separate trials with a container and a non-container object. Before each trial a human experimenter reset the setup by placing the block and the object at marked locations. After grasping the block and positioning its arm at a random location above the object the robot performed the six exploratory behaviors one after another.

seldom sufficient to discriminate containers from non-containers. So, we expected that the robot might capture differences in size or material properties using this behavior, but not functional differences.

5.2 Methodology

5.2.1 Data Collection

Multiple audio and video sequences were collected by the robot while it was performing the six exploratory behaviors, $\mathcal{B} = [$*drop block*, *grasp*, *move*, *shake*, *flip*, *drop object*. The six behaviors were organized into trials and always performed one after another (see Fig. 5.4). For each of the 20 objects, the robot performed 100 trials, for a total of $20 \times 100 = 2000$ trials. Because each trial consisted of 6 behaviors, the robot performed $6 \times 2000 = 12000$ behavioral interactions.

Another way to describe this dataset is to say that each behavior (e.g., *shake*) was performed 100 times on each of the 20 objects. Thus, each of the six behaviors was performed 2000 times. During every interaction the robot recorded the tuple (B, O, A, V), where $B \in \mathcal{B}$ was one of the six behaviors performed on object $O \in \mathcal{O}$, A was the recorded audio sequence, and V was the recorded video sequence. Audio data was sampled at 44.1 KHz over a 16-bit mono channel and stored as wave files. Visual data was captured from the robot's 3-D camera as a sequence of 640x480 color images and 320x240 depth images recorded at roughly 20 fps. The six behaviors lasted between 1 and 4 seconds each. *Drop object* and *grasp* took 1 second to complete; *drop block* and *flip* 2 seconds; *move* 3 seconds; and *shake* 4 seconds.

The order in which the robot interacted with the objects was chosen to minimize the effect of changing background noise. In a dataset of this magnitude, transient ambient noise can negatively impact the results (e.g., noise from the air conditioning system or computer fans). Therefore, the robot performed one trial with each of the twenty objects shown in Fig. 5.3 before moving on to the second trial with the first object, and so on.

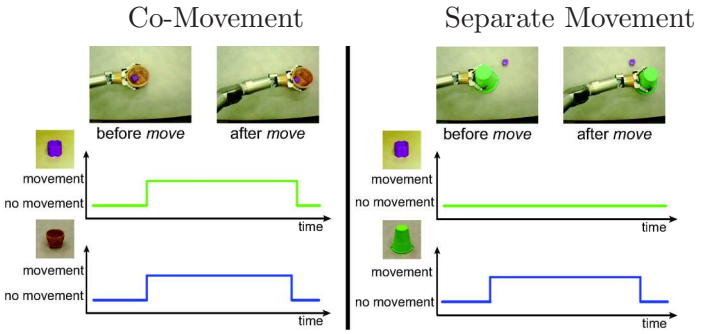


Figure 5.5 Transformation of the video data into movement sequences for two different executions of the *move* behavior. (**Left**) Co-movement was observed during trials in which the block moved when the object moved. Here, the block was inside a container and moved with it when the robot performed the *move* behavior. (**Right**) Separate movement outcomes occurred when the block fell to the side of a container or during trials with non-containers.

5.2.2 Movement Detection

The robot processed the frames from the ZCam to track the positions of the block and the object and to detect their movements. During each trial, the object was tracked using the center of mass of the largest blob with the corresponding color. The same was done for the block, which had a different color from the object. Movement was detected when the $[x, y]$ position of the block or the $[x, y]$ position of the object changed by more than a threshold, δ, over a short temporal window $[t', t'']$. The threshold, δ, was empirically set to 2.5 pixels per two consecutive frames. A box filter with a width of 3 was used to filter out noise in the movement detection data. The movement detection data for the block and the object from one behavioral interaction was used to create a movement sequence (see section 5.2.4). Figure 5.5 shows the sequence of detected movements of the block and the object for two different executions of the *move* behavior.

5.2.3 Auditory Feature Extraction

Auditory features were extracted automatically by representing the sounds produced by each behavioral interaction as a sequence of nodes in a Self-Organizing Map (SOM). The feature extraction process is the same as in our previous work (Sinapov et al., 2009). The three stage process includes: 1) a Discrete Fourier Transform which takes a 44.1 KHz audio

sample, A^i, and converts it to a 33 bin spectrogram, $P_i = [p_1^i, \ldots, p_l^i]$, where $p_j^i \in \mathbb{R}^{33}$ (the DFT window length was 26.6 ms, computed every 10 ms); 2) a 2D SOM that is trained with the spectrograms corresponding to one of the robot's six exploratory behaviors; and 3) a mapping, $\mathcal{M}(p_j^i) \rightarrow a_j^i$, of each spectrogram column vector, p_j^i, to the most highly activated state, a_j^i, in the SOM when p_j^i is presented as an input to the SOM (see Fig. 5.6). The mapping process results in a state sequence $A_i = a_1^i a_2^i \ldots a_{l^i}^i$, where each a_j^i stands for one of the SOM nodes. For each behavioral interaction, the corresponding SOM was trained using only 5% of the available column vectors (see Fig. 5.6), which were randomly selected from the spectrograms captured during this behavior.

The robot performed this procedure six times, once for every behavior. It acquired a set of state sequences, $\{A_i\}_{i=1}^{2000}$, for each of its six behaviors. This feature extraction method was chosen because it does not require a human to select the acoustic features that the robot will have to use. The algorithm identified and computed features in an unsupervised way. See (Sinapov et al., 2009) for further details.

5.2.4 Visual Feature Extraction

The robot extracted visual features using a procedure similar to that used for extracting auditory features (see Fig. 5.7). That is, visual features were extracted automatically by representing the movement sequences of the block and the object produced by each behavioral interaction as a sequence of nodes in a Self-Organizing Map (SOM). The three stage process includes: 1) a movement detection step which takes a recorded video sequence, V^i, captured at 20 frames per second, and converts it into a movement sequence, $M_i = [m_1^i, \ldots, m_l^i]$, where $m_j^i \in \mathbb{R}^2$; 2) a 2D SOM that is trained with the movement sequence corresponding to one of the robot's six exploratory behaviors; and 3) a mapping, $\mathcal{M}(m_j^i) \rightarrow v_j^i$, of each co-movement column vector, m_j^i, to the most highly activated state, v_j^i, in the SOM when m_j^i is presented as an input to the SOM (see Fig. 5.7). The mapping process results in a state sequence $V_i = v_1^i v_2^i \ldots v_{n^i}^i$, where each v_j^i stands for one of the SOM nodes.

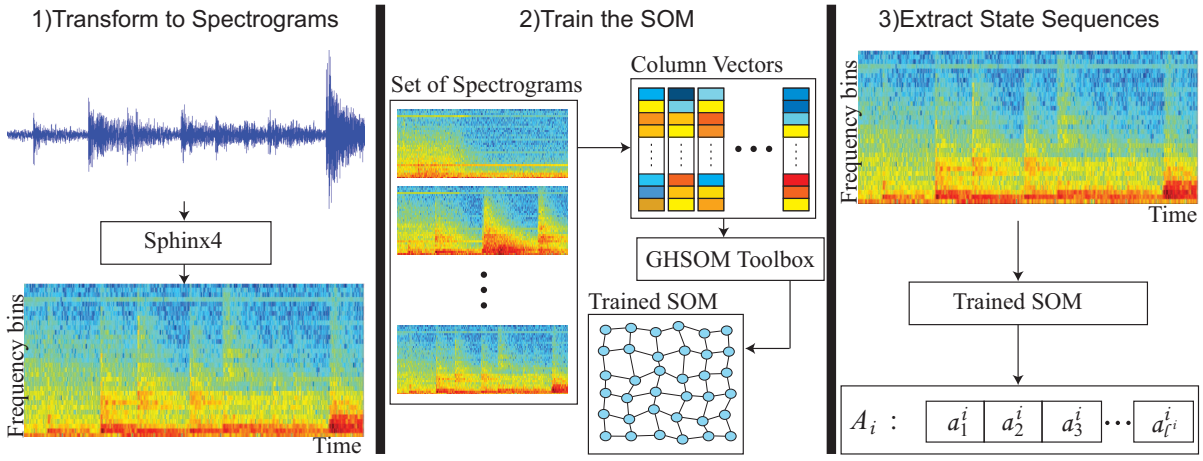


Figure 5.6 The feature extraction process for acoustic observations: 1) The raw sound wave produced by each behavior is transformed to a spectrogram. Each spectrogram has 33 bins (represented as column vectors), which capture the intensity of the audio signal for different frequencies at a given time slice. Red color indicates high intensity while blue color indicates low intensity. 2) An SOM is trained using randomly selected column vectors from the spectrograms for a given behavior. 3) The column vectors of each spectrogram are mapped to a discrete state sequence using the states of the SOM. Each column vector is mapped to the most highly activated SOM node when the column vector is used as an input to the SOM. See the text for more details.

Again, the robot performed this procedure six times, once for every behavior. It acquired a set of state sequences, $\{V_i\}_{i=1}^{2000}$, for each of its six behaviors. The parameters used for training each visual SOM were the same parameters used for training the acoustic SOMs. The only difference between the two feature extraction procedures was the size of the column vectors. The column vectors used to represent spectrograms had 33 rows; the column vectors used to represent co-movement sequences had 2 rows.

5.2.5 Learning Perceptual Outcome Classes

The acoustic outcome patterns produced by a given behavior can be clustered automatically to obtain auditory outcome classes. Similarly, the visual movement patterns produced by a given behavior can be clustered automatically to obtain visual outcome classes. In our case, the robot performed 6 behaviors and captured data from 2 modalities, so its task was to learn $6 \times 2 = 12$ separate sets of outcome classes. More formally, the robot learned k outcome classes

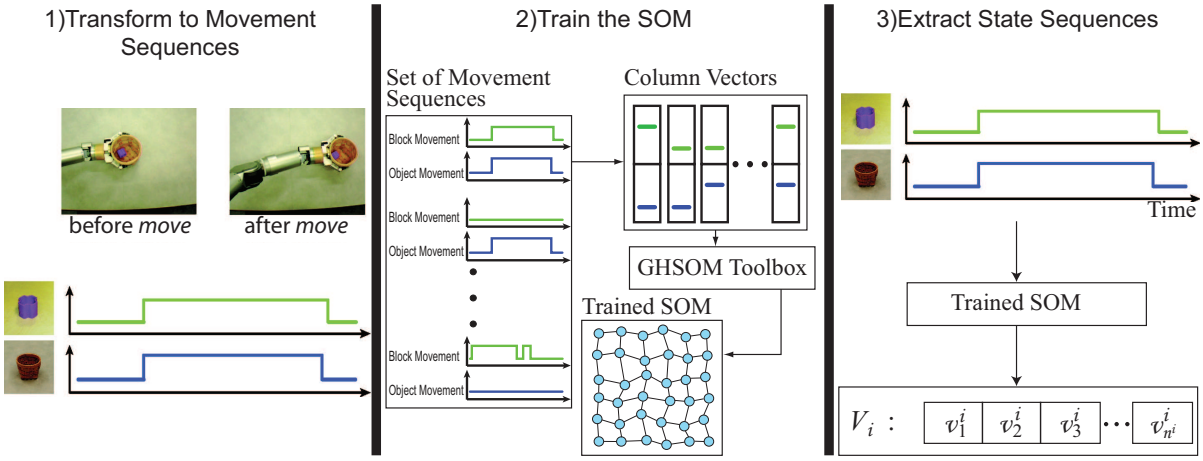


Figure 5.7 The feature extraction process for visual observations: 1) The video data recorded during each execution of a given behavior is transformed into a movement sequence. The co-movement sequence pictured here was obtained after the robot performed the *move* behavior with one of the containers. 2) An SOM is trained using randomly selected column vectors from the set of all movement sequences for a given behavior. 3) Each movement sequence is mapped to a discrete state sequence of SOM states. To do this, each column vector of the movement sequence is mapped to the most highly activated SOM node when the column vector is used as an input to the SOM. See the text for more details.

from the set of SOM state sequences, $\{A_i\}_{i=1}^{2000}$ or $\{V_i\}_{i=1}^{2000}$, observed for one modality during the execution of one of the 6 behaviors. An unsupervised hierarchical clustering procedure based on the *spectral clustering* algorithm was used for this task (spectral clustering is a similarity-based clustering algorithm (von Luxburg, 2007)). The procedure was performed 12 different times to obtain 6 different sets of acoustic outcome classes and 6 different sets of visual outcome classes. Figures 5.8 and 5.9 illustrate the process of learning acoustic outcome classes and visual outcome classes for one behavior, respectively.

The *spectral clustering* algorithm requires a similarity matrix as its input. The similarity between outcomes S_a and S_b, represented as sequences of SOM states produced by two different executions of the same behavior, was determined using the Needleman-Wunsch global alignment algorithm (Needleman and Wunsch, 1970; Navarro, 2001). The algorithm[1] can estimate the similarity between any two sequences if the data is represented as a sequence over a finite

[1]The Needleman-Wunsch algorithm maximizes the similarity between two sequences. An equivalent approach is to minimize the Levenshtein edit distance (Levenshtein, 1966) between the sequences.

alphabet. The general applicability of the algorithm has made it popular for other applications such as comparing biological sequences, text sequences, and more (Navarro, 2001). Computing the similarity of two sequences requires a substitution cost (i.e., a difference function) to be defined for any two tokens in the finite alphabet. Here the substitution cost is defined as the Euclidean distance between any two nodes in the SOM (each node in the 2D SOM has an x and a y coordinate).

The resulting similarity matrix, $\mathbf{W}$, was used as input to the unsupervised hierarchical clustering procedure, which partitions the input data points (i.e., either audio or video sequences) into disjoint clusters. The algorithm exploits the eigenstructure of the matrix to partition the data points. Finding the optimal graph partition is an NP-complete problem. Therefore, the Shi and Malik (2000) approximation algorithm was used, which minimizes the *normalized cut* objective function. The following steps give a summary of the algorithm:

1. Let $\mathbf{W}_{n\times n}$ be the symmetric matrix containing the similarity score for each pair of outcome sequences.

2. Let $\mathbf{D}_{n\times n}$ be the degree matrix of $\mathbf{W}$, i.e., a diagonal matrix such that $\mathbf{D}_{ii} = \sum_j W_{ij}$.

3. Solve the eigenvalue system $(\mathbf{D} - \mathbf{W})x = \lambda \mathbf{D}x$ for the eigenvector corresponding to the second smallest eigenvalue.

4. Search for a threshold of the resulting eigenvector to create a bi-partition of the set of acoustic (or visual) outcomes that minimizes the normalized cut objective function. Accept this bi-partition if the resulting value of the objective function is smaller than a threshold α.

5. Recursively bi-partition subgraphs obtained in step 4 that have at least β audio or video sequences.

The output of this procedure is k outcome classes $C = \{c_1, ..., c_k\}$, which are represented as the leaf nodes in a tree structure (see Fig. 5.8 and Fig. 5.9). In our previous work (Sinapov and Stoytchev, 2009), the value α used in step 4 was set to 0.995. The same value was used

here as well. The value for β used in step 5 was empirically set to 40% of the size of the dataset that was initially passed to the spectral clustering algorithm.

5.3 Object Categorization

5.3.1 Learning Object Categories

The frequency with which some outcomes occur with different objects can be used to cluster the objects into categories. For example, when the robot drops a block over a container, it will hear the sound of the block bouncing inside the container more often than when it drops the block over a non-container, in which case the block falls on the table. Similarly, when the robot moves a container, it will see the block move with the container more often than when it moves a non-container, in which case the block does not move.

Given a set of **outcome classes** $C = \{c_1, \ldots, c_k\}$ extracted by the robot while interacting with objects $\mathcal{O} = \{O_1, \ldots, O_{20}\}$, the robot acquired an outcome occurrence vector $E_u = [e_1^u, \ldots, e_k^u]$ for each object O_u. The value of each e_j^u represents the number of times the outcome c_j occurred with object O_u, divided by the total number of interactions (100 interactions in this case). In other words, each outcome occurrence vector E_u encodes a probability distribution over the set of outcome classes, such that e_j^u specifies the probability of observing outcome class c_j with object O_u over the entire history of interactions.

The robot formed **object categories** by clustering the feature vectors $E_1, \ldots, E_{20}$ (one for each of the 20 objects shown in Fig. 5.3). The X-means unsupervised clustering algorithm was used for the procedure. X-means extends the standard K-means algorithm to automatically estimate the correct number of clusters, k, in the dataset (Pelleg and Moore, 2000). Twelve different categorizations were constructed (one acoustic categorization and one visual categorization for each of the six exploratory behaviors).

5.3.2 Object Categorization Results

Figure 5.10 visualizes the twelve categorizations produced for each behavior–modality combination. The twelve categorizations are described in more detail below.

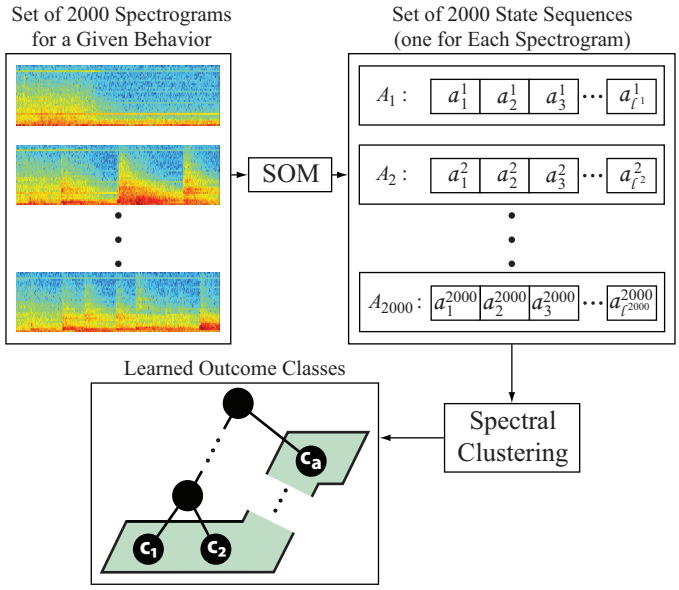


Figure 5.8 Illustration of the process used to learn acoustic outcome classes. Each spectrogram is transformed into a state sequence using the trained SOM, which results in 2000 sequences, $\{A_i\}_{i=1}^{2000}$, for each behavior. The acoustic outcome classes are learned by recursively applying the spectral clustering algorithm on this set of sequences. The acoustic outcome classes, $C = \{c_1, ..., c_a\}$, are the leaf nodes of the tree created by the recursive algorithm.

5.3.2.1 Acoustic Categorizations

Four of the six behaviors produced distinguishable acoustic signals that the robot could use to form object categories: *drop block*, *shake*, *flip*, and *drop object*. The (mostly silent) *grasp* and *move* behaviors produced acoustic signals that were very similar for all objects and the algorithm clustered all 20 objects into the same object class.

The *drop block* behavior produced three clusters that were almost homogeneous. The first cluster had only containers and the tall metal non-container (the only misclassified object). The second cluster had the rest of the non-containers. The last cluster had the three soft material container baskets. The difference between the softness and hardness of the objects' materials was distinctive enough to create two container categories (cluster 1 and 3 in Fig. 5.10). For example, the two wicker baskets and the styrofoam bucket (in cluster 3) are made of soft materials, which muffled the block's sound. In contrast, when the block fell into one of the hard containers (in cluster 1) it bounced around longer and produced a louder sound.

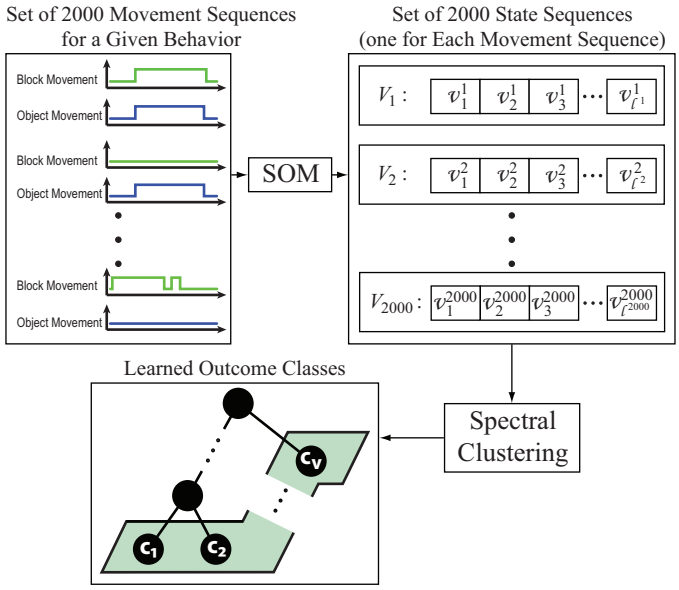


Figure 5.9 Illustration of the process used to learn visual outcome classes. Each movement sequence is transformed into a state sequence using the trained SOM, which results in 2000 state sequences, $\{V_i\}_{i=1}^{2000}$, for each behavior. The set of sequences is recursively bi-partitioned using the spectral clustering algorithm in order to learn visual outcome classes, $C = \{c_1, ..., c_v\}$, which are the leaf nodes of the tree created by the recursive algorithm.

The *shake* behavior produced results similar to the *drop block* behavior. In this case, however, there were only two clusters and the three soft-material container baskets were incorrectly classified as non-containers. These three objects produced very little sound when shaken, even if the block was inside them. Thus, they sounded similar to the non-containers, which seldom made noise during this interaction. The tall metal trash can was again misclassified.

The *flip* behavior was the most reliable way to discriminate between containers and non-containers in our experiments. It produced a perfect classification. Flipping the object over produced a distinct sound in the case of containers as the small block fell onto the table. In the case of non-containers, no sound was generated as the block was already on the table.

The *drop object* behavior resulted in clusters that were completely heterogeneous. The behavior did not produce different acoustic outcomes for containers and non-containers.

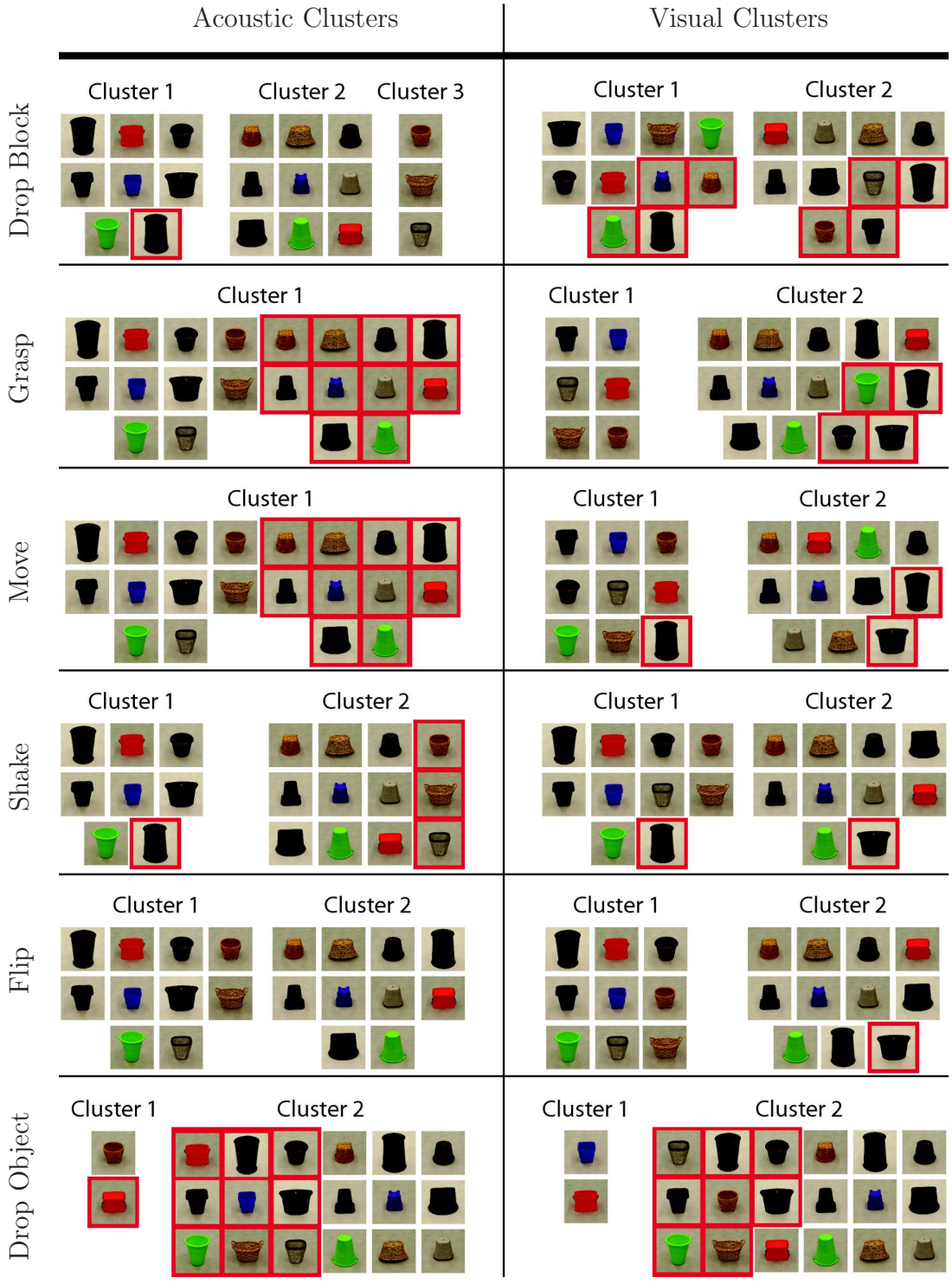


Figure 5.10 Visualization of the object categories formed by the robot for the six exploratory behaviors and the two sensory modalities. Incorrect classifications are framed in red (based on category labels provided by a human and the majority class of the cluster). The quality of each categorization depends on the behavior that was performed and the sensory modality that was used for clustering.

5.3.2.2 Visual Categorizations

All six behaviors produced visual movement patterns that the robot could use for object categorization (see Fig. 5.10). The *drop block* behavior was not a reliable way to categorize containers from non-containers. The categorization resulted in two noisy clusters, in which the robot incorrectly classified four containers and four non-containers. The categorization was similar to a random separation of the objects.

The *grasp* behavior resulted in a categorization with two clusters. Six containers were clustered together and the rest of the objects were classified to the other cluster. The behavior was more useful than expected because it generated a tiny amount of movement. However, in some trials the duration of movement was so short that it was filtered out. Four containers were misclassified due to this noisy data.

The *move* behavior produced a good categorization of containers and non-containers. Only three objects were misclassified: the metal non-container was incorrectly classified as a container; the tall metal trash can and the car trash can were incorrectly classified as non-containers. Each of the three objects has a unique shape, which may help to explain why the objects were misclassified. For example, the metal non-container sometimes functioned as a container since it had a 3/4" lip that could cause the block to come to rest on top of the object. Subsequently, during the *move* behavior the block frequently co-moved with the metal non-container.

The *shake* behavior produced results slightly better than the *move* behavior. Only two objects were misclassified in this case. Shaking the containers produced slight oscillations in the position of the block and the containers when the block was inside them, which allowed the robot to form a good categorization. The skinny car trash can was misclassified probably due to its width—it more readily occluded the block as it was shaken. The narrow shape also kept the block from falling inside the container as often as it fell inside the other containers.

The *flip* behavior produced a near–perfect classification of the objects. Flipping the object over produced a lot of block movement during trials when the block fell out of the containers. In all other trials, the block did not move. The skinny car trash can was again misclassified.

The block *appeared* to move, however, during several trials with the green non-container, which is why it was misclassified as a container. The block often came to rest at the perimeter of the visual field where the depth position fluctuated during these trials.

The *drop object* behavior was not a reliable way to categorize containers and non-containers. The behavior led to two arbitrary clusters. The red bucket and the purple bucket were classified together. The rest of the objects were placed in the other cluster.

5.3.3 Evaluating the Object Categorizations

To check whether the robot was able to extract meaningful object clusters we computed the category information gain. The information gain captures how well the object categories formed by the robot resemble the categories specified by a human. The information gain is high when the category labels assigned to the objects match human-provided category labels. It is low otherwise. In other words, if the information gain is high, then the robot has categorized the objects in a meaningful way (even though the robot does not know the human words corresponding to the categories).

Let $\lambda^{(f)} = [\mathcal{O}^1, \ldots, \mathcal{O}^{M_f}]$ define an object categorization over the set of objects $\mathcal{O}$, for a specific behavior–modality combination B_f, where $\mathcal{O}^i$ is the set of objects in the i^{th} cluster. Let p_c^i and p_{nc}^i be the estimated probabilities that an object drawn from the subset $\mathcal{O}^i$ will be a container or a non-container (as defined by human labels). Given a cluster of objects $\mathcal{O}^i$, the Shannon entropy of the cluster is defined as:

$$\mathcal{H}(\mathcal{O}^i) = -p_c^i log_2(p_c^i) - p_{nc}^i log_2(p_{nc}^i)$$

In other words, an object cluster containing mostly containers (or mostly non-containers) will have low entropy, while a cluster containing an equal number of containers and non-containers will have the maximum entropy. The information gain for the object categorization $\lambda^{(f)} = [\mathcal{O}^1, \ldots, \mathcal{O}^{M_f}]$, which was learned using behavior-modality combination B_f, is given by the following formula:

$$IG(\lambda^{(f)}) = \mathcal{H}(\mathcal{O}) - \sum_{i=1}^{M_f} \frac{|\mathcal{O}^i|}{|\mathcal{O}|} \mathcal{H}(\mathcal{O}^i)$$

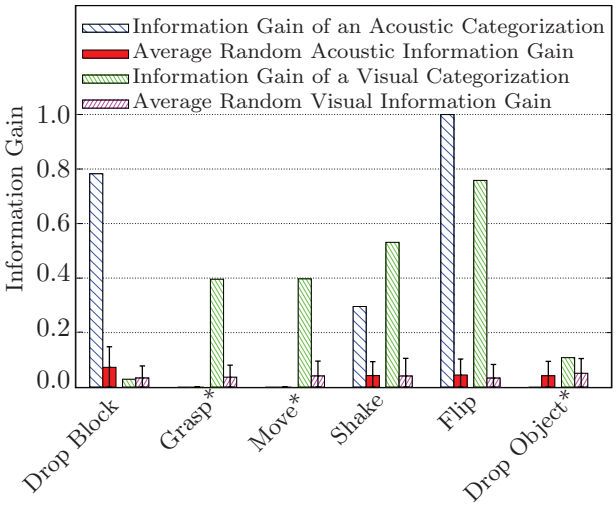


Figure 5.11 Information gain of the object categories formed by the robot for each behavior–modality combination. For comparison, the information gain for a random classification is shown next to the object category information gain. The random information gain was computed by shuffling the labels 100 times and estimating the mean and the standard deviation. When computing the information gain, the correct object labels (container or non-container) were provided by a human. For some behaviors the acoustic information gain was zero, which is denoted with the * symbol.

To get a baseline information gain value for comparison, the information gain was computed for a random labeling. That is, the values for p_c^i and p_{nc}^i were estimated after randomly shuffling the labels of the objects in all clusters $\mathcal{O}^i$ (where $i = 1$ to M_f) while preserving the number of objects in each cluster. This procedure was repeated 100 times to estimate the mean and the standard deviation. Figure 5.11 shows the information gain for each categorization and compares it to the corresponding baseline average random information gain.

The figure shows that the categorization produced using acoustic outcomes from the *flip* behavior most closely matches the labels provided by an adult human. Next in order are: the categorization produced using the acoustic signals from the *drop block* behavior, the categorizations produced using the visual movement patterns from the *flip*, *shake*, *grasp* and *move* behaviors, and the categorization produced using the acoustic signals from the *shake* behavior. All of these categorizations have an information gain that is better than chance. The remaining categorizations have an insignificant information gain with respect to the human-provided labels, which shows that they are not suitable for capturing the functional properties

of containers.

The fact that some clusterings formed by the robot were noisy was expected. Some behaviors are simply better at capturing certain object properties than others. With 20 objects of various shapes, sizes, and materials there are many ways the robot could have categorized the objects. However, no behavior completely separated objects by size or material. On the other hand, seven behavior–modality combinations captured the functional properties of the containers well (i.e., acoustic signals from the *drop block*, *shake*, and *flip* behaviors; and visual movement patterns from the *grasp*, *move*, *shake*, and *flip* behaviors). The next section shows how the different categorizations can be combined into a single one.

5.4 Unified Object Categorization

5.4.1 Unification Algorithm

As Fig. 5.10 shows, by categorizing objects using multiple behaviors and multiple modalities, the robot can form many different categorizations of the objects. Some categorizations closely match the object labels provided by a human; others are noisy. Without a method to unify the different categorizations of the objects, however, an object categorization is *at most* meaningful with respect to the behavior and the modality that were used to produce it.

Therefore, it is desirable to form one unified categorization from multiple categorizations of the objects. That is, given a set of object categorizations $\Lambda = \lambda^{(1)}, \ldots, \lambda^{(r)}$ and a desired number of object categories p, the robot forms a single, unified categorization $\hat{\lambda}$. The categorization $\hat{\lambda}$ defines p categories of objects and is determined to be representative of the input categorizations Λ using the objective function $\phi(\Lambda,\hat{\lambda})$. The function measures the total normalized mutual information between a set Λ containing r object categorizations and a single categorization $\hat{\lambda}$. More formally,

$$\phi(\Lambda, \hat{\lambda}) = \sum_{q=1}^{r} \phi^{NMI}(\hat{\lambda}, \lambda^{(q)})$$

where $\phi^{NMI}(\hat{\lambda}, \lambda^{(q)})$ is the normalized mutual information between categorizations $\hat{\lambda}$ and $\lambda^{(q)}$ (see (Strehl and Ghosh, 2002)). Thus, the best unified categorization is defined as the clustering

of the objects that has the highest possible total normalized mutual information with respect to the multiple input categorizations. Finding the best clustering, however, is intractable. Therefore, it is necessary to search for a clustering that is approximately the best. For this task, we used the hard consensus clustering algorithm (Strehl and Ghosh, 2002). The algorithm takes as input a set of object categorizations Λ and a value k, employs three functions that independently solve for a good approximation, and outputs the best unified clustering that it finds. The output of this procedure is a labeling $L_i \in \mathcal{L}$ for each object $O_i \in \mathcal{O}$.

In this case, the set of object categorizations Λ consisted of the twelve categorizations shown in Figure 5.10. The algorithm was run several times with p varying from 2 to 10. From these runs, the unified object categorization was chosen as the clustering that maximized the objective function. The result of unifying the twelve object categorizations is shown in Fig. 5.12. The figure shows that the hard consensus clustering algorithm was able to find a meaningful categorization even though only seven of the twelve behavior–modality combinations produced a good clustering of the objects. Only one object was misclassified in the unified object categorization.

For completeness, a brief description of the three functions used by the hard consensus clustering algorithm is provided below. The algorithm runs these functions in parallel and picks one of the category results that maximizes the NMI. 1) The Cluster–based Similarity Partitioning Algorithm (CSPA) generates a similarity matrix for the objects. Each entry in this matrix represents the number of times that two objects appear in the same cluster. A similarity–based clustering algorithm is applied to this matrix to cluster the objects. 2) The HyperGraph Partitioning Algorithm (HGPA) constructs a hypergraph and partitions it into k disjoint components by cutting a minimal number of hyperedges. A hypergraph is a special graph in which an edge can connect to many vertices. In our case, the objects are the vertices of the hypergraph and the clusters of objects are the edges. 3) The Meta-CLustering Algorithm (MCLA) groups multiple similar clusters of objects until there are at most k disjoint clusters. For more details see (Strehl and Ghosh, 2002).

5.4.2 Robustness of the Algorithm

To test the generalizability properties of the algorithm we ran three additional experiments that are briefly summarized below. In the first experiment, the consensus clustering algorithm was able to form a meaningful categorization when the object classes were skewed to have more containers than non-containers. The set of objects was skewed by using only 4 of the 10 non-containers. The robot categorized the objects using the same learning framework. This process was repeated 10 times with different sets of 4 randomly chosen non-containers. The algorithm misclassified 2 objects in 8 of these instances and 1 object in another instance. In the second experiment, the interaction data for one random container and one random non-container was removed for each behaviormodality combination. The robot categorized the objects using the same learning framework, and the process was repeated 10 times with different sets of removed data. All resulting unified categorizations matched the categorization shown in Fig. 5.12. Thus, the algorithm was able to form a meaningful categorization even when some of the interaction data was missing. The third experiment tested an alternative approach to forming a unified object categorization by directly concatenating and clustering the feature vectors used to produce the individual categorizations (see section 5.3). The X-means algorithm was used to do the clustering, which produced three clusters with two misclassified objects. This result is inferior to the unified categorization shown in Fig. 5.12. Overall, the algorithm proved to be quite robust.

By combining the different categorizations into a single one, the robot effectively ruled out the nonsense categorizations that it acquired, allowing it to form two object categories that are close to what a human would call containers and non-containers. Furthermore, the unified categorization condensed a large amount of data into a single categorization, which described the functional properties of objects across the robot's whole sensorimotor repertoire. Having a single categorization also meant that a single perceptual model could be learned, and used to infer the object category of novel objects using only passive observation. The next section describes how the robot was able to form a perceptual model for the two object categories shown in Fig. 5.12.

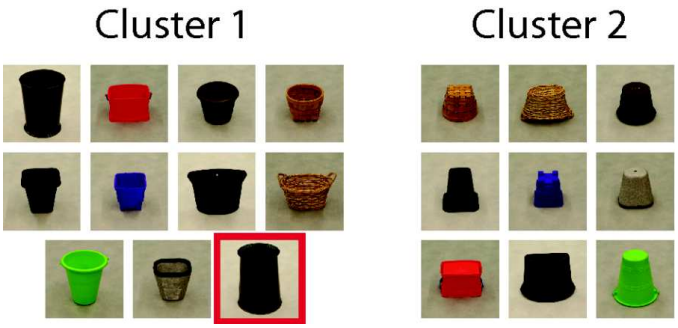


Figure 5.12 Visualization of the unified object categorization produced by the consensus clustering algorithm, which searched for a consolidated clustering of the twelve input clusterings shown in Fig. 5.10. The unified categorization closely matches ground-truth labels provided by a human. Only one object was misclassified.

5.5 Categorizing Novel Objects

It is impractical for a robot to categorize all novel objects by first interacting with them for a long time. To reduce the exploration time, the robot can learn a perceptual model of each acquired object category in the unified object categorization (see Fig. 5.12) and use that model to estimate the category of a novel object. More specifically, let $\mathbf{f_i} \in \mathbb{R}^n$ be the visual feature vector for object O_i, and let $L_i \in \mathcal{L}$ be the category label of that object according to the learned unified categorization, where $\mathcal{L}$ is the set of object categories. Given training examples $(\mathbf{f_i}, L_i)_{i=1}^{i=N}$, the task of the robot is to learn a recognition model $\mathcal{M}$ that can estimate the correct category of a novel object O_{test} given the object's visual features $\mathbf{f_{test}}$. In other words, $\mathcal{M}(\mathbf{f_{test}}) \rightarrow L_{test}$, where $L_{test} \in \mathcal{L}$ is the estimated category of the novel object. The next subsection describes the feature extraction routine used to compute the visual features $\mathbf{f_i} \in \mathbb{R}^n$ for both familiar and novel objects.

5.5.1 Feature Extraction

To extract the visual features of objects, principal component analysis (PCA) was used to find compact representations for the unlabeled visual sensory stimuli. PCA transforms the input data into a new coordinate system, where each coordinate represents a different projection of the input data. The coordinates are ordered based on how well the projections

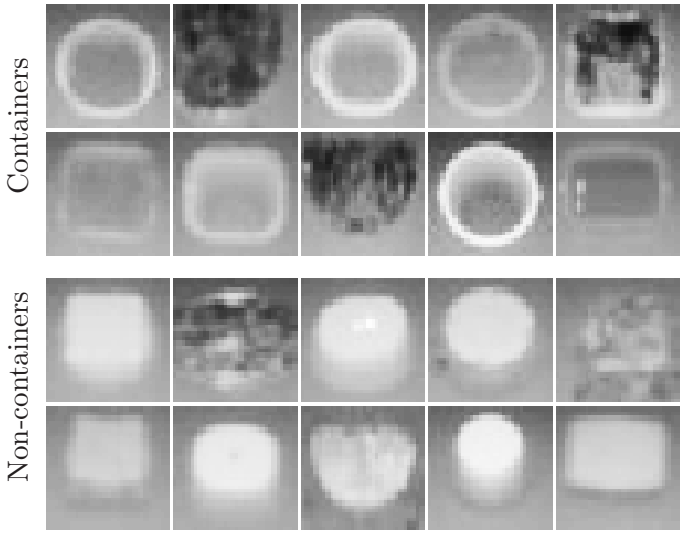


Figure 5.13 The 20 depth images of the objects used as input to the sparse coding algorithm. Each image was generated by finding the object in the larger 320x240 depth image and scaling the region to 30x30 pixels.

explain the variance in the data. More formally, the input images $\mathbf{x_i} \in \mathbb{R}^m$ are transformed into a set of independent basis vectors $\mathbf{b_1}, \ldots, \mathbf{b_n} \in \mathbb{R}^m$ and a vector of weights $\mathbf{f_i} \in \mathbb{R}^n$ such that $\mathbf{x_i} - \bar{\mathbf{x}} \approx \sum_j \mathbf{b_j} f_i^j$, where $\bar{\mathbf{x}}$ is the mean of all input images. The weights $\mathbf{f_i} \in \mathbb{R}^n$ represent the compact features of the high-dimensional input image $\mathbf{x_i}$.

The algorithm was trained on 30x30 depth images, one for each of the 20 objects that the robot interacted with (see Fig. 5.13). The training images were extracted automatically from the larger 320x240 depth images captured by the ZCam. The objects were located using background subtraction and a boundary box was placed around them. The corresponding locations in the depth image were cropped and scaled to 30x30 pixels. The resulting images were used as input to the PCA algorithm. The first five basis vectors computed by the algorithm captured 90% of the variance in the data and are shown in Fig. 5.14. The figure shows that the first vector, which captures 43% of the variance, is a convex feature characteristic of non-containers. The second and the third vectors, which jointly capture 40% of the variance, represent a feature characteristic of containers. The next subsection describes the recognition algorithm that maps the visual features of an object to its estimated object category.

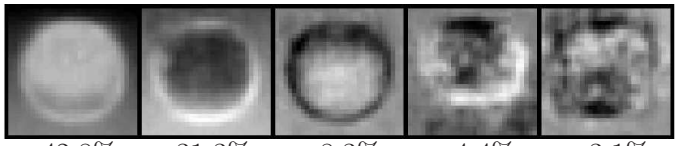


Figure 5.14 A visualization of the first five principal components computed by the PCA algorithm using the images shown in Fig. 5.13 as input. The percentage of the variance explained by each component is listed below it. These five principal components, along with the category labels from Fig. 5.12, were later used to classify novel objects as 'containers' or 'non-containers.'

5.5.2 Recognition Algorithm

The object category recognition model $\mathcal{M}$ was trained on the visual features of the 20 objects that the robot interacted with. Let $\mathbf{f_i} \in \mathbb{R}^2$ represent the extracted visual features for the i^{th} object, and let $L_i \in \mathcal{L}$ be its category according to the unified categorization (see Fig. 5.12). Using this formulation, the robot acquired the set $(\mathbf{f_i}, L_i)_{i=1}^{20}$, which contains the 20 labeled training examples available to it.

The robot's recognition model, $\mathcal{M}$, was implemented as a k-Nearest Neighbors (k-NN) classifier with $k = 3$. K-NN is an instance-based learning algorithm that does not build an explicit model of the data, but simply stores all labeled data points and uses them when the model is queried to make a prediction. Given a novel object, O_{test}, the robot extracted its visual features $\mathbf{f_{test}}$, computed from a 30 x 30 depth image of the object, and the learned basis vectors. Subsequently, k-NN was used to find the k closest neighbors of $\mathbf{f_{test}}$ in the training set, using the Euclidean distance function. Finally, the novel object was labeled with the majority category of the k closest neighbors. For example, if 2 of the closest neighbors to $\mathbf{f_{test}}$ were containers, then the novel object was labeled as a container as well.

The classifier was tested on how well it could detect the object category of 30 novel objects by passively observing them. The set of novel objects included 15 containers, which were selected to have a variety of shapes, sizes, and material properties. The other 15 objects were non-containers, which were the same novel containers only flipped over (see Fig. 5.15). Using the extracted visual features and the k-Nearest Neighbor classifier, the robot assigned the correct object category to 29 of the 30 objects. This result implies that the robot not only has

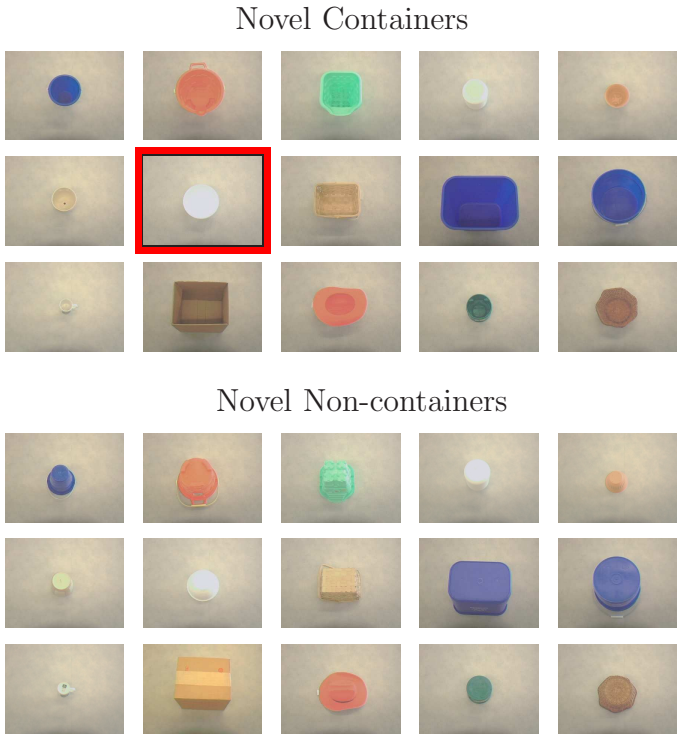


Figure 5.15 The result of using a Nearest Neighbor classifier to label novel objects as 'containers' or 'non-containers'. The mixing bowl (outlined in red) was the only misclassified object. Visual features were extracted for each of the 30 novel objects and used in the classification procedure.

the ability to interactively distinguish between containers and non-containers, but also to learn a visual model that allows it to passively determine the functional category of novel objects.

5.6 Evaluating the Effect of Experience on the Quality of Object Categorizations

Intuitively, the quality of an object categorization should depend on how much experience the robot has had with each object. As Fig. 5.11 shows, the robot formed seven meaningful categorizations after 100 interactions were performed with each object. The unification of all twelve categorizations also produced a meaningful categorization, in which only one object was misclassified. Even fewer interactions, however, may be required to reproduce these results.

To find out how much experience is necessary to form a good object categorization, the categorization quality was evaluated as the number of interactions, N, with each object was

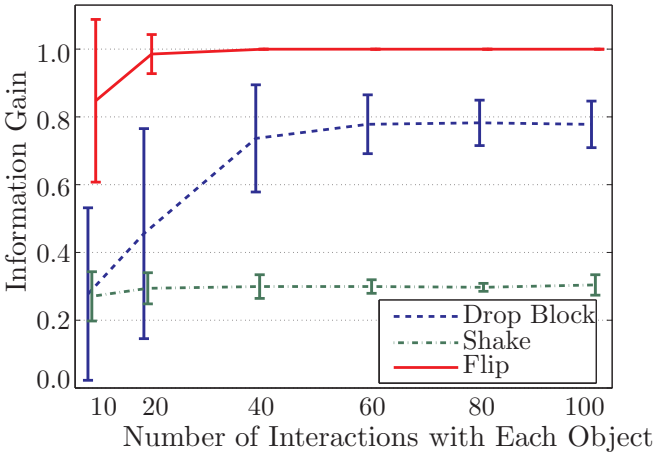


Figure 5.16 Information gain for the acoustic categorizations formed by the *drop block*, *shake*, and *flip* behaviors as the number of interactions with each object is increased. This graph was computed by randomly sampling N interactions from the 100 interactions with each object and re-running the learning algorithms on the smaller dataset. This process was repeated 100 times for each value of N to estimate the mean and standard deviation. Human-provided category labels were used to compute the information gain.

increased from 10 to 100. The learning framework described in sections IV and V was used to produce the categorizations for this evaluation (i.e., the same learning framework used to produce the results in Fig. 5.10, except that the trained SOM was reused to reduce computation time)[2]. The set of N trials used to compute a categorization was randomly sampled from the set of all 100 trials performed on each object. The quality of a categorization was determined by computing its information gain using human–provided labels. The process was repeated 100 times for each value of N to estimate the mean and the standard deviation. Although only seven out of twelve behavior–modality combinations originally led to a meaningful object categorization, the unification procedure was performed on all twelve combinations. The quality of the resulting unified clustering was also evaluated using its information gain.

The results are shown in two graphs to simplify their analysis. Figure 5.16 shows the quality

[2]It is important to point out that the trained SOM represents the robots self–organized "feature extraction" mechanism. This does not have to be part of the robots "object categorization" mechanism. In fact, evidence from developmental psychology suggests that the auditory features that infants learn are fixed by the time they learn words. When infants are around six months old they are sensitive to the sounds that are used in many different languages. By nine months, however, they have learned a fixed set of auditory features, which are specific to their native language (Jusczyk et al., 1993). For example, at this age, it would be more difficult for an infant raised in an English–speaking home to distinguish between some of the sounds that an infant raised in a Chinese–speaking home can distinguish without a problem.

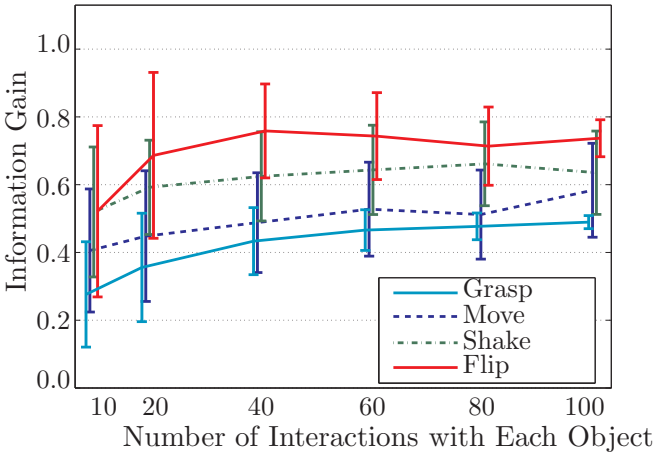


Figure 5.17 Information gain for the visual categorizations formed by the *grasp*, *move*, *shake*, and *flip* behaviors as the number of interactions with each object is increased. This graph was computed using the same procedure as that described in Fig. 5.16.

of the acoustic categorizations for the *drop block*, *shake*, and *flip* behaviors. Figure 5.17 shows the quality of the visual categorizations for the *grasp*, *move*, *shake*, and *flip* behaviors. The other five behavior–modality combinations are not depicted in the graphs because their information gain remained near zero. Also not shown is the quality of the unified categorization, which remained fairly constant at 0.75 as the value for N increased from 10 to 100.

The mean information gain for all of the categorizations converged after about 40 interactions with each object were performed. The quality of the individual categorizations increased as the robot gained more experience. The mean information gain converged when the features used to represent the functional properties of each object stabilized. In contrast with the individual categorizations, the information gain of the unified categorization converged after only 10 interactions with each object. Thus, the unified categorization was meaningful even when the robot had an insufficient amount of data to fully characterize the functional properties of each object for the individual behavior–modality combinations.

The effect that the order–dependent clustering algorithms had on the categorization performance is most clear when the number of interactions, N, with each object is 100. Instead of consistently identifying the same categorization of the objects for each behavior–modality combination, the framework identified a distribution of categorizations. The histograms in Fig. 5.18

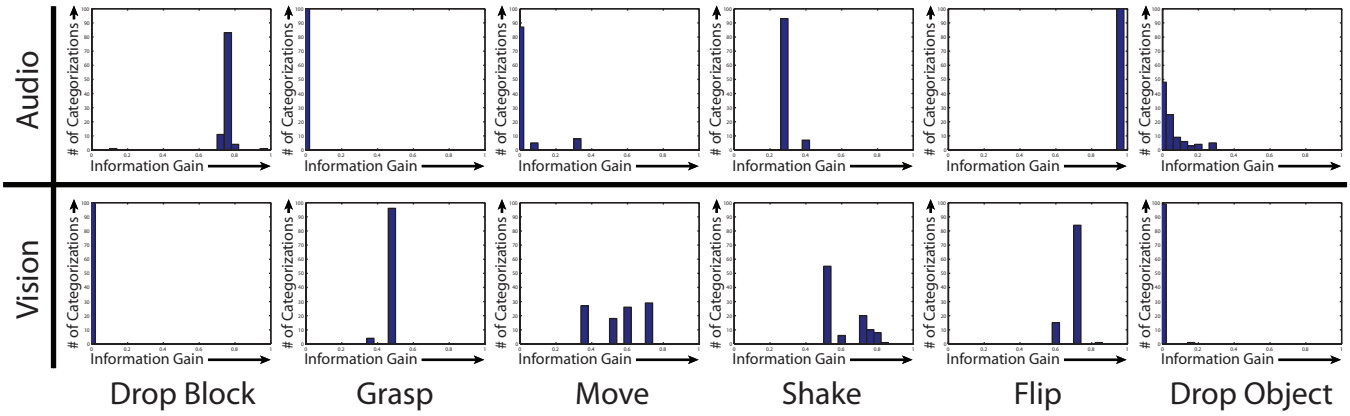


Figure 5.18 The distribution of information gain values for different categorizations obtained with different behavior–modality combinations. Each histogram was generated by computing the information gain values for 100 different categorizations of the objects, which were obtained by running the framework 100 times on different orderings of the dataset.

show that certain behavior–modality combinations were affected more than others. For example, the *grasp–vision* behavior–modality combination was only slightly affected (the flower pot object was misclassified as a non-container 4 times; it was correctly classified as a container 96 times). The *move–vision* behavior–modality combination was significantly affected, however, as three objects oscillated between categories. This analysis was also performed for the unified categorization, where 62 out of 100 categorizations matched the results obtained using the original ordering of the dataset. In general, if the functional properties of an object placed it somewhere between containers and non-containers, then the resulting category label for this object tended to fluctuate. Fluctuations in the individual categorizations seldom reduced the quality of the unified categorization.

5.7 Summary

This chapter described a computational framework for learning object categories, in which a robot explored objects using multiple behaviors and sensed the resulting outcomes using multiple sensory modalities. The framework was evaluated using an object categorization task with 20 containers and non-containers. The robot observed the acoustic signatures and the visual movement patterns of the objects as it performed six different exploratory behaviors. A

separate object categorization was produced for each behavior–modality combination, which resulted in twelve different categorizations of the objects. These categorizations were then unified using consensus clustering into a single object categorization. It was shown that this behavior–grounded object categorization is meaningful when compared with human–provided object labels. It was also shown that this level of categorization performance was attainable after only 10 interactions were performed with each object for each behavior–modality combination. Finally, this chapter also showed that a visual classifier can effectively categorize novel objects when it is trained using the category label for each object.

This is the first framework in which an object categorization formed by a robot was constructed by creating many different categorizations for a set of objects, which correspond to different behavior–modality combinations, and then unifying them into a single one. Our methodology is consistent with Leslie Cohen's definition that object categorization is about finding similarities among perceptually different objects (Cohen, 2003); whereas object recognition is about finding differences among perceptually similar objects (DiCarlo and Cox, 2007). The results showed that some of the perceptual differences (e.g., softness) between the objects were captured by the individual categorizations formed by the robot. However, by unifying many different individual categorizations, the robot ignored these perceptual differences and formed a categorization based only on the containment property, which was the most common thing between the objects.

In the end, the experience that the robot acquired in this large–scale experiment was condensed into a single object categorization. The robot had knowledge of the functional properties of each object in terms of the frequency with which different acoustic outcomes and different visual movement patterns occurred with it. The robot also knew the differences between the objects in terms of this frequency information, which served as the basis for categorizing the objects. Finally, the robot knew the visual appearance of containers and non-containers. Having the option to categorize an object by either its functional properties or its visual appearance is advantageous, and mirrors some of the characteristics of object categorization in humans (Lederman and Klatzky, 1987).

The framework presented here can be extended in several possible directions. One possible extension is to reduce the human input provided to the object categorization framework. For instance, the object IDs were provided by a human and used during the categorization procedure. It may be possible to use object recognition models to eliminate the dependency on human–provided object IDs. It is also desirable to let the robot learn its own exploratory behaviors. In the current framework, the behaviors were encoded by a human programmer. Presumably, it should be possible for a robot to learn these behaviors on its own.

Another possible extension of this work is to make the framework capable of categorizing many different types of objects. Intuitively, finding a meaningful categorization for a large number of object types would require the robot to have an increased amount of experience with each object. The robot could use more sensory modalities with each behavior, however, to reduce the amount of experience that is required. For example, tactile and proprioceptive sensory modalities could be added in order to capture more information during each interaction. Previous work from our lab has shown that by adding more sensory modalities the robot could improve its object recognition abilities (Sinapov and Stoytchev, 2010). The same is probably true for object category recognition.

CHAPTER 6. SEPARATING CONTAINERS FROM NON-CONTAINERS USING SEQUENCES OF MOVEMENT DEPENDENCY GRAPHS

One of the first things that infants learn about containers is that an object inside a container will move with the container when the container is moved (Hespos and Spelke, 2007). Infants easily identify the movement dependencies between the visual features of objects, which provides a powerful cue for learning to manipulate them (Spelke, 1994). Infants also learn how their own movements relate to the movements of objects. They become fascinated with making and breaking co-movement relationships as they insert blocks into containers and then shake them (Largo and Howard, 1979b). Clearly, a lot of information about objects can be gained by observing their co-movement patterns. The ability to track movement dependencies between objects can mitigate some of the perceptual issues that arise during manipulation.

This chapter describes a new representation for movement dependencies between objects during manipulation. This representation uses sequences of graphs (see Fig. 6.1). The vertices in these graphs correspond to the tracked features of objects. The edges indicate movement dependencies between the features. A statistical test for independence is used to determine when to insert/delete edges into the graphs. These graphs evolve over time and form sequences of graphs that reflect how movement dependencies change as the robot manipulates different objects.

The representation was evaluated in an experiment with 20 objects and 5 blocks. The robot observed the movement patterns of the objects as it interacted with them. A sequence of movement dependency graphs was extracted from the movement patterns and used to categorize the objects. The results show that the robot could use the movement dependency graph

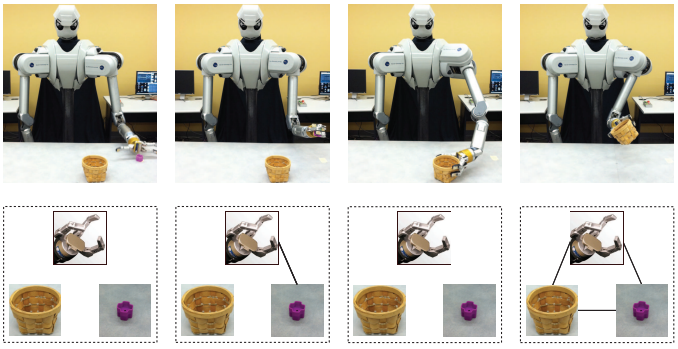


Figure 6.1 **(Top Row)** Our humanoid robot, shown here grasping a small block, shaking it, dropping it inside a container, grasping the container, and shaking the container. **(Bottom Row)** The movement dependency graph as it evolved over time. The nodes correspond to the entities that are tracked visually. The edges between pairs of objects indicate movement dependencies.

representation to form a representation for what a human would call 'containers' and 'non-containers'.

6.1 Related Work

This section covers additional related work that was not covered in Chapter 2 that is specifically related to graphs and graph–based representations.

Approaches based on co-movement representations have been used previously for object categorization tasks (Griffith et al., 2009, 2011). Because these approaches focus specifically on learning about objects, their goal is to represent the overall outcome of an interaction rather than to represent the details of a behavioral interaction. Thus, these approaches lack the ability to identify exactly when co-movement or separate movement occurs during manipulation. This chapter, in contrast, proposes a representation that has this ability.

Movement representations have also been used in previous work to detect visual features attached to the robot's body (Gold and Scassellati, 2009; Stoytchev, 2011; Metta and Fitzpatrick, 2003; Kemp and Edsinger, 2006a). Because these approaches focus on detecting the body, however, they require proprioceptive data: timestamps of motor commands (Gold and Scassellati, 2009; Stoytchev, 2011) or vectors of joint angles (Kemp and Edsinger, 2006a). Using proprioceptive data some robots were also able to learn from the movement patterns of objects,

but only if the robot directly controlled the object (Metta and Fitzpatrick, 2003; Kemp and Edsinger, 2006a). In contrast, this chapter describes a representation that does not require proprioceptive data and can be applied using vision alone.

Aksoy et al. (2010) have shown that a graph representation works well for activity recognition and object categorization. They introduced semantic scene graphs, which can capture the spatiotemporal relationships between many different objects in an image. Each node in the graph represents an object and each edge represents one of four different spatial relationships between two objects: absence, no connection, touching, and overlapping. A sequence of semantic scene graphs can capture enough information for activity recognition or object categorization. This representation, however, does not capture co-movement relationships between the objects.

Sridhar et al. (2008) also proposed an activity graph representation. A single lattice structure was used to encode the evolution of spatiotemporal relationships between objects over an entire video sequence. The graph captured different spatial relationships between the features of objects: disconnects, surrounds, and touches. The graph also captured the times during which different spatial relationships between object features persisted in the video. The representation was used to compare objects from different videos in order to form a hierarchical categorization of the objects (Sridhar et al., 2008).

In our previous work (Griffith et al., 2009), we used visual co-movement to perform object categorization. Co-movement was represented using two features. The first feature captured whether the two objects moved at the same time. The second feature captured whether the two objects moved in the same direction. In these experiments, the robot interacted with a block and several objects by grasping the block, dropping the block above the object placed on a table, and then pushing the object. The co-movement features for each trial were clustered into outcome classes. The frequency with which different outcome classes occurred with each object was sufficient to separate the containers from the non-containers.

In a follow-up study (Griffith et al., 2011), we introduced a second approach for learning from the movement sequences of objects. Instead of extracting specific features for co-movement, a

single string was used to represent both the movement sequence for a block and the movement sequence for an object. Strings from different interaction trials were compared and clustered in order to identify the different co-movement patterns between the objects. The frequency with which different outcome classes occurred with each object was, again, sufficient to separate the containers from the non-containers.

This chapter introduces a new graph representation that captures the movement dependencies between objects during an activity. The representation is an improvement over our previous work (Griffith et al., 2009, 2011) because it can capture the specific times during an activity when different movement dependencies occur between objects. Because interactive learning about objects is one of the main applications for representations of movement (Griffith et al., 2009, 2011; Metta and Fitzpatrick, 2003; Ugur et al., 2007; Montesano et al., 2008), the representation described in this paper is evaluated using an interactive learning task to categorize containers from non-containers.

6.2 Experimental Setup

6.2.1 Robot

The experiments in this chapter were performed using the upper–torso humanoid robot shown in Fig. 6.1, which was described in more details in Chapter 3. Video was captured using the left camera mounted in the robot's head. Rubber finger tips were stretched over each of the three fingers on the robot's left hand in order to increase friction and improve the quality of grasps. The ZCam was not used in these experiments.

6.2.2 Objects

Ten objects were used in the experiments (see Fig. 6.2). They were selected to have a variety of shapes and material properties. Given the small behavioral repertoire of the robot, the objects were containers in one orientation and non-containers when flipped over. All objects had approximately the same height. Five blocks were also used in the experiments (see Fig. 6.2).

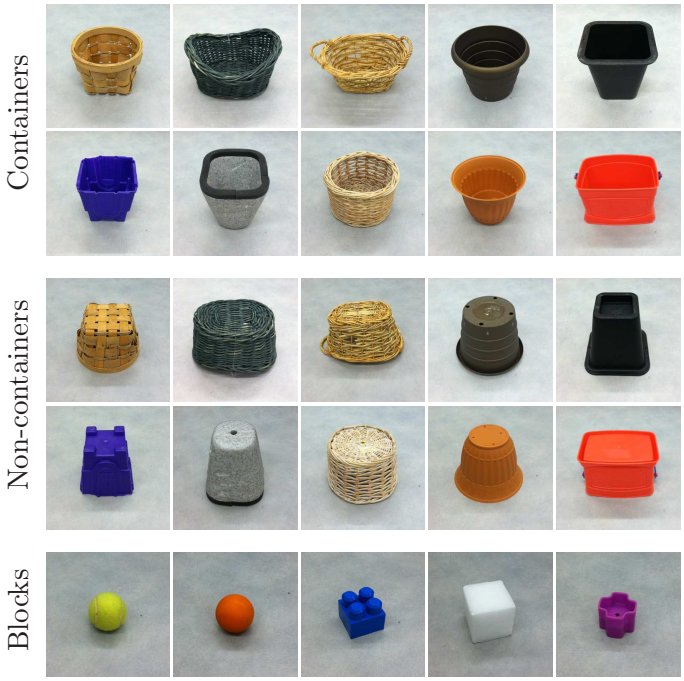


Figure 6.2 The objects and the blocks used in the experiments. (**Containers**) The first two rows show the 10 containers: wicker basket, plant basket, potpourri basket, flower pot, bed riser, purple bucket, styrofoam bucket, candy basket, brown bucket, and red bucket. (**Non-containers**) The second two rows show the same 10 objects as before, but flipped upside down, which makes them non-containers for this particular robot with this particular set of behaviors. (**Blocks**) The last row shows the 5 blocks: tennis ball, rubber ball, mega block, foam cube, and purple block.

The blocks also varied in shape and material properties. All blocks were small enough to fit inside each of the containers, but large enough to be graspable by the robot.

6.2.3 Robot Behaviors

For each object–block combination, the robot performed a short sequence of exploratory behaviors (see Fig. 6.1) in order to learn from their co-movement patterns, if any. The exploration sequence started with the robot grasping the object and waving it back and forth four times. The robot then dropped the block above the object. The drop point was sampled uniformly from a 10 cm x 10 cm area above the object. Next, the robot grasped the object and waved it back and forth four times. Finally, the robot dropped the object on the table.

Before the start of each trial the block and the object were manually placed at marked locations on the table. Double-sided tape was used to mark the location of the block and to

keep it from rolling out of place before the robot could grasp it. Even though the objects were placed in the robot's field of view at the start of each trial, they sometimes left the frame during the process of manipulation.

6.3 Methodology

6.3.1 Data Collection

During each trial the robot captured a sequence of 640x480 color images, recorded at 15 fps. Each trial lasted approximately 40 seconds. Thus, the robot collected roughly 600 images per trial (40 seconds $\times$ 15 frames per second). There were 100 trials in the dataset (20 objects $\times$ 5 blocks). The same data set was also used to obtain the results in Sukhoy et al. (2011) except Sukhoy et al. (2011) used another 100 trials in which a human performed the same manipulation activities as the robot.

6.3.2 Movement Detection

The movements of the features were detected using a combination of color tracking and optical flow. Color tracking was used to identify the centroid of each tracked feature. The optical flow was used to eliminate small vacillations of the feature position during periods of no movement. The optical flow was also used to limit the change in the tracking position of the blobs. More specifically, the displacement of the blob from frame to frame was capped at 1.5 times the magnitude of the largest vector in the segment of the optical flow field corresponding to the color blob. The dense optical flow vectors were computed off-line using the Matlab implementation of Sun *et al.*'s state-of-the-art algorithm (Sun et al., 2010).

The sequence of feature positions was used to extract a movement detection sequence. A feature was treated as moving when its position changed by more than 5 pixels between two consecutive frames. The movement detection sequence for each feature was further filtered using a box filter of width 5. The output for each trial was a movement detection sequence for the block, the object, and the robot's hand (see Fig. 6.3).

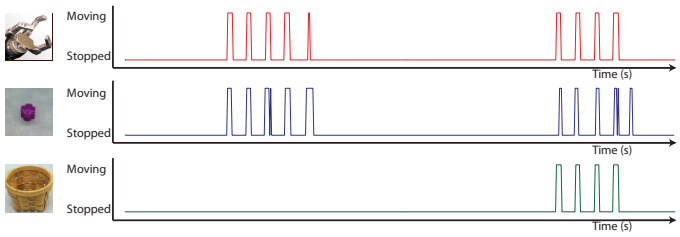


Figure 6.3 Detected movements for the robot's hand, the purple block, and the wicker basket as the robot interacted with them during one trial. The first set of movement spikes was observed when the robot was shaking the block. The second set of spikes was observed when the robot was shaking the container with the block inside it.

6.3.3 Extracting Movement Dependency Graphs

To represent co-movement of different features during trials, we constructed sequences of movement dependency graphs. The vertices in each of these graphs represent the three visual features that were tracked (the robot's hand, the block, and the object). The edges represent movement dependencies between pairs of features. In other words, an edge connects two vertices in the graph if and only if the movement of one feature determines the movement of another feature, at least partially. Dependencies can be observed for lack of movement as well. For example, if two features suddenly stop moving at the same time, then there will be a corresponding edge in the sequence of movement dependency graphs for a short time.

During trials, features can become attached and move together or they can become detached and move independently. For example, the robot can drop the block so that subsequent movements of the block become unrelated to the robot's hand movements. To represent these possibilities, we extracted a sequence of graphs using a sliding temporal window of size 3 seconds (i.e., 45 frames). For each window we constructed a contingency table for each of the three possible pairs of tracked features (i.e., hand–block, hand–object, and block–object). Each of these contingency tables contains four cells that correspond to the four possible combinations of two binary movement variables. Fig. 6.4 shows several example contingency tables for the block–object pair. The elements of a contingency table sum up to 45 because it contains data from 45 frames (i.e., 3 seconds $\times$ 15 frames per second).

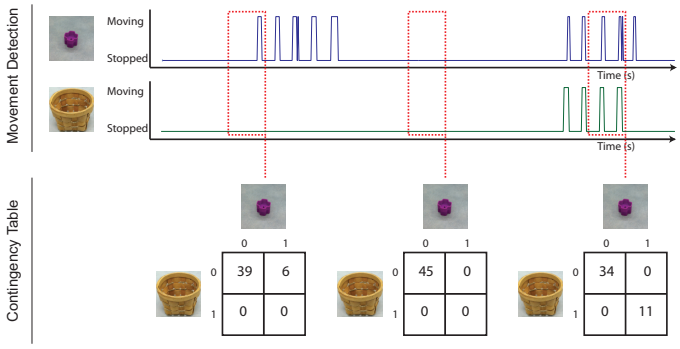


Figure 6.4 The process of extracting contingency tables from the detected movements of the block and the wicker basket. Each contingency table is computed from the movement detection data within a three-second-long sliding window. Three different contingency tables are shown. The first table was generated for a window of time when the robot was waving the block. The second one was generated when the robot was grasping the wicker basket. Finally, the third one was calculated when the robot was waving the basket with the block inside it.

The contingency table summarizes how often the features were moving together or not moving together (diagonal entries) or how often one feature was moving and the other one was not moving (off-diagonal entries). The contingency tables were recalculated incrementally as the temporal window advanced. The next section describes how a contingency table is used to decide if the corresponding edge for a pair of objects is present in the graph or not.

6.3.4 Statistical Test for Movement Dependencies

Given the contingency table for two features A and B, the goal is to decide if the movements of A and B are dependent or independent. This decision is made using the *G-test of independence*, which is a statistical test of independence (Sokal and Rohlf, 1994).

The G-test selects between the *null* hypothesis and the *alternative* hypothesis. In this case, the null hypothesis is that the movements of the two variables are independent. The alternative hypothesis is that they are dependent.

The G-test rejects the null hypothesis and accepts the alternative hypothesis if the p-value is below a chosen significance level. In this work, the significance level was set to 0.05, which is a standard significance level value in statistics. More formally, the rule for deciding if the

p-value indicates that the two variables are independent or not is shown below:

$$\text{Independent}(p) = \begin{cases} \text{Yes}, & \text{if } p \geq 0.05; \\ \text{No}, & \text{if } p < 0.05. \end{cases} \tag{6.1}$$

To compute the p-value, the G-test uses the G statistic, which is defined using the following formula:

$$G = 2 \sum_{i=0}^{1} \sum_{j=0}^{1} N_{ij} \ln \left(\frac{N_{ij}(N_{00} + N_{01} + N_{10} + N_{11})}{(N_{0j} + N_{1j})(N_{i0} + N_{i1})} \right),$$

where N_{00}, N_{01}, N_{10}, and N_{11} are the four elements of a contingency table (see also Fig. 6.4):

N_{00}	N_{01}
N_{10}	N_{11}

Note that the value of the G statistic is proportional to the mutual information between the movements of the two features.

If the null hypothesis is true, i.e., the variables are independent, then the G statistic is approximately distributed according to a χ^2 distribution with 1 degree of freedom. Thus, the p-value can be computed from the value of the G statistic using the following formula:

$$p = 1 - F_{\chi_1^2}(G),$$

where $F_{\chi_1^2}(G) = \Pr(\chi_1^2 < G)$ is the cumulative distribution function of the χ^2 distribution with 1 degree of freedom.

The confidence level, which is measured in percent, is computed from the p-value as shown below:

$$\text{Confidence Level} = (1 - p) \cdot 100\%.$$

The confidence level quantifies the confidence of the G-test in rejecting the null hypothesis. According to the decision rule (6.1), if the confidence level exceeds 95%, which corresponds to a p-value below 0.05, then the two variables are considered dependent and the corresponding edge is added to the graph. Otherwise, the two variables are considered independent and the corresponding edge is removed from the graph. In practice, the confidence level increases

exponentially as the window traverses the section of the movement data in which the movements of the two variables are dependent (see Fig. 6.5).

6.4 Results

6.4.1 Analyzing the Movement Dependency Graphs

A sequence of movement dependency graphs was calculated for each of the 100 trials as described in Section 6.3. To show how these sequences differ between containers and non-containers, the number of edges were calculated for each movement dependency graph and for each of the 100 sequences. This results in 100 sequences of integers between 0 and 3. These integer sequences were partitioned into two groups: 1) the group of sequences for all trials with containers and 2) the group of sequences for all trials with non-containers. For each of these two groups, the median and the median absolute deviation were computed over sequence elements with the same temporal index.

The two resulting sequences, which summarize all trials with containers and non-containers, are shown in Fig. 6.6. The graphs show two peaks, which represent the period of time when the robot was shaking the block (first peak) and the period of time when the robot was shaking the object (second peak). A block was inside a container during 30 out of 50 trials with the containers, i.e., three links were present in the movement dependency graphs when the robot was shaking the object only in 30 of the trials. The containers were empty during the other 20 trials, which meant that the movement dependency graphs had at most one link in these trials. Fig. 6.6 shows that the movement dependency graph representation clearly distinguishes between cases of containment and cases of non-containment in this experiment.

6.4.2 Object Categorization

Each sequence of movement dependency graphs (one sequence per trial) was converted into a string, for a total of 100 different strings. Because an edge was either present or not present in a movement dependency graph, each graph was mapped to a specific number, whose binary equivalent represented the existence of specific edges in the graph. A string represented the

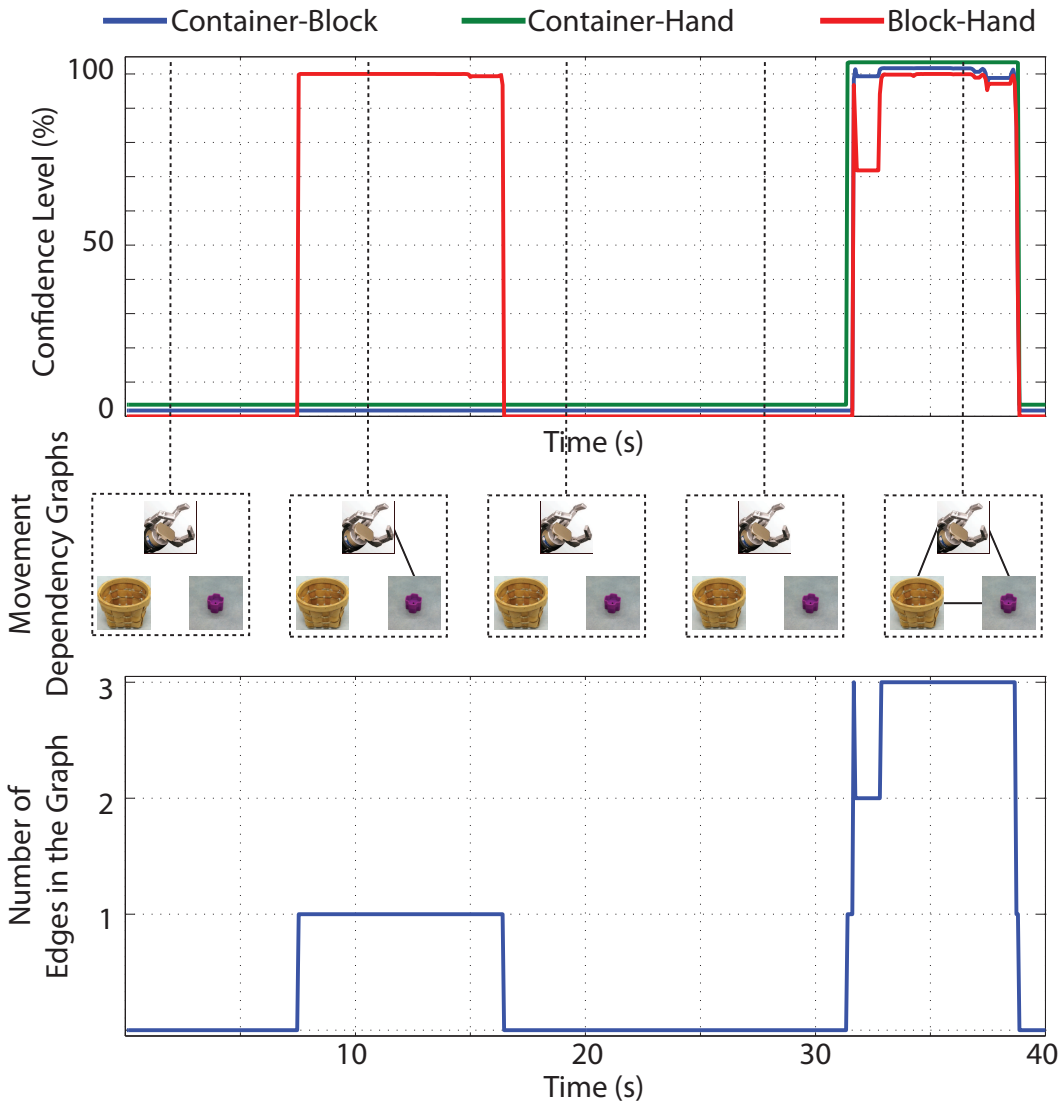


Figure 6.5 The process of extracting a sequence of temporally evolving movement dependency graphs for one trial performed by the robot. An edge between a pair of features in the movement dependency graph is created if the confidence level for that pair is greater than 0.95%. The result is a temporally evolving movement dependency graph, which shows what the robot controls at different points of time during the trial. The lines in the first plot were slightly offset in the y-direction in order to show all three lines, which were equal to zero for most of the trial. The last plot shows the number of edges in the movement dependency graphs over time.

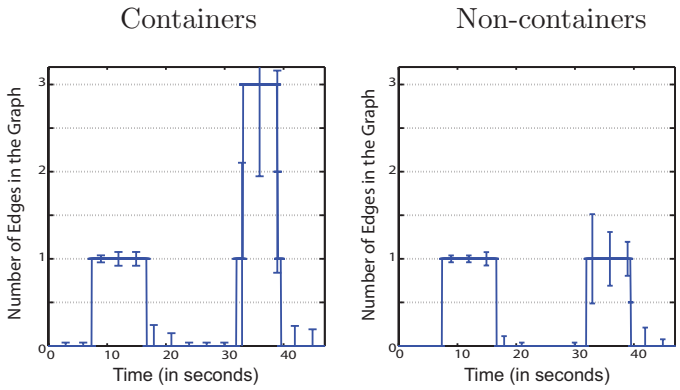


Figure 6.6 The median and the median absolute deviation of the number of movement dependencies, represented by the number of graph edges, for all trials with containers and non-containers. The number of edges in the movement dependency graphs is greater for containers when the robot is shaking them (second peak) because a block can be inside a container, which adds two more edges to the graph.

evolution of the existence of the different edges in the graph. The similarity between two strings was computed using the string edit distance between them. A 100×100 similarity matrix was constructed to represent the similarity between all pairs of strings. The similarity matrix was clustered using Spectral Clustering in order to obtain outcome classes. Finally, the objects were categorized based on the frequency with which different outcomes occurred with each object. See Chapter 5 for more details.

The object categorization results are shown in Fig. 6.7. The brown bucket was the only misclassified object. None of the blocks fell into that container during the trials with it, which is why it was not distinguished from the non-containers. Only a single instance of containment was necessary to distinguish the other containers from the non-containers. Thus, the robot could have easily performed a few more trials per object in order to accurately categorize all objects. In summary, the results suggest that the movement dependency graph representation can be used to distinguish between different object categories.

6.5 Discussion

The representation in this chapter is a significant improvement over previous work. The robot performed only 5 trials per object in this set of experiments, which was much less than the number of trials performed in our previous work (i.e., 100 trials per object in Chapter 4

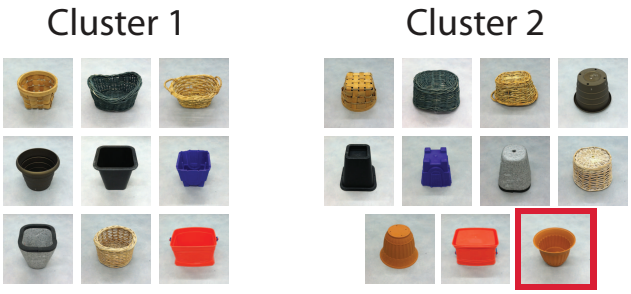


Figure 6.7 The object categorization formed by the robot. The brown bucket was the only misclassified object in this set of experiments.

and in Chapter 5). Previously, about 40 interactions per object were required to accurately identify its type (see Chapter 5) and more than a single instance of containment was necessary to distinguish the containers from the non-containers. In this chapter, however, an object was identified as a container even if a containment event was observed during only one trial with that object.

The representation described in this chapter would become analogous to the representation described in Chapter 4 if the window size was increased to be equal to the number of video frames in one trial instead of being fixed at 45 frames. In this case, each sequence of movement dependency graphs would contain only one graph. That representation of movement dependencies would work well if each trial consisted of only one behavior (e.g., shake). If a trial spans more than one behavior, which is typically the case for manipulation activities, then reducing the co-movement of objects to a single value completely misses the temporal structure of the activity.

Similarly, the representation described in this chapter would be analogous to the representation described in Chapter 5 if the raw visual tracking data was used instead of the sequences of movement dependency graphs. This would replace the movement dependencies between pairs of objects with traces that individual objects produce in visual space. This representation would also work well if a trial consisted of only one behavior (see Chapter 5). However, this approach would not extract the exact moments when objects start or stop moving together. This information is present in the tracking data only implicitly. Thus, the ability to detect

events that correspond to the beginning or the end of co-movement or separate movement, which is crucial for understanding the structure of manipulation tasks, would be missing.

Because the temporal window that was used to construct the graphs spans three seconds, the movement dependencies that persist for only a few frames are ignored. Only statistically significant movement dependencies are extracted. This is important when using the representation in this chapter for comparing human and robot manipulations. The robot may move at a more consistent and slower speed than a human does. Still, a preliminary study showed that the representation in this chapter can be used to compare manipulation tasks performed by the robot with manipulation tasks performed by a human (Sukhoy et al., 2011). In that case, however, a different method was used in order to compare two sequences (i.e., dynamic time warping) (Sukhoy et al., 2011).

6.6 Summary

This chapter introduced a new representation that captures the movement dependencies between objects, which can be used to form object categories. The proposed representation of movement dependencies is a temporal sequence of graphs. The vertices in these graphs correspond to visual features tracked by the robot. The edges are constructed using movement dependencies between these features. An edge between two vertices is present in a graph if and only if a statistical test indicates that the movements of the two corresponding features are dependent. Graphs are combined into sequences of graphs, which reflect how the movement dependencies between features change over time as the robot performs actions with different objects.

The representation was tested on a categorization task with container and non-container objects. The graph sequences were constructed from 100 trials during which the robot attempted to insert a small block into the objects and shake them. The results showed that the sequences of movement dependency graphs captured the differences between containers and non-containers.

Several interesting extensions of this representation are left for future work. Currently,

the movement dependency graphs have unweighted edges, which assumes equal movement dependence relationships between all pairs of objects. This is not completely accurate, though. For example, when the robot is controlling a container that has a loose–fitting block inside it, the robot has less control over the block's movements. Thus, the movements of the block are not as synchronized with the robot's hand as are the movements of the container. It may be possible to extend the proposed representation to account for different types of dependences by assigning weights to the edges of the movement dependency graphs.

Another interesting extension that is left for future work is to augment the representation with spatial information. Currently, the representation captures only movement dependencies between the objects. For different activities, however, a movement dependency between two objects could occur in different spatial relationships. For example, a block can co-move with a container when it is inside the container while the container is shaken or when it is next to the container while the container is pushed. An improved representation that also captures spatial information between objects could distinguish between these two different types of movement dependencies.

CHAPTER 7. SUMMARY, CONCLUSION, AND FUTURE WORK

This chapter summarizes the contributions of this thesis and describes the conclusions that can be made from them. This chapter also suggests several directions for future work.

7.1 Summary

This thesis investigated how a robot could learn a behavior–grounded object category for containers. A computational framework for interactive learning of object categories was first proposed in Chapter 4 and improved in the following two chapters. Chapter 4 demonstrated that it was possible for a robot to interactively distinguish between containers and non-containers using visual co-movement patterns. The robot dropped a block above an object and then pushed the object as it observed the movement patterns between them. The robot used the frequency with which different co-movement outcomes occurred with each object to separate them into categories. The resulting object category labels were used to learn a visual model for the objects, which the robot used to classify novel objects as containers or non-containers.

Chapter 5 expanded upon the object categorization framework by adding more behaviors and more sensory modalities. It showed how a robot with a large behavioral and perceptual repertoire could learn a single, meaningful categorization for a set of objects. The modified framework consisted of six steps. First, the robot performed 6 different behaviors on 20 different objects (10 containers and 10 non-containers) as it captured data from audio and video input streams. Second, features were extracted from the audio and video data in an unsupervised way. In the third step, the robot clustered the features into perceptual outcome classes. Fourth, the robot clustered the objects into categories using the frequency with which different out-

comes occurred with each object. The resulting object categories varied in quality, some were better than others. Fifth, the robot unified the object categories for each behavior–modality combination into a single, meaningful categorization of the objects. Only one of the twenty objects was misclassified in the unified object categorization. In the sixth step, the category labels from the unified categorization were used to learn a visual model, which allowed the robot to identify the correct category for 29 out of 30 novel objects from only passive observation. Finally, in an additional evaluation, it was shown that about 60 interactions per object were necessary to form a meaningful categorization using this framework.

The object categorization framework was again expanded in Chapter 6, which introduced movement dependency graphs as a more powerful way to represent the movement dependencies between objects. The new representation allowed the robot to describe the evolution of an activity based on the changing movement dependencies between objects. The new representation was, again, evaluated using an interactive object categorization task with 10 containers and 10 non-containers. Only a single object was misclassified after the robot performed 5 interactions per object.

7.2 Conclusion

Although containers are one of the simplest kinds of objects, learning to separate containers from non-containers is not easy. The objects have to be manipulated in order to produce sounds and visual movement patterns that the robot can learn from. Furthermore, several interactions with each object have to be performed in order to accurately estimate their functional properties. In this thesis, the robot was able to learn a category for containers after it performed 5 interactions with each object (see Chapter 6). An additional number of interactions may be required to form object categories for more complex types of objects, or for discriminating between categories of objects that are very similar to one another.

The categorization accuracy for a single behavior–modality combination is highly dependent on the behavior that the robot used to produce the categorization. Some behaviors simply capture the ‘container’ property better than others. The behaviors that best discriminated be-

tween containers and non-containers using vision (e.g., the move behavior) tested the movement dependencies between the objects. The behaviors that best discriminated between containers and non-containers using sound caused the block to become contained (which occurred during the *drop block* behavior) and to become uncontained (which occurred during the *flip* behavior). This suggests that the interactive behaviors that can best discriminate between object categories are behaviors that capture some category–specific property. Indeed, the robot performed well when category–specific interactions were used.

In the end, this thesis showed that the robot can split the objects into meaningful categories even though it does not know the mapping between these categories and the human words for them. What the robot does know, however, is that the objects in a given category produce similar distributions of outcomes. The robot also knows that the differences between categories can be explained in terms of the frequencies of the detected events. The robot was able to learn what movement dependency outcomes it could expect to observe with containers. Finally, the robot also knows a visual model for containers, which allowed it to identify the object category of novel objects that it did not interact with. These achievements constitute a small step toward creating a developmental progression of container learning for robots.

7.3 Future Work

This thesis only introduced the general problem of learning how to detect and use containers in a flexible way. Several research directions are left for future work. For example, one research direction of primary importance is how to close the loop in the object categorization framework (see Fig. 5.2). Presumably, the robot could use its visual model to guide its future interactions with containers, which would help it to refine its definition of what it means for an object to be a 'container.' Future work could also improve the categorization framework by eliminating the need for human–provided object IDs.

Future work could also investigate how to integrate tactile and proprioceptive feedback into the learning framework. These sensory modalities are a primary source of information for both humans and animals when they try to distinguish between containers and other types of

objects. For example, adult humans try inserting their hand into an object to test if it is a container (Lederman and Klatzky, 1987). Hermit crabs use tactile and proprioceptive feedback to identify shells with holes in them (Bertness, 1980). Similarly, a robot may also be able to use tactile and proprioceptive feedback to separate containers from non-containers.

Future work could also extend the movement dependency graph representation to include the mutual information of the movement dependencies between two different objects. It may be possible for a robot to use mutual information between its own movements and the movements of objects in order to learn about the controllability of containers and their contents. For example, the mutual information between the robot's movements and movements of the container will be higher than the mutual information between the robot's movements and the movements of the objects inside the container. These levels of controllability can be used to augment the movement dependency graph, e.g., by adding weights to its edges.

Finally, future work could investigate how a robot could use the movement dependencies between objects to simplify learning from imitation. Traditionally, imitation learning in robotics tries to create precise kinematic models of the behaviors that a human performs on an object, which are then mapped to the robot's body in order to exactly replicate the human's actions. It may be possible, however, for a robot to use its own behavioral repertoire to imitate the movement dependencies created by a human during an activity, instead of imitating the human's behaviors themselves. In a preliminary study (Sukhoy et al., 2011) we showed that the movement dependencies created by a robot during an activity usually match the movement dependencies created by a human during the same activity.

BIBLIOGRAPHY

Aguiar, A. and Baillargeon, R. (1998). Eight-and-a-half-month-old infants' reasoning about containment events. *Child Development*, 69(3):636–653.

Aksoy, E., Abramov, A., Worgotter, F., and Dellen, B. (2010). Categorizing object-action relations from semantic scene graphs. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 398–405, Anchorage, AK.

Baillargeon, R. (1994). How do infants learn about the physical world? *Current Directions in Psychological Science*, 3(5):133–140.

Baillargeon, R. (2004). Infants' physical world. *Current Directions in Psychological Science*, 13(3):89–94.

Baillargeon, R. and DeVos, J. (1991). Object permanence in young infants: Further evidence. *Child Development*, 62:1227–1246.

Barsalou, L. W., Simmons, W. K., Barbey, A. K., and Wilson, C. D. (2003). Grounding conceptual knowledge in modality-specific systems. *Trends in Cognitive Sciences*, 7(2):84–91.

Bentivegna, D., Atkeson, C., and Cheng, G. (2004). Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47(2):163–169.

Bertness, M. (1980). Shell preference and utilization patterns in littoral hermit crabs of the bay of Panama. *Journal of Experimental Marine Biology and Ecology*, 48(1):1–16.

Bonniec, P.-L. (1985). From visual-motor anticipation to conceptualization: Reaction to solid and hallow objects and knowledge of the function of containment. *Infant Behavior and Development*, 8(4):413–424.

Bowerman, M. and Choi, S. (2003). Language in mind: Advances in the study of language and cognition. chapter Space under construction: Language specific spatial categorization in first language acquisition, pages 387–428. Cambridge: MIT Press.

Brooks, R. A. (1991). Intelligence without representation. In *Artificial Intelligence*, volume 47, pages 139–159.

Butterworth, G. and Castillo, M. (1976). Coordination of auditory and visual space in newborn human infants. *Perception*, 5(2):155–160.

Caron, A., Caron, R., and Antell, S. (1988). Infants understanding of containment: An affordance perceived or a relationship conceived. *Developmental Psychology*, 24(5):620–627.

Casasola, M., Cohen, L. B., and Chiarello, E. (2003). Six-month-old infants' categorization of containment spatial relations. *Child Development*, 74(3):679–693.

Chang, L., Srinivasa, S., and Pollard, N. (2010). Planning pre-grasp manipulation for transport tasks. In *IEEE Intl. Conf. on Robotics and Automation*, pages 2697–2704.

Cohen, L. (2003). Unresolved issues in infant categorization. In Rakison, D. and Oakes, L. M., editors, *Early category and concept development*, pages 193–209. New York: Oxford University Press.

Crain, W. (1999). Theories of development: Concepts and applications. chapter Piaget's Cognitive-Developmental Theory, pages 110–146. Prentice Hall, Upper Saddle Ricer, NJ.

Damasio, A. R. (1989). Time-locked multiregional retroactivation: a systems-level proposal for the neural substrates of recall and recognition. *Cognition*, 33(1):25–62.

DiCarlo, J. J. and Cox, D. D. (2007). Untangling invariant object recognition. *Trends in Cognitive Sciences*, 11(8):333–341.

Edsinger, A. and Kemp, C. C. (2007). Two arms are better than one: A behavior-based control system for assistive bimanual manipulation. In *Proceedings of the Thirteenth International Conference on Advanced Robotics*.

Feddema, J., Dohrmann, C., Parker, G., Robinett, R., Romero, V., and Schmitt, D. (1997). Control for slosh-free motion of an open container. *IEEE Control Systems Magazine*, 17(1):29–36.

Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini, G. (2003). Learning about objects through action - initial steps towards artificial cognition. In *in Proc. of the 2003 IEEE Intl. Conf. on Robotics and Automation*, pages 3140–3145.

Fitzpatrick, P., Needham, A., Natale, L., and Metta, G. (2008). Shared challenges in object perception for robots and infants. *Journal of Infant and Child Development*, 17(1):7–24.

Freeman, N. H., Lloyd, S., and Sinha, C. G. (1980). Infant search tasks reveal early concepts of containment and canonical usage of objects. *Cognition*, 8(3):243–262.

Gibson, E. J. (1988). Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual review of psychology*, 39(1):1–42.

Gold, K. and Scassellati, B. (2009). Using probabilistic reasoning over time to self-recognize. *Robotics and Autonomous Systems*, 57(4):384–392.

Griffith, S., Sinapov, J., Miller, M., and Stoytchev, A. (2009). 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.

Griffith, S., Sinapov, J., Sukhoy, V., and Stoytchev, A. (2010). 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)*, pages 1852–1859, Anchorage, AK.

Griffith, S., Sinapov, J., Sukhoy, V., and Stoytchev, A. (2011). A behavior–grounded approach to forming object categories: Separating containers from non-containers. *To appear in the IEEE Transactions on Autonomous Mental Development*.

Griffith, S. and Stoytchev, A. (2010). Interactive categorization of containers and non-containers by unifying categorizations derived from multiple exploratory behaviors. In *Proc. of the 24th Nat. Conf. on Artificial Intelligence (AAAI)*, pages 1931–1932, Atlanta, GA.

Hasher, L. and Zacks, R. T. (1984). Automatic processing of fundamental information: The case of frequency of occurrence. *American Psychologist*, 39:1372–1388.

Hespos, S. and Baillargeon, R. (2001a). Reasoning about containment events in very young infants. *Cognition*, 78(3):207–245.

Hespos, S. and Baillargeon, R. (2001b). Reasoning about containment events in very young infants. *Cognition*, 78(3):207–245.

Hespos, S. and Baillargeon, R. (2006). Decalage in infants' knowledge about occlusion and containment events: Converging evidence from action tasks. *Cognition*, 99(2):B31–B41.

Hespos, S. and Spelke, E. (2004). Conceptual precursors to language. *Nature*, 430(22):453–456.

Hespos, S. and Spelke, E. (2007). Precursors to spatial language: The case of containment. In Aurnague, M., Hickmann, M., and Vieu, L., editors, *The Categorization of Spatial Entities in Language and Cognition*, pages 233–245. Benjamins Publishers, Amserdam, Netherlands.

Horst, J. S., Oakes, L. M., and Madole, K. L. (2005). What does it look like and what can it do? Category structure influences how infants categorize. *Child Development*, 76(3):614–631.

Jusczyk, P., Cutler, A., and Redanz, N. (1993). Preference for the predominant stress patterns of English words. *Child Development*, 64(3):675–687.

Kemp, C. and Edsinger, A. (2006a). What can I control? The development of visual categories for a robot's body and the world that it influences. In *Proc. of the 5th Intl. Conf. on Development and Learning (ICDL), Special Session on Autonomous Mental Development*.

Kemp, C., Edsinger, A., and Torres-Jara, E. (2007). Challenges for robot manipulation in human environments. *IEEE Robotics and Automation Magazine*, 14(1):20–29.

Kemp, C. C. and Edsinger, A. (2006b). Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *Proceedings of the Fifth Annual Conference on Development and Learning, Special Session on Classifying Activities in Manual Tasks*.

Krotkov, E., Klatzky, R. L., and Zumel, N. B. (1997). Robotic perception of material: Experiments with shape-invariant acoustic measures of material type. In *The 4th International Symposium on Experimental Robotics IV*, pages 204–211, London, UK. Springer-Verlag.

Largo, R. and Howard, J. (1979a). Developmental progression in play behavior of children between nine and thirty months: I: Spontaneous play and imitation. *Developmental Medicine and Child Neurology*, 21(3):299–310.

Largo, R. and Howard, J. (1979b). Developmental progression in play behavior of children between nine and thirty months: II: Spontaneous play and language development. *Developmental Medicine and Child Neurology*, 21(4):492–503.

Lederman, S. and Klatzky, R. (1987). Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368.

Lee, H., Battle, A., Raina, R., and Ng., A. Y. (2007). Efficient sparse coding algorithms. In *Proc. of NIPS*, pages 801–888.

Leslie, A. M. and DasGupta, P. (1991). Infants' understanding of a hidden mechanism: Invisible displacement. Paper presented at symposium on "Infants' reasoning about spatial relationships." SRCD Biennial Conference, Seattle.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Lorenz, K. (1996). Learning as self-organization. chapter Innate bases of learning, pages 1–54. Mahwah, NJ: Lawrence Erlbaum and Associates, Publishers.

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15(4):151–190.

Luo, Y. and Baillargeon, R. (2005). When the ordinary seems unexpected: evidence for incremental physical knowledge in young infants. *Cognition*, 95(3):297–338.

MacLean, D. and Schuler, M. (1989). Conceptual development in infancy: The understanding of containment. *Child Development*, 60(5):1126–1137.

Mandler, J. (2004). Thought before language. *Trends in Cognitive Sciences*, 8(11):508–513.

Metta, G. and Fitzpatrick, P. (2003). Early integration of vision and manipulation. *Adaptive Behavior*, 11(2):109–128.

Montesano, L. and Lopes, M. (2009). Learning grasping affordances from local visual descriptors. In *IEEE 8th Intl. Conf. on Development and Learning*, pages 1–6.

Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: From sensory-motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26.

Nakamura, T., Nagai, T., and Iwahashi, N. (2007). Multimodal object categorization by a robot. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2415–2420.

Navarro, G. (2001). A guided tour to approximate string matching. *ACM Computing Surveys*, 33(1):31–88.

Needham, A., Cantlon, J., and Holley, S. O. (2006). Infants' use of category knowledge and object attributes when segregating objects at 8.5 months of age. *Cog. Psychology*, 53(4):345–360.

Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453.

Nguyen, H. and Kemp, C. (2008). Bio-inspired assistive robotics: Service dogs as a model for human-robot interaction and mobile manipulation. In *Proc. of the 2nd Biennial IEEE/RAS-EMBS Intl. Conf. on Biomedical Robotics and Biomechatronics*, pages 542–549.

Okada, K., Kojima, M., Tokutsu, S., Mori, Y., Maki, T., and Inaba, M. (2009). Integrating recognition and action through task-relevant knowledge for daily assistive humanoids. *Advanced Robotics*, 23(4):459–480.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code of natural images. *Nature*, 381:607–609.

Pelleg, D. and Moore, A. W. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proc. of the Seventeenth International Conference on Machine Learning*, pages 727–734.

Pfeifer, R. and Scheier, C. (1997). Sensory-motor coordination: The metaphor and beyond. In *Robotics and Autonomous Systems*, volume 20, pages 157–178.

Pinz, A. (2005). Object categorization. *Foundations and Trends in Computer Graphics and Vision*, 1(4):255–353.

Power, T. (2000). *Play and Exploration in Children and Animals*. Mahwah, NJ: Laurence Erlbaum Associates.

Rakison, D. H. and Oakes, L. M., editors (2003). *Early Category and Concept Development: Making Sense of the Blooming, Buzzing Confusion*. Oxford University Press, USA, 1 edition.

Richmond, K. and Pai, D. (2000). Active measurement of contact sound. In *in Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 2146–2152.

Robinson, C. W. and Sloutsky, V. M. (2004). Auditory dominance and its change in the course of development. *Child Development*, 75(5):1387–1401.

Rochat, P. (1989). Object manipulation and exploration in 2- to 5-month-old infants. *Developmental Psychology*, 25(6):871–884.

Rochat, P. and Striano, T. (1998). Primacy of action in early ontogeny. *Human Development*, 41:112–115.

Romero, V. and Ingber, M. (1995). A numerical model for 2-D sloshing of pseudo-viscous liquids in horizontally accelerated rectangular containers. In *Proc. of the 17th International Conference on Boundary Elements*.

Rooks, B. (2006). The harmonious robot. *Industrial Robot*, 36(2):125–130.

Rosch, E. (1978). Principles of categorization. In Rosch, E. and Lloyd, B. B., editors, *Cognition and Categorization*, pages 189–206. Hillsdale, NJ: Erlbaum.

Rusu, R., Holzbach, A., Diankov, R., Bradski, G., and Beetz, M. (2009). Perception for mobile manipulation and grasping using active stereo. In *the 9th IEEE-RAS Intl. Conf. on Humanoid Robots*, pages 632–638.

Sahai, R., Griffith, S., and Stoytchev, A. (2009). Interactive identification of writing instruments and writable surfaces by a robot. In *Proc. of the RSS 2009 Workshop - Mobile Manipulation in Human Environments*, Seattle, WA.

Sally, D. (2005). Can I say "bobobo" and mean "There's no such thing as cheap talk"? *Journal of Economic Behavior and Organization*, 57(3):245–266.

Saxena, A., Drimeyer, J., and Ng, A. (2008). Robotic grasping of novel objects using vision. *International Journal of Robotics Research*, 27(2):157–173.

Schmidt, R. (1995). Consciousness and foreign language learning: A tutorial on the role of attention and awareness in learning. In Schmidt, R., editor, *Attention and awareness in foreign language learning (Technical Report 9)*, pages 1–63. Honolulu: University of Hawai'i, Second Language Teaching and Curriculum Center.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Simmons, W. and Barsalou, L. (2003). The similarity-in-topography principle: reconciling theories of conceptual deficits. *Cognitive Neuropsychology*, 20(3):451–486.

Sinapov, J. and Stoytchev, A. (2008). Detecting the functional similarities between tools using a hierarchical representation of outcomes. In *Proceedings of the 7th IEEE International Conference on Development and Learning*, Monterey, CA.

Sinapov, J. and Stoytchev, A. (2009). From acoustic object recognition to object categorization by a humanoid robot. In *Proc. of the RSS 2009 Workshop - Mobile Manipulation in Human Environments*, Seattle, WA.

Sinapov, J. and Stoytchev, A. (2010). The boosting effect of exploratory behaviors. In *Proc. of the 24th National Conference on Artificial Intelligence (AAAI)*, pages 1613–1618, Atlanta, GA.

Sinapov, J. and Stoytchev, A. (2011). Object category recognition by a humanoid robot using behavior–grounded relational learning. In *in Proc. of the 2011 IEEE Intl. Conf. on Robotics and Automation*.

Sinapov, J., Sukhoy, V., Sahai, R., and Stoytchev, A. (2011). Vibrotactile recognition and categorization of surfaces by a humanoid robot. *To appear in the IEEE Transactions on Robotics*.

Sinapov, J., Wiemer, M., and Stoytchev, A. (2009). Interactive learning of the acoustic properties of household objects. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3937–3943, Kobe, Japan.

Sitskorn, S. M. and Smitsman, A. W. (1995). Infants' perception of dynamic relations between objects: Passing through or support? *Developmental Psychology*, 31:437–447.

Smith, L. B. (2005). Shape: A developmental product. In Carlson, L. and VanderZee, E., editors, *Functional Features in Language and Space*, pages 235–255. Oxford University Press.

Sokal, R. and Rohlf, F. (1994). *Biometry: the principles and practice of statistics in biological research*. Freeman, New York, 3 edition.

Spelke, E. (1994). Initial knowledge: six suggestions. *Cognition*, 50(1):431–445.

Spelke, E. S. and Kinzler, K. D. (2007). Core knowledge. *Developmental Science*, 10(1):89–96.

Sridhar, M., Cohn, A., and Hogg, D. (2008). Learning functional object categories from a relational spatio-temporal representation. In *Proc. of the 2008 conf. on ECAI*.

Stoytchev, A. (2005). Behavior-grounded representation of tool affordances. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3071–3076.

Stoytchev, A. (2006). Five basic principles of developmental robotics. In *NIPS 2006 Workshop on Grounding Perception, Knowledge and Cognition in Sensor-Motor Experience*.

Stoytchev, A. (2009). Some basic principles of developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 1(2):122–130.

Stoytchev, A. (2011). Self-detection in robots: A method based on detecting temporal contingencies. *Robotica*, 29:1–21.

Strehl, A. and Ghosh, J. (2002). Cluster ensembes - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

Sukhoy, V., Griffith, S., and Stoytchev, A. (2011). Toward imitating object manipulation tasks using sequences of movement dependency graphs. In *Proc. of the RSS 2011 Workshop on the State of Imitation Learning*, Los Angeles, CA.

Sun, D., Roth, S., and Black, M. (2010). Secrets of optical flow estimation and their principles. In *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2439, San Francisco, CA.

Sutton, M., Stark, L., and Bowyer, K. (1994). Gruff-3: generalizing the domain of a functional-based recognition system. *Pattern Recognition*, 27(12):1743–1766.

Sutton, R. (2001). Verification, the key to AI. on-line essay. `http://www.cs.ualberta.ca/~sutton/IncIdeas/KeytoAI.html`.

Torres-Jara, E., Natale, L., and Fitzpatrick, P. (2005). Tapping into touch. In *Proc. of the Fifth Intl Workshop on Epigenetic Robotics*, pages 79–86.

Tzamtzi, M., Koumboulis, F., and Kouvakas, N. (2007). A two stage robot control for liquid transfer. In *IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 1324–1333.

Ugur, E., Dogar, M., Cakmak, M., and Sahin, E. (2007). The learning and use of traversability affordance using range images on a mobile robot. In *Proc. of the IEEE Intl Conf. on Robotics and Automation*, pages 1721–1726.

Vance, R. (1972). The role of shell adequacy in behavioral interactions involving hermit crabs. *Ecology*, 53(6):1075–1083.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wang, S. and Baillargeon, R. (2008). Can infants be "taught" to attend to a new physical variable in an event category? The case of height in covering events. *Cognitive Psychology*, 56:284–326.

Wang, S., Baillargeon, R., and Peterson, S. (2005). Detecting continuity violations in infancy: a new account and new evidence from covering and tube events. *Cognition*, 95(2):129–173.

ZCam (2008). 3DV Systems. http://www.3dvsystems.com/technology/product.html.