Learning the Affordances of Tools Using a Behavior-Grounded Approach

Alexander Stoytchev

Department of Electrical and Computer Engineering
Iowa State University, Ames IA 50011, USA
alexs@iastate.edu

Abstract. This paper introduces a behavior-grounded approach to representing and learning the affordances of tools by a robot. The affordance representation is learned during a behavioral babbling stage in which the robot randomly chooses different exploratory behaviors, applies them to the tool, and observes their effects on environmental objects. As a result of this exploratory procedure, the tool representation is *grounded* in the behavioral and perceptual repertoire of the robot. Furthermore, the representation is *autonomously* testable and verifiable by the robot as it is expressed in concrete terms (i.e., behaviors) that are directly available to the robot's controller. The tool representation described here can also be used to solve tool-using tasks by dynamically sequencing the exploratory behaviors which were used to explore the tool based on their expected outcomes. The quality of the learned representation was tested on extension-of-reach tasks with rigid tools.

1 Introduction

The ability to use tools is one of the hallmarks of intelligence. Tool use is fundamental to human life and has been for at least the last two million years. We use tools to extend our reach, to amplify our physical strength, to transfer objects and liquids, and to achieve many other everyday tasks. A large number of animals have also been observed to use tools [1]. Some birds, for example, use twigs or cactus pines to probe for larvae in crevices which they cannot reach with their beaks. Sea otters use stones to open hard-shelled mussels. Chimpanzees use stones to crack nuts open and sticks to reach food, dig holes, or attack predators. Orangutans fish for termites with twigs and grass blades. Horses and elephants use sticks to scratch their bodies. These examples suggest that the ability to use tools is an adaptation mechanism used by many organisms to overcome the limitations imposed on them by their anatomy.

Despite the widespread use of tools in the animal world, however, studies of *autonomous* robotic tool use are still rare. There are industrial robots that use tools for tasks such as welding, cutting, and painting, but these operations are carefully scripted by a human programmer. Robot hardware capabilities, however, continue to increase at a remarkable rate. Humanoid robots such as Honda's Asimo, Sony's Qrio, and NASA's Robonaut feature motor capabilities similar to those of humans. In the near future similar robots will be working side by side with humans in homes, offices, hospitals, and in outer space. It is difficult to imagine how these robots that will look like us, act like us, and live in the same physical environment like us, will be very useful if they are not capable of something so innate to human culture as the ability to use tools. Because of their humanoid "anatomy" these robots undoubtedly will have to use external objects in a variety of tasks, for instance, to improve their reach or to increase their physical strength. These important problems, however, have not been well addressed by the robotics community.

Another motivation for studying robot tool behaviors is the hope that robotics can play a major role in answering some of the fundamental questions about tool-using abilities of animals and humans. After ninety years of tool-using experiments with animals (see next section) there is still no comprehensive theory attempting to explain the origins, development, and learning of tool behaviors in living organisms.

Progress along these two lines of research, however, is unlikely without initial experimental work which can be used as the foundation for a computational theory of tool use. Therefore, the purpose of this paper is to empirically evaluate one specific way of representing and learning the functional properties or affordances [2] of tools.

The tool representation described here uses a behavior-based approach [3] to *ground* the tool affordances in the existing behavioral repertoire of the robot. The representation is learned during a behavioral babbling stage in which the robot randomly chooses different exploratory behaviors, applies them to the tool, and observes their effects on environmental objects. The quality of the learned representation is tested on extension-of-reach tool tasks. The experiments were conducted using a mobile robot manipulator. As far as we know, this is one of the first studies of this kind in the Robotics and AI literature.

2 Related Work

2.1 Affordances and Exploratory Behaviors

A simple object like a stick can be used in numerous tasks that are quite different from one another. For example, a stick can be used to strike, poke, prop, scratch, pry, dig, etc. It is still a mystery how animals and humans learn these affordances [2] and what are the cognitive structures used to represent them.

James Gibson defined affordances as "perceptual invariants" that are directly perceived by an organism and enable it to perform tasks [2]. Gibson is not specific about the way in which affordances are learned but he suggests that some affordances are learned in infancy when the child experiments with objects. For example, an object affords throwing if it can be grasped and moved away from one's body with a swift action of the hand and then letting it go. The perceptual invariant in this case is the shrinking of the visual angle of the object as it is flying through the air. This highly interesting "zoom" effect will draw the attention of the child [2, p. 235].

Gibson defines tools as detached objects that are graspable, portable, manipulable, and usually rigid [2, p. 40]. A hammer, for example, is an elongated object that is graspable at one end, weighted at the other end, and affords hitting or hammering. A knife, on the other hand, is a graspable object with a sharp blade that affords cutting. A writing tool like a pencil leaves traces when applied to surfaces and thus affords trace-making [2, p. 134].

The related work on animal object exploration indicates that animals use stereotyped exploratory behaviors when faced with a new object [4, 5]. This set of behaviors is species specific and may be genetically predetermined. For some species of animals these tests include almost their entire behavioral repertoire: "A young corvide bird, confronted with an object it has never seen, runs through practically all of its behavioral patterns, except social and sexual ones." [5, p. 44].

Recent studies with human subjects also suggest that the internal representation for a new tool used by the brain might be encoded in terms of specific past experiences [6]. Furthermore, these past experiences consist of brief feedforward movement segments used in the initial exploration of the tool [6]. A tool task is later solved by dynamically combining these sequences [6].

Thus, the properties of a tool that an animal is likely to learn are directly related to the behavioral and perceptual repertoire of the animal. Furthermore, the learning of these properties should be relatively easy since the only requirement is to perform a (small) set of exploratory behaviors and observe their effects. Based on the results of these "experiments" the animal builds an internal representation for the tool and the actions that it affords. Solving tool tasks in the future is based on dynamically combining the exploratory behaviors based on their expected results.

Section 3 formulates a behavior-grounded computational model of tool affordances based on these principles.

2.2 Experiments with Primates

According to Beck [1], whose taxonomy is widely adopted today, most animals use tools for four different functions: 1) to extend their reach; 2) to amplify the mechanical force that they can exert on the environment; 3) to enhance the effectiveness of antagonistic display behaviors; and 4) to control the flow of liquids. This paper focuses only on the extension of reach mode of tool use.

Extension of reach experiments have been used for the last 90 years to test the intelligence and tool-using abilities of primates [7–9]. In these experiments the animal is prevented from getting close to an incentive and thus it must use one of the available tools to bring the incentive within its sphere of reach.

Wolfgang Köhler was the first to systematically study the tool behaviors of chimpanzees. He performed a large number of experiments from 1913 to 1917. The experimental designs were quite elaborate and required use of a variety of tools: straight sticks, L-sticks, T-sticks, ladders, boxes, rocks, ribbons, ropes, and coils of wire. The incentive for the animal was a banana or a piece of apple which could not be reached without using one or more of the available tools. The experimental methodology was to let the animals freely experiment with the available tools for a limited time period. If the problem was not solved during that time, the experiment was terminated and repeated at some later time.

In more recent experimental work, Povinelli et al. [8] replicated many of the experiments performed by Köhler and used statistical techniques to analyze the results. The main conclusion was that chimpanzees solve these tasks using simple rules extracted from experience like "contact between objects is necessary and sufficient to establish covariation in movement" [8, p. 305]. Furthermore, it was concluded that chimpanzees do not reason about their own actions and tool tasks in terms of abstract unobservable phenomena such as force and gravity. Even the notion of contact that they have is that of "visual contact" and not "physical contact" or "support" [8, p. 260]. Similar results have been reported by Visalberghi and Trinca [9].

The conclusions of these studies were used to guide the design of the robot's perceptual routines (see Section 4).

2.3 Related Work in Robotics and AI

Krotkov [10] notes that relatively little robotics research has been geared towards discovering external objects' properties other than shape and position. Some of the exploration methods employed by the robot in Krotkov's work use tools coupled with sensory routines to discover object properties. For example, the "whack and watch" method uses a wooden pendulum to strike an object in order to estimate its mass and coefficient of sliding friction. The "hit and listen" method uses a blind person's cane to determine the acoustic properties of objects. Fitzpatrick et al. [11] used a similar approach to program a robot to poke objects with its arm (without using a tool) and learn the rolling properties of the objects from the resulting displacements.

Bogoni and Bajcsy describe a system that evaluates the applicability of differently shaped pointed objects for cutting and piercing operations [12, 13]. A robot manipulator is used to move the tool into contact with various materials (e.g., wood, sponge, plasticine) while a computer vision system tracks the outline of the tool and measures its penetration into the material. The outlines of the tools are modeled by superquadratics and clustering algorithms are used to identify interesting properties of successful tools. This work is one of the few examples in the robotics literature that has attempted to study object functionality with the intention of using the object as a tool by a robot.

Several computer vision projects have focused on the task of recognizing objects based on their functionality [14, 15]. Hand tools are probably the most popular object category used to test these systems. One problem with these systems, however, is that they try to reason about the functionalities of objects without actively interacting with the objects.

3 Behavior-Grounded Tool Representation

3.1 Robots, Tools, and Tasks

Several definitions for tool use have been given in the literature. Arguably, the most comprehensive definition is the one given by Beck [1, p. 10]:

> *"Tool use is the external employment of an unattached environmental object to alter more efficiently the form, position, or condition of another object, another organism, or the user itself when the user holds or carries the tool during or just prior to use and is responsible for the proper and effective orientation of the tool."*

The notion of robotic tool use brings to mind four things: 1) a robot; 2) an environmental object which is labeled a tool; 3) another environmental object to which the tool is applied (labeled an attractor); and 4) a tool task. For tool use to occur all four components need to be present. In fact, it is meaningless to talk about one without taking into account the other three. What might be a tool for one robot may not be a tool for another because of differences in the robots' capabilities. Alternatively, a tool might be suitable for one task (and/or object) but completely useless for another. And finally, some tasks may not be within the range of capabilities of a robot even if the robot is otherwise capable of using tools. Thus, the four components of tool use must always be taken into consideration together.

This is compatible with Gibson's claim that objects afford different things to people with different body sizes. For example, an object might be graspable for an adult but may not be graspable for a child. Therefore, Gibson suggests that a child learns "his scale of sizes as commensurate with his body, not with a measuring stick" [2, p. 235]. For example, an object is graspable if it has opposable surfaces the distance between which is less than the span of the hand [2, p. 133].

Because of these arguments, any tool representation should take into account the robot that is using the tool. In other words, the representation should be *grounded* in the behavioral and perceptual repertoire of the robot. The main advantage of this approach is that the tool's affordances are expressed in concrete terms (i.e., behaviors) that are available to the robot's controller. Note that this is in sharp contrast with other theories of intelligent systems reasoning about objects in the physical world [16, 14]. They make the assumption that object properties can be expressed in abstract form (by a human) without taking into account the robot that will be using them.

Another advantage of the *behavior-grounded* approach is that it can handle changes in the tool's properties over time. For example, if a familiar tool becomes deformed (or a piece of it breaks off) it is no longer the same tool. However, the robot can directly test the accuracy of its representation by executing the same set of exploratory behaviors that was used in the past. If any inconsistencies are detected in the resulting observations they can be used to update the tool's representation. Thus, the accuracy of the representation can be directly tested by the robot.

3.2 Theoretical Formulation

The previous sections presented a justification for the *behavior-grounded* representation. This section formulates these ideas using the following notation.

Let $\beta_{e_1}, \beta_{e_2}, \ldots, \beta_{e_k}$ be the set of exploratory behaviors available to the robot. Each behavior, has one or more parameters that modify its outcome. Let the parameters for behavior β_{e_i} be given as a parameter vector $E_i = [e_1^i, e_2^i, \ldots e_{p(i)}^i]$, where $p(i)$ is the number of parameters for this behavior. The behaviors, and their parameters, could be learned by imitation, programmed manually, or learned autonomously by the robot. In this paper, however, the issue of how these behaviors are selected and/or learned will be ignored.

In a similar fashion, let $\beta_{b_1}, \beta_{b_2}, \ldots, \beta_{b_m}$ be the set of binding behaviors available to the robot. These behaviors allow the robot to attach tools to its body. The most common binding behavior is grasping. However, there are many examples in which a tool can be controlled even if it is not grasped. Therefore, the term *binding* will be used. The parameters for binding behavior β_{b_i} are given as a parameter vector $B_i = [b_1^i, b_2^i, \ldots b_{q(i)}^i]$.

Furthermore, let the robot's perceptual routines provide a stream of observations in the form of an observation vector $O = [o_1, o_2, \ldots, o_n]$. It is assumed that the set of observations is rich enough to capture the essential features of the tasks to which the tool will be applied.

A change detection function, $\mathcal{T}(O(t'), O(t'')) \rightarrow \{0, 1\}$, that takes two observation vectors as parameters is also defined. This function determines if an "interesting" observation was detected in the time interval $[t', t'']$. In the current set of experiments $\mathcal{T} = 1$ if the attractor object was moving during the execution of the last exploratory behavior. The function $\mathcal{T}$ is defined as binary because movement is either detected or it is not.

With this notation in mind, the functionality of a tool can be represented with an *Affordance Table* of the form:

Binding Behavior	Binding Params	Exploratory Behavior	Exploratory Params	O^s	O^e	Times Used	Times Succ

In each row of the table, the first two entries represent the binding behavior that was used. The second two entries represent the exploratory behavior and its parameters. The next two entries store the observation vector at the start and at the end of the exploratory behavior. The last two entries are integer counters used to estimate the probability of success of this sequence of behaviors.

Binding Behavior	Binding Params	Exploratory Behavior	Exploratory Params	O^s	O^e	Times Used	Times Succ
β_{b_1}	$\tilde{b}_1^1$	β_{e_3}	$\tilde{e}_1^3, \tilde{e}_2^3$	$\tilde{O}(t')$	$\tilde{O}(t'')$	4	3

The meanings of these entries are best explained with an example. Consider the following sample row in which the binding behavior β_{b_1} which has one parameter was performed to grasp the tool. The specific value of the parameter for this behavior was $\tilde{b}_1^1$ (a $\tilde{}$ sign is used to represent a specific fixed value). Next, the exploratory behavior β_{e_3} was performed with specific values $\tilde{e}_1^3$ and $\tilde{e}_2^3$ for its two parameters. The value of the observation vector prior to the start of β_{e_3} was $\tilde{O}(t')$ and it value after β_{e_3} has completed was $\tilde{O}(t'')$. This sequence of behaviors was performed 4 times. It resulted in observations similar to the first time this row of the affordance table was created in 3 of these instances, i.e., its probability of success is 75%. Section 6 and Figure 5 provide more information about the organization of the affordance table.

Initially the affordance table is blank. When the robot is presented with a tool it performs a *behavioral babbling* routine which picks binding and exploratory behaviors at random, applies them to the tools and objects, observes their effects, and updates the table. New rows are added to the table only if $\mathcal{T}$ was on while the exploratory behavior was performed. During learning, the integer counters of all rows are set to 1. They are updated during testing trials.

4 Experimental Setup

All experiments were performed using the CRS+ A251 mobile manipulator shown in Figure 1. Five tools were used in the experiments: stick, L-stick, L-hook, T-stick, and T-hook (see Figure 1). The tools were built from pine wood and painted with spray paint. The choice of tools was motivated by the similar tools that Köhler's used in his experiments with chimpanzees [7]. An orange hockey puck was used as an attractor object. The experimental setup is shown in Figure 2 and is described in more detail below.

A Sony EVI-D30 camera was mounted on a tripod overlooking the robot's working area (see Figure 2). The robot's wrist, the tools, and the attractor were color coded so that their positions can be uniquely identified and tracked using

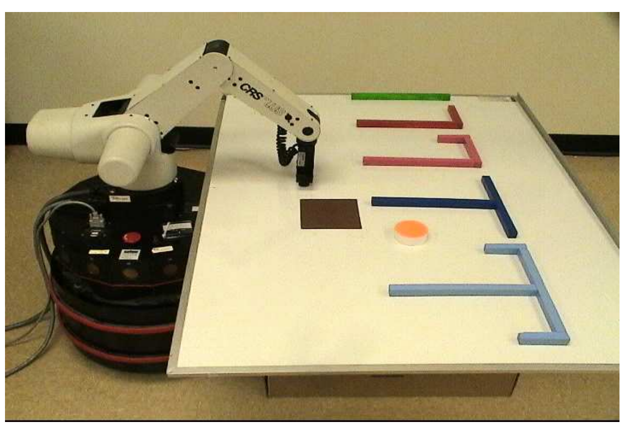

Fig. 1. The figure shows the CRS+ A251 mobile manipulator, the five tools, and the hockey puck that were used in the experiments.

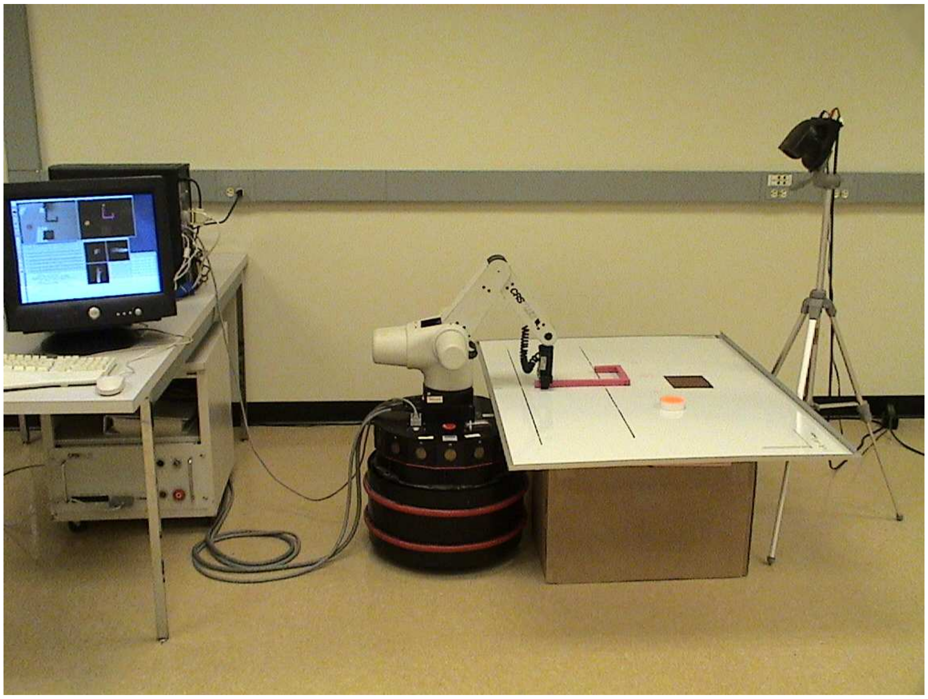

Fig. 2. Experimental setup.

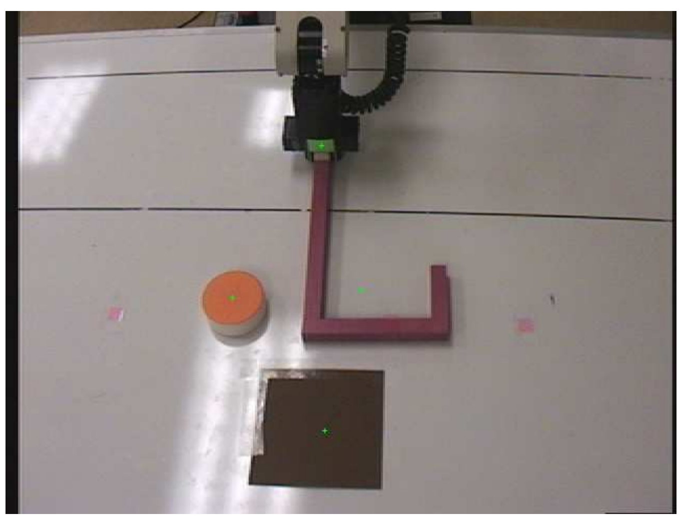

Fig. 3. The image shows the field of view of the robot through the Sony EVI-D30 camera. The robot's wrist, the attractor object, the tools, and the goal region were color coded and their positions were tracked using color segmentation (see Figure 4).

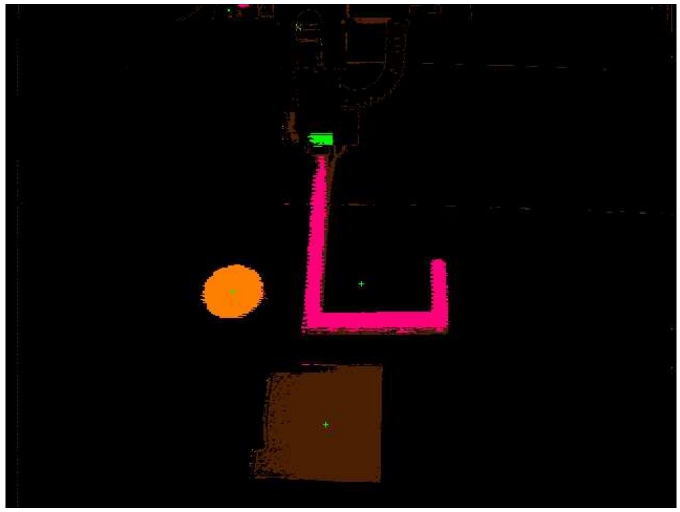

Fig. 4. Color segmentation results for the image frame shown in Figure 3. The positions of the color coded objects were calculated after calibrating the camera using Roger Tsai's method [17, 18]. Although the shape of the tool can be extracted form this image it is not required and used by the behavior-grounded approach.

computer vision (see Figures 3 and 4). The computer vision code was run at 15Hz in 640x480 resolution mode.

To ensure consistent tracking results between multiple robot experiments the camera was calibrated every time it was powered up. A 6×6 calibration pattern was used. The pattern consists of small color markers placed on a cardboard, 5 inches apart, so that they form a square pattern. The pixel coordinates of the 36 uniformly colored markers were identified automatically using color segmentation. The centroid positions of the 36 color markers were used to calculate a mapping function which assigns to each (x,y) in camera coordinates a (X,Y,Z) location in world coordinates. This calculation is possible because the markers are coplanar and the equation of the plane in which they lie is known (e.g., Z=0 is the plane of the table). The mapping function was calculated using Roger Tsai's method [17, 18] and the code given in [19].

5 Exploratory Behaviors

All behaviors used here were encoded manually from a library of *motor schemas* and *perceptual schemas* [3] developed for this specific robot. The behaviors result in different arm movement patterns as described below.

The first four behaviors move the arm in the indicated direction while keeping the wrist perpendicular to the table on which the tool slides. These behaviors have a single parameter which determines how far the arm will travel relative to its current position. Two different values for this parameter were used (2

Exploratory Behaviors	**Parameters**
Extend arm	offset_distance
Contract arm	offset_distance
Slide arm left	offset_distance
Slide arm right	offset_distance
Position wrist	x,y

and 5 inches). The *position wrist* behavior moves the manipulator such that the centroid of the attractor is at offset (x, y) relative to the wrist.

5.1 Grasping Behavior

There are multiple ways in which a tool can be grasped. These represent a set of affordances which we will call first order (or *binding affordances*), i.e., the different ways in which the robot can attach the tool to its body. These affordances are different from the second order (or *output affordances*) of the tool, i.e., the different ways in which the tool can act on other objects. This paper focuses only on output affordances, so the binding affordances were specified with only one grasping behavior. The behavior takes as a parameter the location of a single grasp point located at the lower part of the tool's handle.

5.2 Observation Vector

The observation vector has 12 real-value components. In groups of three, they represent the position of the attractor object in camera-centric coordinates, the position of the object relative to the wrist of the robot, the color of the object, and the color of the tool.

Observation	**Meaning**
o_1, o_2, o_3	X,Y,Z positions of the object (camera-centric)
o_4, o_5, o_6	X,Y,Z positions of the object (wrist-centric)
o_7, o_8, o_9	R,G,B color components of the object
o_{10}, o_{11}, o_{12}	R,G,B color components of the tool

The change detection function $\mathcal{T}$ was defined with the first three components, o_1, o_2, o_3. To determine if the attractor is moving, $\mathcal{T}$ calculates the Euclidean distance and thresholds it with an empirically determined value (0.5 inches). The *times-successful* counter is incremented if the observed attractor movement is within 40 degrees of the expected movement stored in the affordance table.

6 Learning Trials

During the learning trials the robot was allowed to freely explore the properties of the tools. The exploration consisted of trying different behaviors, observing their results, and filling up the affordance table. The initial positions of the attractor and the tool were random. If the attractor was pushed out of tool reach by the robot then the learning trial was temporarily suspended while the attractor was manually placed in a new random position. The learning trials were limited to one hour of run time for every tool.

6.1 What Is Learned

Figure 5 illustrates what the robot can learn about the properties of the T-hook tool based on a single exploratory behavior. In this example, the exploratory behavior is "Contract Arm" and its parameter is "5 inches." The two observation vectors are stylized for the purposes of this example. The information that the robot retains is not the images of the tool and the puck but only the coordinates of their positions as explained above. If a different exploratory behavior was selected by the robot it is possible that no movement of the puck will be detected. In this case the robot will not store any information (row) in the affordance table.

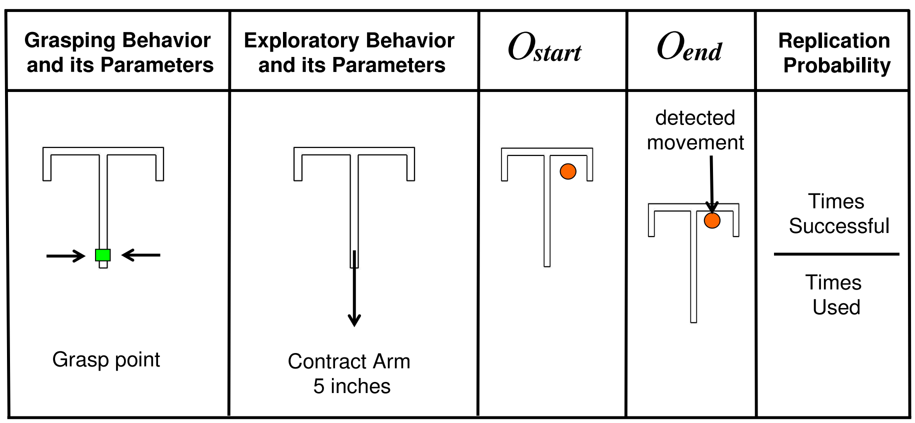

Fig. 5. Contents of a sample row of the affordance table for the T-hook tool.

When the robot performs multiple exploratory behaviors a more compact way to represent this information is required. A good way to visualize what the robot learns is with graphs like the ones shown in Figure 6. The figures show the observed outcomes of the exploratory behaviors when the T-hook tool was applied randomly to the hockey puck. Each of the eight graphs shows the observed movements of the attractor object when a specific exploratory behavior was performed. The movements of the attractor object are shown as arrows. The start of each arrow corresponds to the initial position of the attractor relative to the wrist of the robot (and thus relative to the grasp point) just prior to the start of the exploratory behavior. The arrow represents the observed distance and direction of movement of the attractor in camera coordinates at the end of the exploratory behavior. In other words, each of the arrows shown in Figure 6

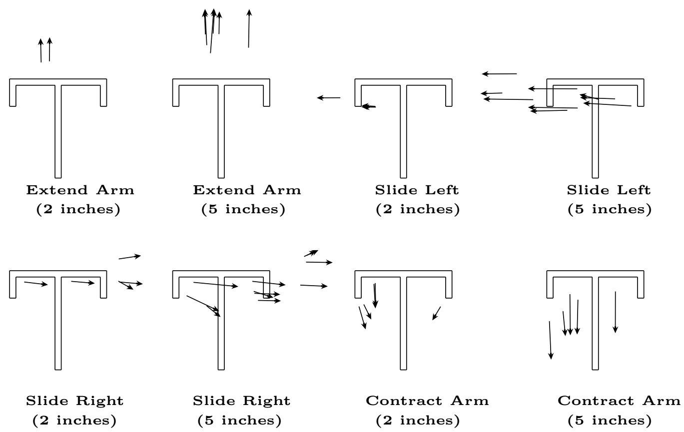

Fig. 6. Visualizing the affordance table for the T-hook tool. Each of the eight graphs show the observed movements of the attractor object after a specific exploratory behavior was performed multiple times. The start of each arrow corresponds to the position of the attractor in wrist-centered coordinates (i.e., relative to the tool's grasp point) just prior to the start of the exploratory behavior. The arrow represents the total distance and direction of movement of the attractor in camera coordinates at the end of the exploratory behavior.

represents one observed movement of the puck similar to the "detected movement" arrow show in Figure 5. The arrows in Figure 6 are superimposed on the initial configuration of the tool and not on its final configuration as in Figure 5.

This affordance representation can also be interpreted as a *predictive* model of the results of the exploratory behaviors. In other words, the affordances are represented as the expected outcomes of specific behaviors. This interpretation of affordances is consistent with the idea that biological brains are organized as predictive machines that anticipate the consequences of actions – their own and those of others [20, p. 1]. It is also consistent with some recent findings about the internal representation of the functional properties of novel objects and tools in humans. For example, "if the brain can predict the effect of pushing or pulling an object this is effectively an internal model of the object that can be used during manipulation"[6]. A recent result in the theoretical AI literature also shows that the state of a dynamic system can be represented by the outcomes of a set of tests [21, 22]. The tests consist of action-observation sequences. It was shown that the state of the system is fully specified if the outcomes of a basis set of test called *core tests* are known in advance [22].

6.2 Querying the Affordance Table

After the affordance table is populated with values it can be queried to dynamically create behavioral sequences that solve a specific tool task. The behaviors in these sequences are the same behaviors that were used to fill the table. This subsection describes the search heuristic used to select the best affordance for

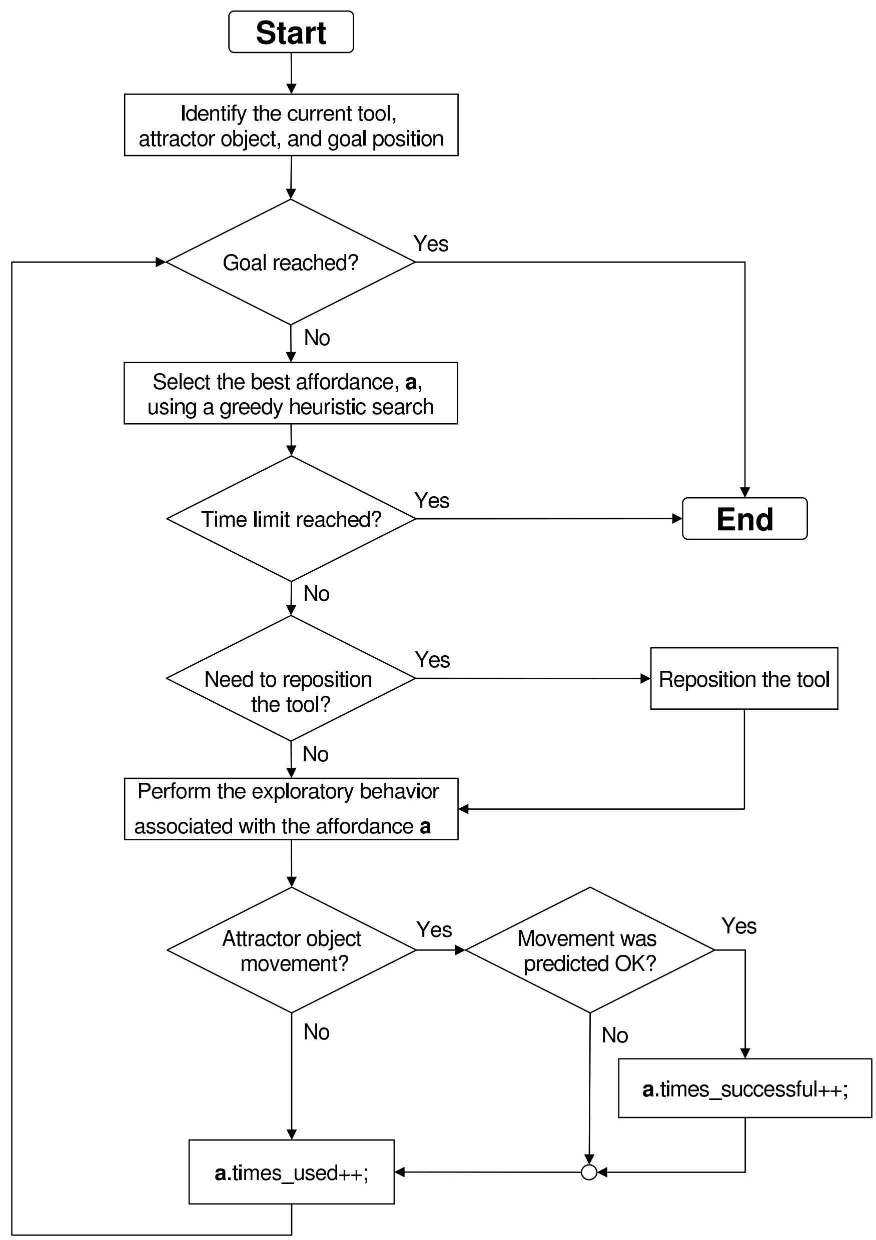

Fig. 7. Flowchart diagram for the procedure used by the robot to solve tool-using tasks with the help of the behavior-grounded affordance representation.

the current task configuration. This heuristic is used by the procedure for solving tool-using tasks shown in Figure 7

During testing trials, the best affordance for a specific step in a tool task was selected using a greedy heuristic search. The query method that was adopted uses empirically derived heuristics to perform multiple nested linear searches through the affordance table. Each successive search is performed only on the rows that were not eliminated by the previous searches. Four nested searches were performed in the order shown below:

1) Select all rows that have observation vectors consistent with the colors of the current tool and object.

2) From the remaining rows select those with probability of success greater than 50%. In other words, select only those rows that have a replication probability (times_successful/times_used) greater than $\frac{1}{2}$ (the reasons for choosing this threshold value are described below).

3) Sort the remaining rows (in increasing order) based on the expected distance between the attractor object and the goal region if the behavior associated with this row were to be performed.

4) From the top 20% of the sorted rows choose one row which minimizes the re-positioning of the tool relative to its current location.

As it was mentioned above the greedy one-step-lookahead heuristic was derived empirically. The performance of the heuristic was fine tuned for speed of adaptation in the presence of uncertainty which is important when multiple robot trials have to be performed. For example, the threshold value of 50% used in step 2 above was chosen in order to speed up the elimination of outdated affordances when the geometry of the tool suddenly changes (see the experiment described in Section 7.2). With this threshold value it takes only one unsuccessful behavioral execution in order to eliminate an affordance from further consideration. Future work should attempt to formulate a more principled approach to this affordance-space planning problem, preferably using performance data derived from tool-using experiments with animals and humans (e.g., [6]).

7 Testing Trials

Two types of experiments were performed to test the behavior-grounded approach. They measured the quality of the learned representation and its adaptation abilities when the tool is deformed, respectively.

7.1 Extension of Reach

In the first experiment the robot was required to pull the attractor over a color coded goal region. Four different goal positions were defined. The first goal position is shown in Figure 1 (the dark square in front of the robot). The second goal position was located farther away from the robot (see Figure 2). To achieve it the robot had to push the attractor away from its body. Goals 3 and 4 were placed along the mid-line of the table as shown in Figure 8.

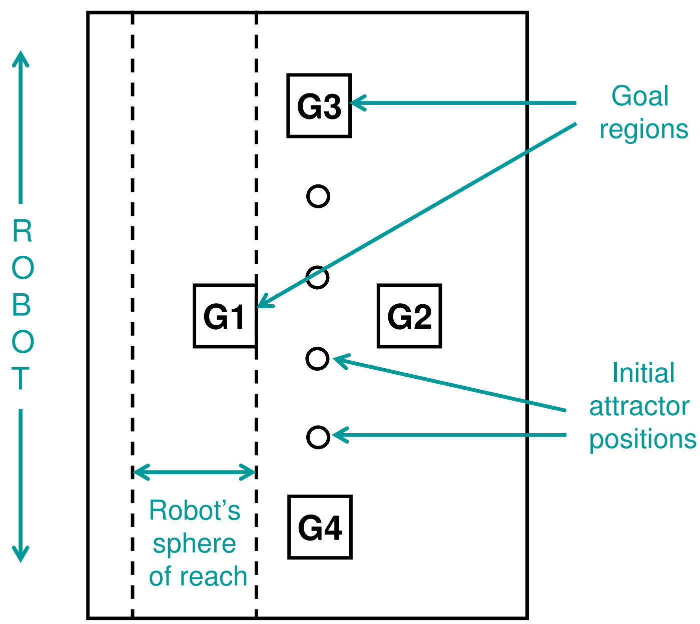

Fig. 8. The figure shows the positions of the four goal regions (G1, G2, G3, and G4) and the four initial attractor positions used in the extension of reach experiments. The two dashed lines indicate the boundaries of the robot's sphere of reach when it is not holding any tool.

In addition to that there were 4 initial attractor positions per goal. The initial positions are located along the mid-line of the table, 6 inches apart as shown in Figure 8. The tool was always placed in the center of the table. A total of 80 trials were performed (4 goals $\times$ 4 attractor positions $\times$ 5 tools). The table below summarizes the results. The values represent the number of successful solutions per goal, per tool. Four is the maximum possible value as there are only four initial starting positions for the attractor object.

Tool	**Goal 1**	**Goal 2**	**Goal 3**	**Goal 4**
Stick	0	2	4	4
L-stick	4	2	4	4
L-hook	4	3	4	4
T-stick	3	3	4	4
T-hook	4	4	4	4

As can be seen from the table, the robot was able to solve this task in the majority of the test cases. The most common failure condition was due to pushing

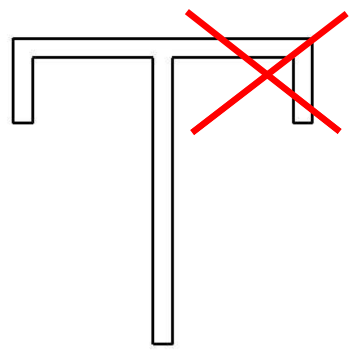

Fig. 9. A T-hook missing its right hook is equivalent to an L-hook.

the attractor out of tool's reach. This failure was caused by the greedy one-step-lookahead heuristic used for selecting the next tool movement. If the robot plans the possible movements of the puck for 2 or 3 moves ahead these failures will be eliminated. A notable exception is the Stick tool, which could not be used to pull the object back to the near goal (G1). The robot lacks the required exploratory behavior (*turn-the-wrist-at-an-angle-and-then-pull*) that is required to detect this affordance of the stick. Adding the capability of learning new exploratory behaviors can resolve this problem.

7.2 Adaptation After a Tool Breaks

The second experiment was designed to test the flexibility of the behavior-grounded representation in the presence of uncertainties. The uncertainly in this case was a tool that can break. For example, Figure 9 shows the tool transformation which occurs when a T-hook tool loses one of its hooks. The result is a L-hook tool. This section describes the results of an experiment in which the robot was exposed to such tool transformation after it had already learned the affordances of the T-hook tool.

To simulate a broken tool, the robot was presented with a tool that has the same color ID as another tool with a different shape. More specifically, the learning was performed with a T-hook which was then replaced with an L-hook. Because color is the only feature used to recognize tools the robot believes that it is still using the old tool.

The two tools differ in their upper right sections as shown in Figure 9. Whenever the robot tried to use affordances associated with the missing parts of the tool they did not produce the expected attractor movements. Figure 10 shows frames from a sequence in which the robot tried in vain to use the upper right part of the tool to move the attractor towards the goal. After several trials the replication probability of the affordances associated with that part of the tool was reduced and they were excluded from further consideration. Figure 11 shows frames from the rest of this sequence in which the robot was able to complete the task with the intact left hook of the tool.

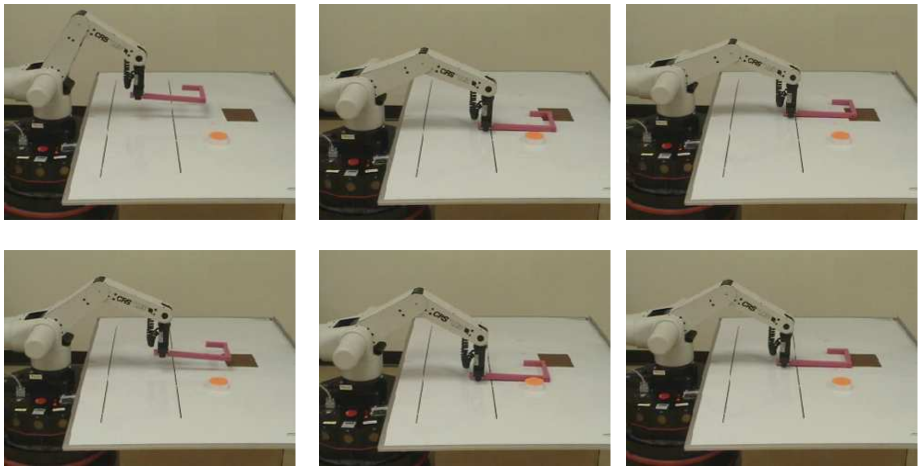

Fig. 10. Using a broken tool (Part I: Adaptation) - Initially the robot tries to move the attractor towards the goal using the missing right hook. Because the puck fails to move as expected the robot reduces the replication probability of the affordances associated with this part of the tool.

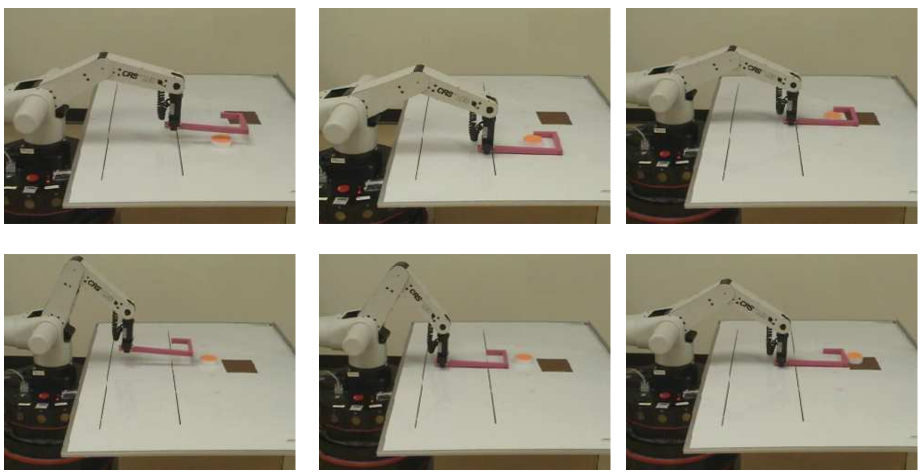

Fig. 11. Using a broken tool (Part II: Solving the task) - After adapting to the modified affordances of the tool, the robot completes the task with the intact left hook

A total of 16 trials similar to the one shown in Figure 10 were performed (i.e., 4 goal regions $\times$ 4 initial attractor positions). In each of these experiments the robot started the testing trial with the original representation for the T-hook tool and modified it based on actual experience. The robot was successful in all 16 experiments, i.e., the robot was able to place the attractor over the target goal region with the “broken” tool in all 16 experiments.

8 Conclusions and Future Work

This paper introduced a novel approach to representing and learning tool affordances by a robot. The affordance representation is *grounded* in the behavioral and perceptual repertoire of the robot. More specifically, the affordances of different tools are represented in terms of a set of exploratory behaviors and their resulting effects. It was shown how this representation can be used to solve tool-using tasks by dynamically sequencing exploratory behaviors based on their expected outcomes.

The behavior-grounded approach represents the tool's affordances in concrete terms (i.e., behaviors) that are available to the robot's controller. Therefore, the robot can directly test the accuracy of its tool representation by executing the same set of exploratory behaviors that was used in the past. If any inconsistencies are detected in the resulting observations they can be used to update the tool's representation. Thus, the accuracy of the representation can be directly tested by the robot. It was demonstrated how the robot can use this approach to adapt to changes in the tool's properties over time, e.g., tools that can break.

A shortcoming of the behavior-grounded approach is that there are tool affordances that are unlikely to be discovered since the required exploratory behavior is not available to the robot. This problem has also been observed in animals, e.g., macaque monkeys have significant difficulties learning to push an object with a tool away from their bodies because this movement is never performed in their normal daily routines [23]. This problem can be resolved, however, if the ability to learn new exploratory behaviors is added.

There are some obvious extensions to this work that are left for future work. First, the current implementation starts the exploration of a new tool from scratch even though it may be similar to an already explored tool. Adding the ability to rapidly infer the affordances of a new tool from its shape similarity to previous tools would be a nice extension.

Second, the current implementation uses a purely random behavioral babbling exploration procedure. Different strategies that become less random and more focused as information is structured by the robot during the exploration could be used to speed up the learning process.

Third, the behavior-grounded approach should be compared experimentally with planners for pushing objects (e.g., [24]). We expect that the behavior-grounded method would approach asymptotically the accuracy of these planners as the number and diversity of the exploratory behaviors is increased. We also expect, however, that our approach would excel in situations that cannot be predicted by the planners, e.g., tools that can break or objects whose center of mass can shift between trials.

References

1. Beck, B.B.: Animal Tool behavior: The use and manufacture of tools by animals. Garland STMP Press, New York (1980)
2. Gibson, J.J.: The ecological approach to visual perception. Houghton Mifflin, Boston (1979)

3. Arkin, R.: Behavior-based robotics. MIT Press (1998)
4. Power, T.G.: Play and Exploration in Children and Animals. Lawrence Erlbaum Associates, Publishers, Mahwah, NJ (2000)
5. Lorenz, K.: Innate bases of learning. In Pribram, K.H., King, J., eds.: Learning as Self-Organization. Lawrence Erlbaum Associates, Publishers, Mahwah, NJ (1996)
6. Mah, C.D., Mussa-Ivaldi, F.A.: Evidence for a specific internal representation of motion-force relationships during object manipulation. Biological Cybernetics **88**(1) (2003) 60–72
7. Köhler, W.: The mentality of apes. Harcourt, Brace, and Co. (1931)
8. Povinelli, D., Reaux, J., Theall, L., Giambrone, S.: Folk Physics for Apes: The Chimpanzee's theory of how the world works. Oxford Univ. Press, Oxford (2000)
9. Visalberghi, E., Trinca, L.: Tool use in capuchin monkeys: distinguishing between performing and understanding. Primates **30** (1989) 511–21
10. Krotkov, E.: Perception of material properties by robotic probing: Preliminary investigations. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Montreal (August 1995) 88–94
11. Fitzpatrick, P., Metta, G., Natale, L., Rao, S., Sandini, G.: Learning about objects through action - initial steps towards artificial cognition. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan (May 12-17, 2003)
12. Bogoni, L., Bajcsy, R.: Interactive recognition and representation of functionality. Computer Vision and Image Understanding **62**(2) (1995) 194–214
13. Bogoni, L.: Identification of functional features through observations and interactions. PhD thesis, University of Pennsylvania (1995)
14. Stark, L., Bowyer, K.: Generic Object Recognition using Form and Function. Volume 10 of Machine Perception and AI. World Scientific, Singapore (1996)
15. Rivlin, E., Dickinson, S.J., Rosenfeld, A.: Recognition by functional parts. Computer Vision and Image Understanding **62**(2) (1995) 164–176
16. Hayes, P.J.: The second naive physics manifesto. In: Formal Theories of the Commonsense World. Ablex (1985)
17. Tsai, R.Y.: An efficient and accurate camera calibration technique for 3D machine vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami Beach, FL (1986) 364–374
18. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation **RA-3**(4) (August 1987) 323–344
19. Willson, R.: Tsai Camera Calibration Software. (1995) (the source code can be downloaded from http://www-2.cs.cmu.edu/~rgw/TsaiCode.html).
20. Berthoz, A.: The brain's sense of movement. Harvard University Press (2000)
21. Singh, S., Littmn, M., Sutton, R., Stone, P.: Learning predictive state representations. Paper Draft (unpublished) (2002)
22. Littman, M.L., Sutton, R.S., Singh, S.: Predictive representation of state. In: Advances in Neural Information Processing Systems. Volume 14. (2002)
23. Ishibashi, H., Hihara, S., Iriki, A.: Acquisition and development of monkey tool-use: behavioral and kinematic analyses. Canadian Journal of Physiology and Pharmacology **78** (2000) 958–966
24. Mason, M.T.: Mechanics of Robotic Manipulation. MIT Press (2001)