Hierarchical Skills and Skill-based Representation

Shiraj Sen and **Grant Sherrick** and **Dirk Ruiken** and **Rod Grupen**
Laboratory for Perceptual Robotics
Computer Science Department
University of Massachusetts Amherst
{shiraj, sherrick, ruiken, grupen}@cs.umass.edu

Abstract

Autonomous robots demand complex behavior to deal with unstructured environments. To meet these expectations, a robot needs to address a suite of problems associated with long term knowledge acquisition, representation, and execution in the presence of partial information. In this paper, we address these issues by the acquisition of broad, domain general skills using an intrinsically motivated reward function. We show how these skills can be represented compactly and used hierarchically to obtain complex manipulation skills. We further present a Bayesian model using the learned skills to model objects in the world, in terms of the actions they afford. We argue that our knowledge representation allows a robot to both predict the dynamics of objects in the world as well as recognize them.

1 Introduction

Robot programming traditionally involves specific sensory and motor sequences learned or coded in for each use case, with little or no transfer from case to case. This is due to the lack of a knowledge representation that supports transfer between contexts. Representations that are useful for continuous control of robotic systems and discrete, symbolic reasoning presents a significant challenge for integrating reasoning and control research in robotics. Ideally, these areas of research should be able to inform one other.

In this paper, we present a solution to this representational discontinuity by having the agent build a functional model of the world in terms of sensorimotor programs and uses it to create efficient behavioral contingencies for novel runtime situations. The sensorimotor programs - *schemas* are acquired using an intrinsic reward function for generalizable control programs (Hart 2009a). The schemas are then composed hierarchically to learn complicated motor programs for grasping and manipulation. A Bayesian framework for modeling control knowledge in the environment is then presented. We show how the model provides a functional description of objects and hence can be used for both object recognition as well as action selection.
Section 2 describes the framework for acquiring closed loop programs by using an intrinsically motivated reward function. The mathematical framework for representing these motor programs (and compositions of them) is also presented. Section 3 shows how these control programs can be used by the robot to organize knowledge about objects in the world. Section 4 presents an algorithm that describes the utility of this representation as a predictive model for action selection. Section 5 presents a discussion regarding extensions to the current implementation including learning relationships between objects.

2 Control Programs - SEARCHTRACK

Primitive control actions, $c \equiv \phi|_\tau^\sigma$ are closed-loop feedback controllers that are constructed by combining potential functions, ϕ, with feedback signals, σ, and motor resources, τ ($\langle \sigma \subseteq \Omega_\sigma, \phi \in \Omega_\phi, \tau \subseteq \Omega_\tau \rangle$). The sensitivity of the potential to changes in the motor variables provides a control gradient that is used to derive reference inputs ($\mathbf{u}_\tau$) for synergies of motor units defined by subsets, $\tau \subseteq \Omega_\tau$, of the robot's effector resources. The error dynamics created when the controller interacts with the environment provides a natural discrete abstraction of the underlying continuous state space (Coelho and Grupen 1997). In this work, we employ a four level discrete logic, $p(c) \in \{X, -, 0, 1\}$, where 'X' indicates unknown control state, '$-$' indicates that the reference signal is not available, '0' indicates the transient control response and '1' denotes convergence/quiescence. A collection of n distinct primitive control actions forms a discrete state space $\mathbf{s}^k = [p_1, \cdots, p_n]^k \in \mathcal{S}$.

There are two distinct types of actions that share potential functions and effector resources, but are distinguished by the source of their input signals : TRACK and SEARCH. A TRACK action uses effectors, τ, to track a reference signal, σ. A convergence event ($0 \rightarrow 1$) is considered rewarding to the learning agent if the reference stimuli being tracked belongs to the external environment (Hart 2009b). This provides a computational approach to learning concepts analogous to Gibsonian *affordances* in which the potential for action is explicitly modeled (Gibson 1977). We say that the environment affords controller c_i when this control action causes a $0 \rightarrow 1$ state transition.

A SEARCH action "orients" the sensorimotor resources to discover trackable affordances. The search actions are of the form $\phi|_\tau^{\tilde{\sigma}}$ (sharing potential functions and effector

resources with their TRACK counterparts), deriving their input, $\tilde{\sigma}$ by sampling from probabilistic models describing distributions over effector reference values($\mathbf{u}_\tau$) where rewarding TRACKing actions have been discovered in the past, $p(\phi|^{\sigma}_{\tau}) = 1$. The effector reference values are learned relative to the spatial attributes, f of the trackable feature (position, orientation, scale). Initially the distribution $Pr(\mathbf{u}_\tau|p(\phi|^{\sigma}_{\tau}) = 1)$ is uniform; however, as it is updated over the course of many learning episodes, this distribution will reflect the long term statistics of the run-time environment.

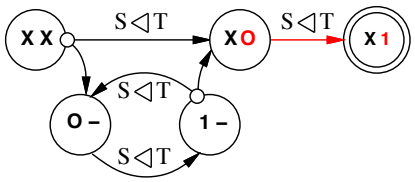
Sequential programs can be assembled out of control primitives by using Reinforcement Learning (RL) (Sutton and Barto 1998). Hart (Hart 2009a) showed that restricting the sensory and effector resources to which the robot has access can lead to the acquisition of new and interesting behavior. In the simplest context, the robot was restricted to proprioceptive feedback from the pan/tilt head and large scale motion cues arising from a single camera. Effector resources were likewise restricted to motor controllers associated with the pan and tilt axes of the visual system. Under this developmental context, the control basis yields a small variety of SEARCH and TRACK actions,

$$\mathcal{A} = \{\widetilde{\phi|^{(u,v)}_{pt}}, \quad \phi|^{(u,v)}_{pt}, \quad (\widetilde{\phi|^{(u,v)}_{pt}} \triangleleft \phi|^{(u,v)}_{pt}), \quad (\phi|^{(u,v)}_{pt} \triangleleft \widetilde{\phi|^{(u,v)}_{pt}})\}$$

where pt designates the pan and tilt axes of the head as a sensor (in the superscript position) or an effector (as a subscript) and (u, v) designates the centroid of the motion cue relative to the image center. The shorthand, $c_2 \triangleleft c_1$ (read "c_2 *subject-to* c_1") is used in the following to describe priority relations in concurrent control actions achieved by projecting subordinate actions into the nullspace of superior actions (Nakamura, Hanafusa, and Yoshikawa 1987). The only rewarding event that can be generated by these set of actions is the convergence of the TRACK-ing controller $\phi|^{(u,v)}_{pt}$. Therefore, the developmental context is designed to teach the robot how to find and track motion features.

The state space defined by this developmental stage is the vector of controller states $\mathbf{q} = [p^{search} \quad p^{track}]$. Figure 1(a) shows the SEARCHTRACK schema acquired after 25 learning trials in this developmental context using Q-learning. In Figure 1(a), concurrent SEARCH and TRACK control actions are permitted and the resulting SEARCHTRACK policy begins by attempting to concurrently SEARCH for and TRACK a motion cue. If a motion cue exists in the signal, the policy attempts to continue TRACK-ing. If no target is immediately available, the policy selects a SEARCH behavior in which it samples new pan/tilt configurations from the distribution in Figure 1(b) in a loop until the target stimulus is found at which point, the policy tracks the features and receives reward on the $0 \rightarrow 1$ transition. The shorthand, $\Phi|^{\sigma}_{\tau}$ is used to describe a SEARCHTRACK program for tracking a signal, σ using effector resources, τ.

Control programs (SEARCHTRACK schema) can further be sequenced to generate complex programs that can find and track multiple stimuli in the environment. Hart (Hart, Sen, and Grupen 2008) presented a detailed description



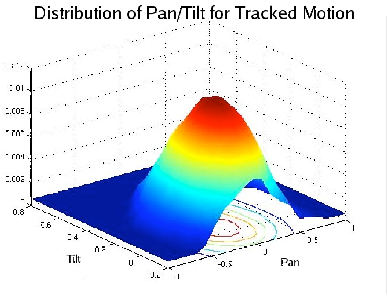


Figure 1: Action schemas that represent SEARCHTRACK behavior in terms of state $[p^{search} \quad p^{track}]$. A new SEARCH goal is sampled whenever SEARCH is executed from states for which $p^{search} = (X||1)$ (designated by small circles). The schema in panel (a) uses co-articulated action. Panel (b) shows the resulting distribution $Pr(\mathbf{u}_\tau|p(\phi|^{\sigma}_{\tau}) = 1)$ after 50 presentations.

of the various manipulation programs (touching, grasping, picking up, placing and inspecting objects) that can be learned in a hierarchical fashion from the previously acquired control programs. All of these programs can be viewed abstractly as a sequence of SEARCHTRACK programs that can be used to find and track independent, generic features : visual; tactile; invariants in many sensor signals (e.g., grasping, pick-and-place and manipulation tasks), and each serves as an orienting action for detecting dependent signals. Figure 2 shows a hierarchical schema learned by the robot to reliably track a reference force using its end effector. The learned program REACHGRASP involves concurrently tracking multiple visual stimuli (indicated by the '+' symbol for the first schema) followed by a SEARCHTRACK schema which tracks forces using its end effector. In this hierarchical schema, the Cartesian feature tracker becomes part of the search behavior that orients the robot to get more reward. Thus schemas can be used hierarchically as a temporally extended action if it leads to more reward.

The use of the term "schema" was proposed by the German philosopher Immanuel Kant (Kant 1965) as a way of mapping concepts to percepts over categories of objects. He talked about grounding concepts in sensations that would lend support to reasoning and intuition. Jean Piaget suggested that schema are formed to meet new demands through a process of *accommodation* and that existing schema respond to new experiences through *assimilation* (Piaget 1952). Computational schema have been demonstrated in rule-based systems (Nilsson 1994) and empirical cause-and-effect systems in discrete domains (Drescher 1991), as well

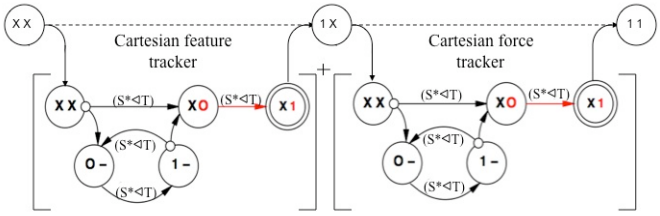


Figure 2: Sequential programs can be learned by sequencing a set of previously learned SEARCHTRACK schemas. The robot learns how to "grasp" by sequencing two different SEARCHTRACK schemas that establishes spatial features followed by invariants in the force domain associated with prehensile behavior. The '+' sign for the first schema indicates that the robot might need to track multiple different spatial features before it can reliably track a force. The shorthand REACHGRASP will be used to describe the grasp schema.

as continuous domains that can be explored through active learning (Mugan and Kuipers 2007). Lyons (Lyons 1986) presented a schema theory approach for designing a formal language for robot programming called Robot Schema (RS). In this approach, perceptual and motor schemas are combined into coordinated control programs (Arbib 1995).

Our computational framework acquires programs for controlling interaction with the environment and manages redundant sensory and motor resources to discover and maintain intrinsically rewarding relationships in dynamic environments. The acquired control programs and their long term statistics represent a domain general way of interacting with stimuli in the environment. The schemas capture *common sense* knowledge acquired by the robot. The environment, however, presents important kinds of structure in terms of *objects* — sets of spatially related co-affordances. In the next section, a Bayesian framework for acquiring these domain specific knowledge structures in terms of distribution over SEARCHTRACK programs is presented.

3 Control Affordances in the Environment - Objects

Representing knowledge about the world in terms of affordances, provides a powerful and computationally efficient way for an agent to encode its experiences. Since the formulation of the theory of affordances by J. J. Gibson (Gibson 1977), a great deal of work has been done to formalize this concept in a manner that can be modeled computationally. Specifically, Stoytchev (Stoytchev 2005b), (Stoytchev 2005a) and Fitzpatrick (Fitzpatrick et al. 2003) showed that affordance-related concepts can be used to differentiate objects in the course of interaction with the environment. In this work, we propose the use of distinct patterns of control affordances as a representation for objects with which we plan sensory and motor interactions. We describe an object with id i as a spatial distribution over the state of N_i-control affordances. This representation defines an affordance, given

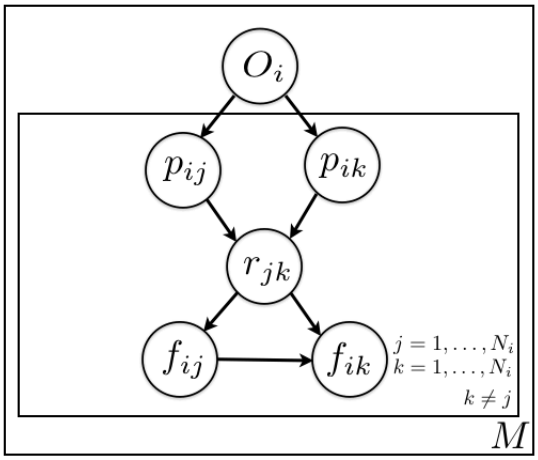


Figure 3: A Bayesian network model G representing objects $O_1, \cdots, O_M$ as a spatial distribution over N control affordances. The random variables p_{i*} model the state of the SEARCHTRACK schemas. f_{i*} models the position, orientation and scale of a feature in the world frame as observed by the robot. r_{jk} models the relative distance, orientation and scale between two trackable actions. This variable encodes the spatial dependencies of various affordances.

the probable existence of one or more objects, according to not only its sensor signal σ and effector τ, but also by its location, orientation, and scale with respect to other affordances.

Figure 3 is a Bayesian network that encodes the logical dependencies between the variables of the environment affordance model. Each of the M modeled objects is represented by a Bernoulli random variable $O_i : i = 1, \cdots, M$ denoting the probability that an object exists in the current environment. For each object, there exists N_i affordances that have a non-zero probability of occurring. Each of these affordances is represented by a multinomial random variable p_{i*} describing the stable dynamics of each associated SEARCHTRACK action. This spatial region is modeled by f_{i*} as a spatial blob with mean position, x, y, z, orientation of the principal axis, θ and length of the principal axis, sc. The spatial relationships between control affordances - relative distance, orientation and scale are modeled by the nodes $r_{jk} : j = 1, \cdots, N_i,\ k = 1, \cdots, N_i,\ j \neq k$. The resulting model provides a generative manner of describing objects, affordances, and the relationships between them. Utilizing past experience encoded as priors, this model is able to aid in accomplishing new tasks with the same or similar objects.

One of the biggest advantages of representing objects in this manner is that it allows a robot to interact with its environment, observe the effects of these interactions, and then to make predictions about future actions while incorporating task specific constraints. For instance, given an object recognition task with observations $Z = \{z^p, z^f\}$ where $z^p \subseteq p(\Phi|^{\sigma}_{\tau})$, $z^f \subseteq f$, the distribution over likely objects can be found by computing $Pr(O_i|Z)$. In addition, it is pos-

sible to make decisions concerning control actions the robot can execute (or not execute) on an object ($Pr(p_j|Z)$). In the next section, we briefly describe how the affordance-based representation can be used by a robot to intelligently select actions to achieve a task.

4 Task Specific Action Selection

Ideally the goals for an action, f_{ij}, can be sampled from the Bayesian model given the environment affordance model and observations. However, in the presence of partial information, choosing an action given that it may be expensive or destructive (w.r.t. sensor measurements) requires safeguards to ensure that the robot chooses the next action that will optimally lead towards successfully completing its intended task. The procedure for taking such an action (a_g) is described in Algorithm 1.

In the beginning, when the robot hasn't discovered any affordances in its environment, the only evidence (E) available is the priors over objects from the trained model. The inference algorithm proceeds by selecting actions that lead to a maximum reduction in uncertainty over the distribution of task goals. This is achieved by computing the mutual information between the goals and other possible affordances given the evidence. The action which is predicted to have the maximal mutual information is the one that the robot executes next to optimally reduce its uncertainty over the goal affordance. This process is repeated until the uncertainty in the goal affordance is low enough for the robot to try executing the goal action.

Algorithm 1 TASKGOAL(a_g, ϵ, E)

Evidence, $e \leftarrow \{E\}$
Discovered Affordances, $A \leftarrow \{\}$
repeat
  Compute posterior over goal region given evidence of affordances, $\Pr(f_g|e, p_g)$
  Compute Entropy over goal affordance given evidence, $H_g \leftarrow H(f_g|e, p_g)$
  if $H_g < \epsilon$ **then**
    Execute a_g with $f_g \sim \Pr(f_g|e, p_g)$
    $e \leftarrow e \bigcup p(a_g) \bigcup f_g$
  else
    for all $a_i \notin A \bigcup a_g$ **do**
      Compute posterior of possible regions of affordance, $\Pr(f_i|e, p_i)$
      Compute mutual Information, I_{a_i} between the goal and affordance a_i, $I((f_g|e, p_g); (f_i|e, p_i))$
    end for
    Select the action with maximum mutual information, $a_{next} = \arg\max_j \ I_{a_j}$
    Execute action, a_{next}.
    Make observation, $f_{next} \leftarrow < x, y, z, \theta, sc >$
    $e \leftarrow e \bigcup p(a_{next}) \bigcup f_{next}$
    $A \leftarrow A \bigcup a_{next}$
  end if
until $p(a_g) = 1$ {Goal Action Succeeds}

Example: Radio

As a proof of concept, we describe here a grasping task for a radio given an empirically derived environment affordance model containing only a single object, the radio. In Figure 4, the full radio model can be seen. The affordances that we chose to represent were the yellow knob (in the center), the black antenna, the green bottom piece, the bounding box, and reach goals for a grasp oriented along the principal axis of the object. We assume in this instance prior information indicating that this object is on a table or that the object was only seen on a table during training implying that the model is ignorant of grasping points that would be impossible while laying on a table. Additionally, the robot is assumed to only have knowledge of one type of grasp, which aligns the hand to the principal axis of the object.

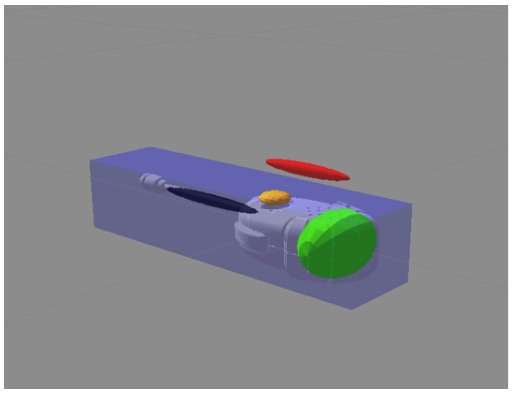


Figure 4: 3D model visualization of radio model G_{radio}, red = REACHGRASP, green = (visual) SEARCHTRACK green, yellow = (visual) SEARCHTRACK yellow, black = (visual) SEARCHTRACK black, blue = (visual) SEARCHTRACK bounding box

Figure 5 shows the resulting posterior of grasp goals, $\Pr(f_g|e, p_g)$ after one round of action selection in Algorithm 1. In the first round, the algorithm chooses the (visual) SEARCHTRACK green action because the predicted resultant feature provides the maximum decrease in uncertainty of the grasping posterior. It is interesting to note here the dilation occurring to the reach goal distribution, pictured in red, for each possible affordance. Because the prior for each affordance is modeled with a multivariate Gaussian, there is an inherent symmetry introduced. This results in bilateral symmetry in the majority of cases, but because of the ambiguity with respect to orientation present in the round knob, pictured in yellow, this case results in a rotational symmetry about the centroid of the knob. This will be true for any object that does not have well defined principal axes.

Figure 6 shows the results after a second round of action selection. After combining the new evidence obtained from the execution of this action, the uncertainty in the goal affordance goes below the desired threshold.

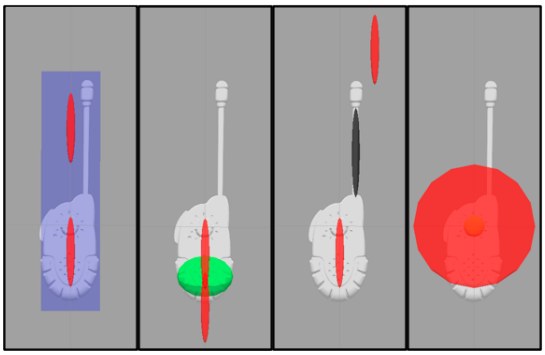


Figure 5: Visualization of the grasp posterior (red) given the environment model with the radio object and the existence of each other affordance (Panel a: bounding box, Panel b: green bottom piece, Panel c: black antenna, Panel d: knob) as evidence.

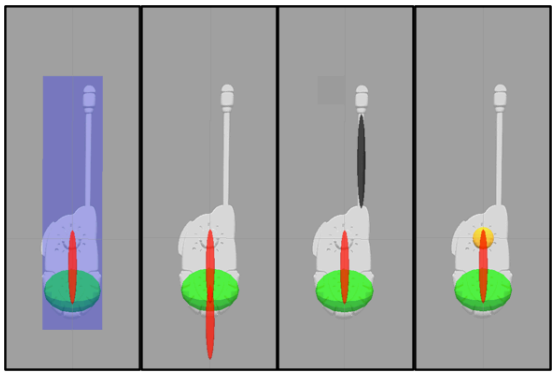


Figure 6: Visualization of the grasp posterior given the environment model and one round of action selection.

5 Discussion

The above Bayesian formulation of objects as spatially structured schemas provides a powerful mechanism for autonomous learning and planning for a robot performing manipulation tasks. However, until now, we have only considered the case where one object is present in the environment. This is almost always not the case. For example, to grasp a tool lying on a table, the robot needs to interact with two objects - tool and table. While each object in isolation can be described by their affordance model, the model distributions change when objects are interacting (or maintaining certain spatial relationships) with one another. An action that reaches for the tool lying on the table cannot choose any grasp goal on the tool that is in contact with the table. The presented Bayesian formulation provides a principled way of re-computing these distributions based on observations that are consistent with multiple objects.

The distribution, $Pr(f_{ij}|p_{ij}, O_i)$, which describes a feature in the robot's sensor space, changes in the presence of other objects. Given a reference, f_{ij} for a schema, $\Phi|_\tau^\sigma$ afforded by object, O_i, the posterior of that reference can be recomputed to reduce the probability in locations where this action is afforded by one object but not by the other.

6 Conclusions

In this paper, we introduced a knowledge representation framework that organizes knowledge about objects in terms of long term statistics of controllable interaction. We showed how a robot can acquire broad domain general schemas by using an intrinsic reward function that favors finding new affordances in the environment. These schemas are reused to learn complex manipulation skills and provide a basis set for modeling objects in the environment as a distribution over spatially located co-affordances. We provided some preliminary results of using this approach for modeling objects and using these models for object recognition and action selection. We are presently working on applying these ideas on a real robot.

Acknowledgment

This work was supported by the DARPA grant iRobot Corp. Prime Army W91CRB-10-C-0127.

References

Arbib, M. 1995. Schema theory. In *The Handbook of Brain Theory and Neural Computation*, 830–834. Cambridge, MA: MIT Press.

Coelho, J., and Grupen, R. 1997. A control basis for learning multifingered grasps. *Journal of Robotic Systems* 14(7):545–557.

Drescher, G. 1991. *Made-Up Minds: A Constructionist Approach to Artificial Intelligence*. Cambridge, MA: MIT Press.

Fitzpatrick, P.; Metta, G.; Natale, L.; Rao, S.; and Sandini, G. 2003. Learning about objects through action: Initial steps towards artificial cognition. In *IEEE International Conference on Robotics and Automation*.

Gibson, J. 1977. The theory of affordances. In *Perceiving, acting and knowing: toward an ecological psychology*, 67–82. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Hart, S.; Sen, S.; and Grupen, R. 2008. Intrinsically motivated hierarchical manipulation. In *Proceedings of 2008 IEEE Conference on Robotics and Automation*.

Hart, S. 2009a. *The Development of Hierarchical Knowledge in Robot Systems*. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts Amherst.

Hart, S. 2009b. An intrinsic reward for affordance exploration. In *International Conference on Development and Learning*.

Kant, I. 1965. *Critique of Pure Reason, Translated by Norman Kemp Smith*. Macmillan and Company, Ltd.

Lyons, D. 1986. A formal model of distributed computation for sensory-based robot control. Technical Report 86-43, COINS Department, University of Massachusetts Amherst.

Mugan, J., and Kuipers, B. 2007. Learning distinctions and rules in a continuous world through active exploration. In *Proceedings of 7th International Conference on Epigenetic Robotics*.

Nakamura, Y.; Hanafusa, H.; and Yoshikawa, T. 1987. Task-priority based redundancy control of robot manipulators. *Int. J. Rob. Res.* 6(2):3–15.

Nilsson, N. 1994. Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research* 139–158.

Piaget, J. 1952. *The Origins of Intelligence in Childhood*. International University Press.

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

Stoytchev, A. 2005b. Toward learning the binding affordances of objects: A behavior-grounded approach. In *Proceedings of the AAAI Spring Symposium on Developmental Robotics*.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning*. Cambridge, Massachusetts: MIT Press.