Learning Grasp Context Distinctions that Generalize

Robert Platt
Dexterous Robotics Laboratory
Johnson Space Center, NASA
robert.platt-1@nasa.gov

Roderic A. Grupen
Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts, Amherst
grupen@cs.umass.edu

Andrew H. Fagg
Symbiotic Computing Lab
School of Computer Science
University of Oklahoma
fagg@ou.edu

***Abstract*— Control-based approaches to grasp synthesis create grasping behavior by sequencing and combining control primitives. In the absence of any other structure, these approaches must evaluate a large number of feasible control sequences as a function of object shape, object pose, and task. This work explores a new approach to grasp synthesis that limits consideration to variations on a generalized localize-reach-grasp control policy. A new learning algorithm, known as *schema structured learning*, is used to learn which instantiations of the generalized policy are most likely to lead to a successful grasp in different problem contexts. Two experiments are described where Dexter, a bimanual upper torso, learns to select an appropriate grasp strategy as a function of object eccentricity and orientation. In addition, it is shown that grasp skills learned in this way can generalize to new objects. Results are presented showing that after learning how to grasp a small, representative set of objects, the robot's performance quantitatively improves for similar objects that it has not experienced before.**

I. Introduction

In the control-based approach to grasp synthesis, complex grasping behavior is represented in terms of simpler reaching and grasping primitives. For example, in order to pick up an object, a robot can execute a localize controller, followed by a reach controller and a grasp controller. Reach primitives move the manipulator to a reference pose derived from visual information or prior knowledge regarding the target object. Grasp primitives (i.e. grasp controllers) displace manipulator contacts based on tactile feedback so as to optimize the grasp [1], [2]. However, in order to be successful, the robot must select and parameterize reach and grasp controllers as a function of grasp context. This work explores an approach to autonomously learn a generalizable mapping from object shape and pose to controller parameterizations that results in a successful grasp.

One of the distinctive features of this approach is that the robot simultaneously learns a qualitative grasp strategy alongside a quantitative manipulator pose relative to the object. Many previous approaches to grasp learning in the literature have focused on one or the other of these problems. Cutkosky and Howe proposed an expert system (Grasp-Exp) that selected a robot grasp based on object and task attributes [3]. Based on observations regarding how manufacturing workers used different tools, the system was able to use tool and task criteria to select a grasp strategy from a predefined menu. In an attempt to develop an association that generalizes to new objects, Iberall used a neural network to learn the association between object and grasp type and an appropriate selection of palmer or (finger) pad opposition [4]. In the above approaches, the robot learns a qualitative grasp strategy that is not grounded in robust closed-loop controllers. In contrast, the qualitative choices that the robot learns to make in this work are actually parameterizations that directly specify how the reach or grasp controllers should behave.

Other researchers considered the problem of learning the correct manipulator or contact pose as a function of object characteristics. Moussa proposed an approach where the system learns an object-centric homogeneous transform that correctly positions the gripper based on trial-and-error experience [5]. Kamon, Flash, and Edelman described experiments where a parallel jaw gripper learns the relationship between features derived from a two-dimensional visual object outline and desired grasp points [6]. Both of these approaches to grasp learning assume that only one type of reach and one type of grasp will occur. In contrast, this paper's approach allows the robot to learn to select between multiple qualitatively different reaches and grasps.

Learning the mapping from object shape and pose to reach and grasp strategies can also be viewed as an instance of affordance learning. Gibson defined an affordance to be some aspect of the environment that an agent is able to make use of [7]. In the case of grasping, an object affords a grasp if the agent is able to pick it up. The different ways in which an object can be grasped are the object's *grasp affordances.* De Granville *et al.* showed that grasp affordances can be represented as parametric probability distributions learned from human grasp data [8]. Stoytchev proposed an approach to learning tool affordances autonomously whereby the robot discovers how different tools can be used to push an object around on a table [9]. In this work, the set of viable grasp strategies is represented by controller parameterizations and a non-parametric distribution over object-centric manipulator poses.

This paper explores a new approach to learning a mapping from object characteristics to the set of reach and grasp controllers that are likely to result in a good grasp. A controller representation known as the *control basis* (an overview is given in Section II) is used that can represent qualitatively different types of reach and grasp controllers by parameterizing reach and grasp artificial potential functions differently [10]. This paper uses *schema structured learning*, a new machine learning algorithm (discussed in Section III) introduced in a previous paper to learn grasp strategy as a function of object parameters [11]. Whereas the previous paper focused on the schema structured learning algorithm and gave only brief grasp results, Section IV of the current paper focuses more on the application of schema structured learning to grasp affordance learning. This paper gives more detail regarding how the target object is visually characterized and what kind of controllers implement the reach-grasp behavior. Section V shows that the schema structured learning approach can learn to select a qualitatively different reach strategy based on object eccentricity and a quantitatively different reference pose as a function of object orientation. In addition, it is shown that the general representation of object and grasp strategy enables the robot to improve its grasp performance measurably on objects it has no experience with, extrapolating from a relatively small set of training objects.

II. The Control Basis Approach

When using a control-based approach to solve multi-step tasks, a framework is needed that allows controllers to be sequenced in an organized way. The *control basis* framework accomplishes this by organizing the set of viable controllers and providing a robust way of evaluating system state [12].

The control basis can systematically specify an arbitrary closed-loop controller by matching an *artificial potential function* with a *sensor transform* and *effector transform* [12]. The potential function specifies controller objectives, the effector transform specifies what degrees of freedom the controller uses, and the sensor transform implements the controller feedback loop and specifies the controller reference. In the following, a controller will sometimes be identified by its artificial potential. In these cases, the artificial potential is written in small caps. For example, consider a REACH controller. The sensor transform specifies which part of the manipulator is to reach and where that part must reach to. The effector transform specifies what degrees of freedom are used to accomplish the task.

In general, the control basis realizes a complete controller by selecting one potential function from a set $\Phi = \{\phi_1, \phi_2, \ldots\}$, one sensor transform from a set $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$, and one effector transform from a set $\Upsilon = \{\tau_1, \tau_2, \ldots\}$. Given Φ, Σ, and Υ, the set of controllers that may be generated is $\Pi \subseteq \Phi \times \Sigma \times \Upsilon$. When specifying a fully-instantiated controller, the notation $\phi_i|^{\sigma}_{\tau}$ denotes the controller constructed by parameterizing potential function ϕ_i with sensor transform σ and effector transform τ. When the controller has a non-zero reference, x, the sensor will be written $\sigma(x)$.

The control basis framework allows composite controllers to be constructed that execute multiple constituent controllers concurrently. Each constituent controller is assigned a priority, and controllers with lower priority are executed in the null space of controllers with higher priority. Composite controllers are denoted, $\phi_b|^{\sigma}_{\tau} \triangleleft \phi_a|^{\sigma}_{\tau}$, where $\phi_b|^{\sigma}_{\tau}$ is said to execute "subject-to" (i.e., in the null space of) $\phi_a|^{\sigma}_{\tau}$.

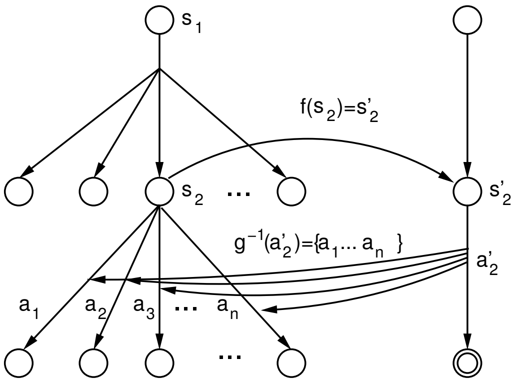

Fig. 1. Projecting the abstract policy onto the underlying state-action space: Assume that the robot is in state s_2. The state mapping, f, projects this to abstract state, s'_2. The abstract policy specifies that abstract action a'_2 is to be taken next. This inverse action mapping, g^{-1} projects a'_2 back onto the set of feasible action instantiations.

The control basis approach measures system state in terms of controller dynamics. At any point in time, the instantaneous error and the instantaneous gradient of error can be evaluated. Although the more general system dynamics can be treated [13], the current work only considers controller convergence to establish system state. Controller error is calculated by evaluating the controller's potential function ϕ for a particular sensor transform σ. Let $\mathcal{R}$ be the set of compatible potential functions and sensors: $\mathcal{R} \subseteq \Phi \times \Sigma$. System state is defined to be the elements of $\mathcal{R}$ that are converged with low error:

$$s_k = \{(\phi_i, \sigma_j) \in \mathcal{R} | \phi_i \text{ is converged for } \sigma_j \text{ with low error.}\}. \quad (1)$$

The set of all states that can be represented this way is the power set $2^{\mathcal{R}}$. For example, if the system is in state $s_k \subseteq \mathcal{R}$, then $(\phi_i, \sigma_j) \in s_k$ when ϕ_i is converged for σ_j. If $(\phi_i, \sigma_j) \notin s_k$, then ϕ_i is not converged for σ_j.

III. Schema Structured Learning

Without any structure, the control-basis approach would require a robot to search through a large space of possible controller combinations and sequences in order to determine which ones are likely to generate the desired behavior. Because this can require extensive experience and long learning times, it is frequently useful to constrain the kinds of policies that the system is allowed to consider. This can be accomplished by carefully designing the state and action space such that the resulting learning problem is tractable. Schema structured learning is a way of implicitly constraining the state and action space by restricting consideration to variations of a generalized solution, represented by an action schema [11].

An action schema is a tuple, $\mathcal{S} = \langle S', A', \pi', T' \rangle$, where S' and A' are an abstract state and action space, $\pi' : S' \rightarrow A'$ is an abstract policy, and $T' : S' \times A' \rightarrow S'$ is an abstract transition function that encodes desired transition behavior. It is assumed that the robot operates in an *underlying* Markov state and action space, S and A, but that a mapping exists between the underlying and abstract state and action spaces. The abstract policy, π', is a generalized solution, defined in the abstract space, that has many policy instantiations in the underlying space. These policy instantiations are defined in terms of state and action mappings, $f : S \rightarrow S'$ and $g : A \rightarrow A'$, that assign each underlying state and action to an abstract state and action. The set of policy instantiations is,

$$\forall s_t \in S, \;\; \pi(s_t) \in g^{-1}(\pi'(f(s_t))), \tag{2}$$

where $g^{-1}(a') = \{a \in A | g(a) = a'\}$ is the inverse of g. This is illustrated in Figure 1. Suppose that the robot is in state $s_2 \in S$. The state mapping, $f(s_2) = s'_2$, projects this state onto $s'_2 \in S'$. From this abstract state, the abstract policy takes abstract action a'_2, $\pi'(s'_2) = a'_2$. Finally, the inverse action mapping, g^{-1}, projects this abstract action onto a set of action choices, $g^{-1}(a'_2) = \{a_1, \dots, a_n\}$.

The goal of schema structured learning is to discover the policy instantiation(s) that maximizes the probability of meeting the transition constraints encoded by T'. When executing underlying action $a \in A$ from state $s_t \in S$, the next state, $s_{t+1} \in S$ must satisfy,

$$s_{t+1} \in f^{-1}(T'(f(s_t), g(a))), \tag{3}$$

where $f^{-1}(s') = \{s \in S | f(s) = s'\}$ is the inverse of f. As long as action $a \in A$ causes the robot to transition to one of these next states, the action is said to *succeed*. Otherwise, the action *fails*. If an entire sequence of actions in a policy instantiation succeeds, then the policy instantiation will be said to succeed. An *optimal* policy instantiation, π^*, is one which maximizes the probability of success. Let $P^{\pi}(a|s_t)$ be the probability of a successful policy instantiation given that the system takes action $a \in A$, starting in state $s_t \in S$, and follows policy instantiation π after that. If Π is defined to be the set of all possible policies, then

$$P^*(a|s_t) = \max_{\pi \in \Pi} P^{\pi}(a|s_t) \tag{4}$$

is the maximum probability of a successful trajectory taken over all possible policies. This allows the optimal policy to be calculated using

$$\pi^*(s_t) = \arg \max_{a \in B(s_t)} P^*(a|s_t), \tag{5}$$

where $B(s_t) = g^{-1}(\pi'(f(s_t)))$ (see Equation 2) is the set of actions that are consistent with the abstract transition function when the system is in state $s_t \in S$.

Given an action schema and the appropriate mapping, schema structured learning discovers the optimal policy instantiation online through a trial-and-error process. The algorithm gains experience by repeatedly executing policy instantiations of the action schema. While the system initially executes random instantiations of the abstract policy, performance quickly improves. Through experience, the system develops better and better approximations of the probability that a given action will succeed from a given state. The algorithm uses dynamic programming to estimate the set of optimal policy instantiations. For algorithmic details regarding schema structured learning, see [11].

In addition to structuring the solution space, an important characteristic of schema structured learning is that it can be used when the underlying state and action space is large or real-valued. A sample-based approach can be used to approximate the probability distribution of transition success because the distribution is binomial instead of multinomial, *i.e.* the algorithm is estimating $P(\text{success}|s_t, a)$ instead of $P(s_{t+1}|s_t, a)$. When the underlying state and action space is real-valued, the action schema can have a large or infinite number of policy instantiations. However, instead of maximizing over a large or infinite set of actions in Equation 5, it is possible to evaluate only elements of a finite sample set. As the algorithm gains experience, its estimate of $P^*(a|s_t)$ improves and the algorithm can re-samples the action set so that it more densely represents actions likely to succeed.

IV. The Localize-Reach-Grasp Action Schema

The LOCALIZE-REACH-GRASP action schema is used to model grasping behavior. This action schema maps onto the controllers described in this section.

A. Controllers

1) LOCALIZE*:* The LOCALIZE controller, $\phi_l|^{\sigma_l}_{\tau_l}$, visually characterizes the object to be grasped in terms of a small number of parameters. First, the object is segmented from the background in both image planes. Next, the three-dimensional Cartesian object location is determined by triangulating on the centroid of the "blob" in each image plane. Next, LOCALIZE calculates the eigenvalues and eigenvectors of the covariance matrix describing the blob in each image plane. Essentially, this step characterizes the object as an ellipsoid, as illustrated in Figure 2. By triangulating on one end of the object ellipsoid, LOCALIZE calculates the three-dimensional Cartesian position, length, orientation, and eccentricity of the object.

2) REACH*:* REACH controllers are referenced with respect to the last object detected by the LOCALIZE controller. The reach controllers are based on two artificial potentials: reach-to-position, ϕ_{pos}, and reach-to-orientation, ϕ_{rot}. ϕ_{pos} can be parameterized only by sensor and effector transforms, $\sigma_p(y, x)$ and $\tau_p(y)$, respectively. These transforms are parameterized by a set of manipulator control points, y, and a control reference offset, x. The fully instantiated position control primitive, $\phi_{pos}|^{\sigma_p(y,x)}_{\tau_p(y)}$, moves the y manipulator control points to a point along the object's major axis, at a fraction of x between the middle and one end of the major axis. The reach-to-orientation artificial potential, ϕ_{rot}, is parameterized by $\sigma_r(y, \theta)$ and $\tau_r(y)$. The fully instantiated rotation control primitive, $\phi_{rot}|^{\sigma_r(y,\theta)}_{\tau_r(y)}$, orients the y manipulator control points to an offset of θ from the object's major axis. Each contact is associated with a line from the contact frame centroid through the contact itself. $\phi_{rot}|^{\sigma_r(y,\theta)}_{\tau_r(y)}$ orients the manipulator so that the average angle between each contact's line and the object major axis (for the y set of contacts) is θ.

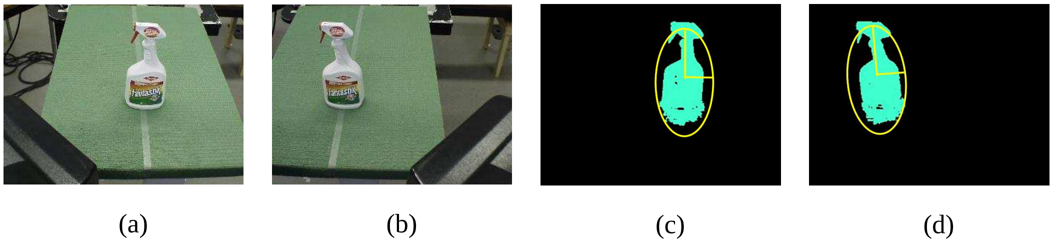

Fig. 2. The robot characterizes objects in terms of an ellipsoid fit to the segmented object. (a) and (b) illustrate the left and right camera views of a squirt bottle. (c) and (d) illustrate the corresponding segmented "blobs" and their ellipsoids.

Recall that the control basis allows control primitives to be combined using the subject-to relation, $\triangleleft$, to create composite controllers. The current work only allows two combinations: $\phi_{pos}|^{\sigma_p(y,x)}_{\tau_p(y)}$ and $\phi_{rot}|^{\sigma_r(y,\theta)}_{\tau_r(y)} \triangleleft \phi_{pos}|^{\sigma_p(y,x)}_{\tau_p(y)}$. When $\phi_{pos}|^{\sigma_p(y,x)}_{\tau_p(y)}$ executes alone, the manipulator reaches toward a position without controlling for orientation. When $\phi_{rot}|^{\sigma_r(y,\theta)}_{\tau_r(y)} \triangleleft \phi_{pos}|^{\sigma_p(y,x)}_{\tau_p(y)}$ executes, the manipulator reaches toward both the specified position and rotation offsets.

3) GRASP: GRASP controllers displace contacts toward good grasp configurations using feedback control [1], [2]. This approach uses tactile feedback to calculate an error gradient and displace grasp contacts on the object surface without a geometric object model. After making light contact with the object using sensitive tactile load cells, the controller displaces contacts toward minima in the grasp error function using discrete probes [1] or a continuous sliding motion [14].

Grasp controllers descend an artificial potential, ϕ_g, derived from wrench error,

$$\epsilon_w = \vec{\rho}^T \vec{\rho}, \quad \vec{\rho} = \sum_{1 \le i \le n} \vec{w}_i, \tag{6}$$

where $\vec{w}_i$ is the contact wrench applied by the i^{th} contact, assuming no surface friction. $\vec{w}_i$ is calculated directly from tactile feedback by using the approach of Bicci, *et al.*, to estimate contact location [15]. The control law converges when the contacts have been displaced to locations where the net applied wrench is minimized. If the minimum corresponds to zero net wrench, then, in the presence of friction, such a grasp achieves wrench closure because it fulfills the conditions for non-marginal equilibrium. Non-marginal equilibrium requires the contact forces achieving net zero force lie strictly inside their corresponding friction cones and has been shown to be a sufficient condition for wrench closure [16].

Two GRASP controllers are used: $\phi_g|^{\sigma_g(123)}_{\tau_g(123)}$ and $\phi_g|^{\sigma_g(12)}_{\tau_g(12)}$. $\phi_g|^{\sigma_g(123)}_{\tau_g(123)}$ uses three physical contacts to synthesize a grasp, while $\phi_g|^{\sigma_g(12)}_{\tau_g(12)}$ combines two physical contacts (out of three) into a *virtual finger* [17] that is considered to apply a single force that opposes a third physical contact.

B. LOCALIZE-REACH-GRASP

The LOCALIZE-REACH-GRASP action schema basically represents the class of policies where a LOCALIZE action is followed by a REACH and a GRASP action. It is defined over a set of four abstract states, $S' = \{(000), (001), (011), (111)\}$, and three abstract actions, $A' = \{\phi_l, \phi_{pos}, \phi_g\}$. The four abstract states represent the possible combinations of controller convergence by representing the artificial potentials, ϕ_l, ϕ_{pos}, and ϕ_g as bits (see Table I).

Artificial Potential	Bit
ϕ_l	001
ϕ_{rp}	010
ϕ_g	100

TABLE I

ARTIFICIAL POTENTIALS AND THEIR CORRESPONDING STATE BITS. STATES ARE REPRESENTED AS BIT STRINGS WHERE A BIT IS SET TO 1 WHEN THE CORRESPONDING ARTIFICIAL POTENTIAL IS IN THE STATE.

The underlying state and action space is defined as described in Section II. The state space is defined in terms of the artificial potentials,

$$\Phi_{lrg} = \{\phi_l, \phi_{pos}, \phi_g\}, \tag{7}$$

and the set of sensor transforms,

$$\begin{aligned} \Sigma_{lrg} = \; & \{\sigma_l, \\ & \{\sigma_p(y,x) | y \subseteq \{1,2,3\}, x \in [0,1]\}, \\ & \{\sigma_r(y,\theta) | y \subseteq \{1,2,3\}, \theta \in [0,\pi/2]\}, \\ & \{\sigma_g(y) | y \subseteq \{1,2,3\}\}\}, \end{aligned} \tag{8}$$

as described in Section IV-A. The set of underlying states, S is a subset of $2^{\mathcal{R}}$ (see Equation 1), where $\mathcal{R} \subseteq \Phi_{lrg} \times \Sigma_{lrg}$ is the set of compatible artificial potentials and sensory transforms.

The above abstract and underlying state and action spaces suggest the following g and f function: $g((\pi_k, \ldots, \pi_1)) = h(\pi_1)$, where $h((\phi_i, \sigma_i, \tau_i)) = \phi$, and $f(s) = \{\phi \in \Phi | \exists \sigma \in \Sigma \text{ s.t. } (\phi, \sigma) \in s\}$. g essentially looks only at ϕ_1 and "strips" it of its sensor and effector transforms to leave only a potential function. In a similar way, f maps onto abstract states by "stripping" sensor transforms from the converged pairs of artificial potentials and sensor transforms in the underlying state.

The LOCALIZE-REACH-GRASP action schema also defines a policy and transition function over the abstract state and

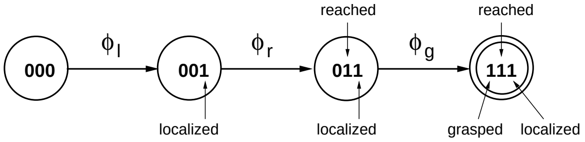

Fig. 3. The localize-reach-grasp action schema. The circles with binary numbers in them represent abstract states. The arrows represent abstract actions and possible transitions.

action space, as illustrated in Figure 3. The policy is

$$\begin{aligned} \pi'_{lrg}(000) &= \phi_l \\ \pi'_{lrg}(001) &= \phi_{rp} \\ \pi'_{lrg}(011) &= \phi_g. \end{aligned} \tag{9}$$

and the transition function is

$$\begin{aligned} T'(000, \phi_l) &= 001 \\ T'(001, \phi_{rp}) &= 011 \\ T'(011, \phi_g) &= 111. \end{aligned} \tag{10}$$

V. Experiments

This paper characterizes the ability of schema structured learning to make appropriate grasp distinctions in two experiments that show that schema structured learning can learn to select different grasp strategies as a function of object eccentricity and orientation. This paper also shows that these contextual distinctions generalize from a small set of training objects to a much larger set of test objects. All experiments were performed using Dexter, the UMass bi-manual humanoid robot [18]. Dexter consists of a 4-degree-of-freedom (DOF) bisight head and two Barrett Technologies whole-arm manipulators (WAMs). Each Barrett WAM is equipped with a 3-finger, 4-DOF Barrett Hand. Mounted on the tip of each Barrett hand finger is a 6-axis force-torque sensor.

A. Conditioning on object eccentricity, orientation, and length

1) Eccentricity: Recall from Section IV that Dexter can choose to reach to a position without specifying orientation or it can specify both position and orientation. This experiment demonstrates that schema structured learning can discover when each of these two reach strategies is appropriate. In this experiment, Dexter alternately reached toward a vertically presented towel roll (10cm diameter and 20cm high) or a round ball (16.5cm diameter) 42 times. At the beginning of each grasp trial, the object was placed in approximately the same tabletop location. Then, schema structured learning executed (using the LOCALIZE-REACH-GRASP action schema) until either the absorbing state was reached or an action failed. When either of these events occurred, the system was reset and a new trial was started. For this experiment, Dexter was limited to executing only three-fingered grasps. In addition, the algorithm only considered object eccentricity, ignoring size, orientation, and position.

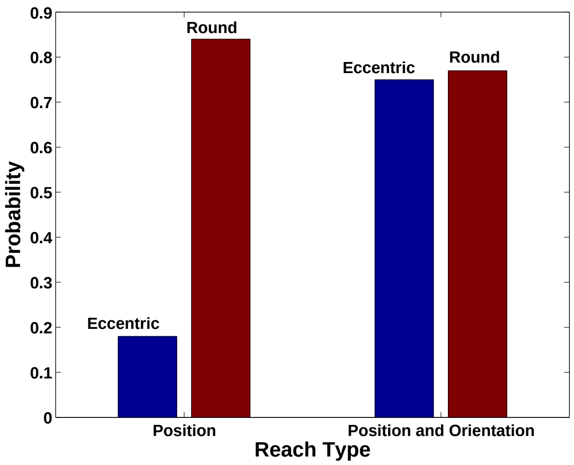

Fig. 4. Conditioning on eccentricity: the four bars in this graph plot the maximum estimated probability of grasp success (for round and eccentric objects) when reaching to both a position and orientation, and when reaching to a position without specifying orientation. When attempting to grasp the round object, the algorithm learns that a reach that specifies either position and orientation or position alone will work. However, when attempting to grasp an eccentric object, the system learns that orientation is important.

The results given in Figure 4 show the maximum probability of successfully grasping the round object (the ball) and the eccentric object (the vertical towel roll) using both reach types. The two bars labeled "Position and Orientation" are the maximum probabilities of a successful grasp when the REACH controller specifies both position and orientation. The two bars labeled "Position" are the maximum probability of a successful grasp given a REACH where only position offset was specified. Notice that for the eccentric object, a much higher probability of success can be achieved when both position and orientation offset are specified. In contrast, for the round object, it is possible to achieve high success rates using either type of REACH controller.

2) Orientation: The second experiment, Dexter learned to condition its choice of reach controller position reference based on object orientation. This experiment used LOCALIZE-REACH-GRASP-HOLD-LIFT, an augmented version of the LOCALIZE-REACH-GRASP action schema. After reaching and grasping the object, LOCALIZE-REACH-GRASP-HOLD-LIFT also holds the object and lifts it. In this experiment, a cracker box (measuring 5x5x10cm with a mass of 280g) was alternately presented to Dexter horizontally and vertically. After lifting the box, the hold action was only considered to have succeeded if all manipulator contacts remained in contact and the object did not exert a large moment on the manipulator, i.e. the object was grasped near its center of mass (COM). After 60 trials, SCHEMA STRUCTURED LEARNING had learned to use different grasp strategies based on vertical elevation of the object. Figure 5(a) shows that when the box was presented vertically, the probability of success was maximized when the manipulator was oriented perpendicular to the object major axis. However, note that the position of the manipulator along the major axis did not matter. Figure 5(b) shows that when the

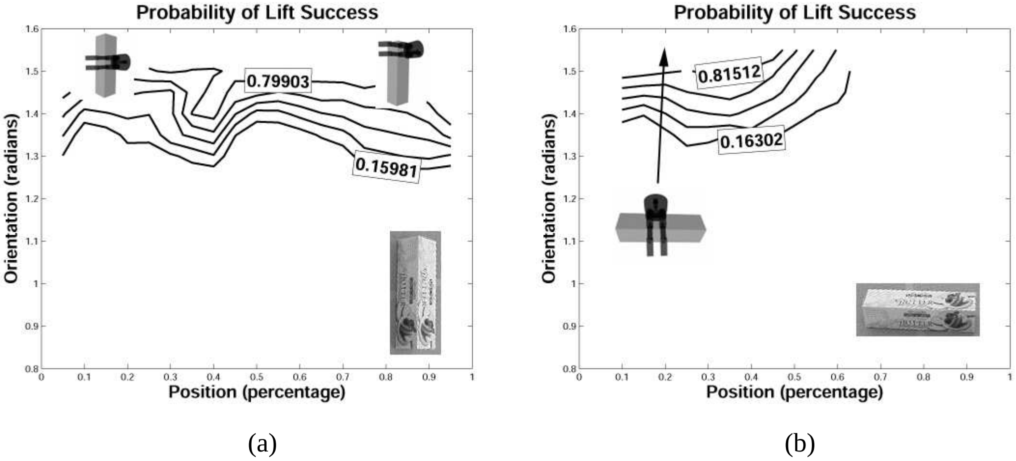

Fig. 5. The results of learning to lift the cracker box when it is presented vertically, (a), versus when it is presented horizontally, (b). These contour plots illustrate the probability of a successful lift as a function of REACH position and orientation. Dexter learns that, in both poses, the manipulator should be oriented perpendicular to the object (in both plots, probability is maximized near the top of the graph.) However, Dexter also learns that if the object is presented horizontally, it must be grasped near its center of mass (in (b), the probability is maximized on the left of the graph.) If the horizontal object is grasped near one end, the object may twist out of the grasp and drop.

box was presented horizontally, it was necessary to grasp the box near its center as well as to use the correct orientation. This difference is due to the effect of the object COM on the success of lifting. When the box is horizontal, the COM can exert a large moment that can cause the object to drop if it is grasped from from its center. However, this is not a problem when the box is presented vertically.

B. Generalization to New Objects: Bagging Groceries

Although the previous experiments show that SCHEMA STRUCTURED LEARNING can discover shape- and pose-appropriate reach-grasp strategies, it is not yet clear whether these grasp skills generalize to new objects. In this experiment, Dexter learns to grasp a set of five training objects by distinguishing them based on object length, eccentricity, and orientation (see Section IV-A.1). The grasping skills learned from the five objects was evaluated by attempting to grasp a much larger set of 19 test objects that Dexter had not previously experienced.

The system was trained using the five objects shown in Figure 6. The butter cracker box (Figure 6(e)) was always presented horizontally. For each of the five training objects, SCHEMA STRUCTURED LEARNING learned to grasp and lift it over the course of approximately 60 trials. Dexter was constrained only to grasp with two virtual fingers, $\phi_g|^{\sigma_{12}}_{\tau_{12}}$.

The LOCALIZE-REACH-GRASP-HOLD-LIFT skills learned in the context of the five training objects were tested on the 19 different test objects shown in Figure 7. For each test object, the LOCALIZE-REACH-GRASP-HOLD-LIFT action schema was executed 16 times: eight times without using the experience acquired from the test objects and eight times with this experience. During the eight executions that tested performance without experience, SCHEMA STRUCTURED LEARNING essentially selected random instantiations of the action schema. During the eight executions that did use the training data, the algorithm effectively interpolated (in the space of the four visual features) the action schema instantiation from among the neighboring training objects.

Figure 8 illustrates the results. In both graphs, the horizontal axis corresponds to the object number in Figure 7. Figure 8(a) shows the grasp error after REACH controller and before GRASP controller execution. A low grasp error indicates that the manipulator is close to a good grasp configuration. The dashed line in Figure 8(a) shows the mean initial grasp error for the eight LOCALIZE-REACH-GRASP-HOLD-LIFT trials that did not benefit from the skills learned on the training set. The solid line shows the mean initial error for the eight trials that did use the training data. Although it is not universally true, this graph shows that the average initial grasp error for many of the 19 test objects was lower when SCHEMA STRUCTURED LEARNING used previous training experience than when it did not. For 14 out of the 19 test objects, the performance without training was worse than one standard deviation away from the mean performance with training.

Figure 8(b) suggests a similar conclusion. This graph analyzes the value of training experience in terms of the probability of successfully holding and lifting the test object. A hold is considered successful only if all of the grasping contacts continue to apply the reference hold force and the contacts do not apply a large moment on the object. This is only true when a good grasp has been established close to the object center of mass. Figure 8(b) shows that, averaged over all 19 objects, the probability of successfully grasping and lifting the object without using training experience is around 50%. However, this probability rises significantly when the training experience is used. In fact, the figure shows that the probability of successfully grasping and lifting the object almost always rises (except in one case) when the algorithm is allowed to use training experience.

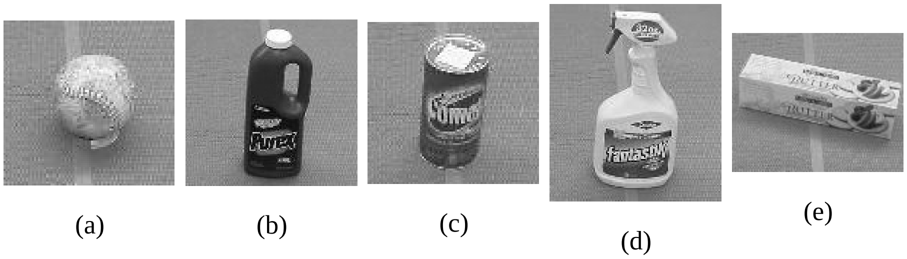

Fig. 6. The five training objects used in the grocery bagging experiment.

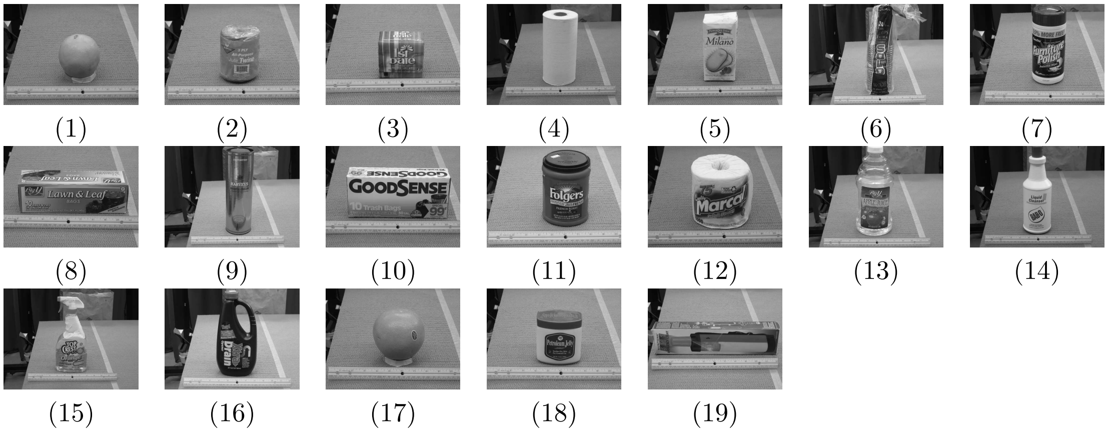

Fig. 7. The 19 test objects used in the grocery bagging experiment.

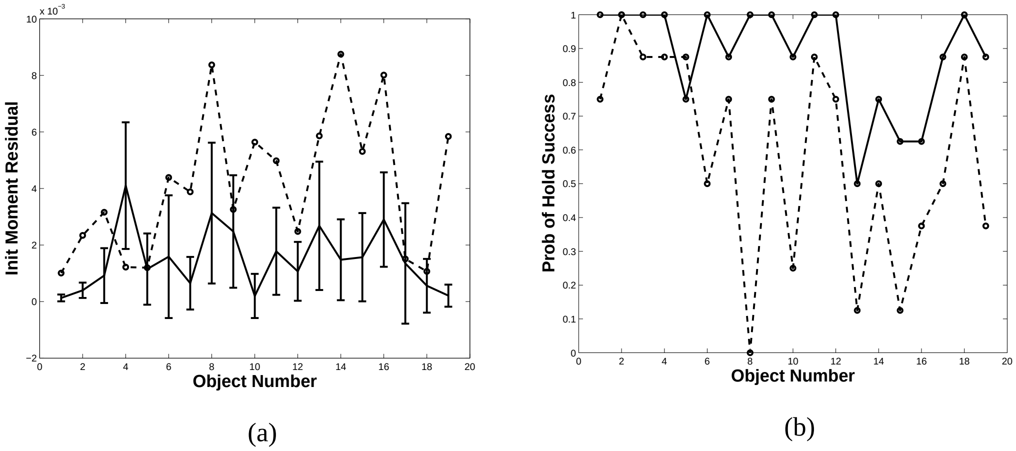

Fig. 8. Performance of grasping the 19 test objects with (the solid line) and without (the dashed line) previous experience grasping the five training objects. In both plots, the horizontal axis represents the object number from Figure 7. (a) plots the mean grasp error after executing the REACH and before executing the GRASP for the 19 test objects. The error bars plot one standard deviation above and below the mean. (b) plots the mean probability of successfully lifting the object.

This experiment demonstrates that it is possible to learn general reach-grasp skills based on experience with a limited set of objects and apply these skills to new objects. Although the experimenter selected the 19 test objects, they are a representative sampling of a large class of objects that can be found in most grocery stores.

VI. Conclusion

This paper takes a control-based approach to grasp synthesis, whereby the problem is recast as that of correctly sequencing and combining reach and grasp controllers. Instead of considering every possible sequence of controllers, consideration is limited to those sequences that match a generalized grasp strategy, encoded as an action schema. Using schema structured learning, the robot autonomously learns how to instantiate the generalized policy as a function of grasp context. This approach is experimentally demonstrated to be capable of learning suitable grasp strategies as a function of object eccentricity and orientation. In addition, it is shown that once grasp strategies have been learned for a representative set of objects, these strategies can improve grasp performance other objects, even when those objects have not been experienced before.

ACKNOWLEDGMENT

The work described in this paper was completed while the first author was a student at the University of Massachusetts. The authors would like to thank Emily Horrell for her help maintaining Dexter and running experiments. This work was supported by NASA grant NNJ05HB61A, ARO grant DAAD 19-03-R-0017, and NASA GSRP Fellowship NNJ04jf76H.

REFERENCES

[1] J. Coelho and R. Grupen, "A control basis for learning multifingered grasps," *Journal of Robotic Systems*, 1997.

[2] R. Platt, A. H. Fagg, and R. A. Grupen, "Nullspace composition of control laws for grasping," in *IEEE Int'l Conf. on Intelligent Robots and Systems, Lausanne, Switzerland*, September 2002.

[3] M. Cutkosky and R. Howe, *Dextrous Robot Hands*. NY: Springer-Verlag, 1990, ch. Human grasp choice and robotic grasp analysis, pp. 5–31.

[4] C. MacKenzie and T. Iberall, *The Grasping Hand*. North-Holland, 1994.

[5] M. Moussa, "Combining expert neural networks using reinforcement feedback for learning primitive grasping behaviour," *IEEE Transaction on Neural Networks*, 2003.

[6] I. Kamon, T. Flash, and S. Edelman, "Learning to grasp using visual information," Mathematics and Computer Science, Weizmann Institue Of Science, Tech. Rep. CS94-04, 1994.

[7] J. Gibson, "The theory of affordances," in *Perceiving, Acting, and Knowing*, R. E. Shaw and J. Bransford, Eds. Lawrence Erlbaum, 1977.

[8] C. de Granville, J. Southerland, and A. Fagg, "Learning grasp affordances through human demonstration," in *Proceedings of the Int'l Conf. on Development and Learning*, 2006.

[9] A. Stoytchev, "Toward learning the binding affordances of objects: a behavior-grounded approach," in *Proceedings of the AAAI Spring Symposium on Developmental Robotics*, 2005.

[10] M. Huber and R. Grupen, "A hybrid discrete event dynamic systems approach to robot control," University of Massachusetts, Tech. Rep., 1996.

[11] R. Platt, R. Grupen, and A. Fagg, "Improving grasp skills using schema structured learning," in *Proceedings of the Int'l Conf. on Development and Learning*, 2006.

[12] M. Huber, "A hybrid architecture for adaptive robot control," Ph.D. dissertation, U. Massachusetts, 2000.

[13] J. Coelho, J. Piater, and R. Grupen, "Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot," *Robotics and Autonomous Systems Journal, special issue on Humanoid Robots*, vol. 37, no. 2-3, November 2001.

[14] R. Platt, A. H. Fagg, and R. A. Grupen, "Manipulation gaits: Sequences of grasp control tasks," in *IEEE Int'l Conf. Robotics Automation, New Orleans, Louisiana*, April 2004.

[15] A. Bicchi, J. Salisbury, and D. Brock, "Contact sensing from force measurements," *International Journal of Robotics Research*, vol. 12, no. 3, 1993.

[16] J. Ponce, S. Sullivan, A. Sudsang, J. Boissonnat, and J. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects," *Int. J. Rob. Res.*, 1996.

[17] R. Platt, A. Fagg, and R. Grupen, "Extending fingertip grasping to whole body grasping," in *IEEE Int'l Conference on Robotics and Automation, Taipei, Taiwan*, May 2003.

[18] R. Platt, O. Brock, A. H. Fagg, D. Karupiah, M. Rosenstein, J. Coelho, M. Huber, J. Piater, D. Wheeler, and R. Grupen, "A framework for humanoid control and intelligence," in *Proceedings of the 2003 IEEE International Conference on Humanoid Robots, Munich, Germany*, October 2003.