Learning Grasp Context Distinctions that Generalize

Robert Platt
Dextrous 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 paper 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. Experiments are described where Dexter, a dexterous bimanual humanoid, learns to select appropriate grasp strategies for different objects as a function of object eccentricity and orientation. In addition, it is shown that grasp skills learned in this way generalize well 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.**

Fig. 1. Dexter grasping a cylindrical object.

I. Introduction

In the control-based approach to grasp synthesis, complex grasping behavior is represented in terms of parameterizable reaching and grasping control primitives. For example, in order to pick up an object, a robot can execute a reach controller followed by a grasp controller. Reach primitives move the manipulator to an offset from a visually-determined object pose. Grasp primitives (i.e. grasp controllers) displace manipulator contacts based on tactile feedback so as to optimize the grasp [1], [2], [3]. In order to grasp successfully, the reach controller must be parameterized with the appropriate goal pose and the grasp controller must be parameterized with an appropriate grasp type. Although it is possible for the system designer to hard-code parameter choices for known special cases, a robot that learned appropriate controller parameterizations that generalized to new situations would be more flexible and require less programming. This paper explores autonomously learning context-appropriate parameterizations of reach and grasp controllers for grasp tasks where the specific objects to be grasped are not known *a priori.* The robot learns appropriate reach and grasp controller parameterizations as a function of coarse visual features including object pose and eccentricity. The quality of a grasp resulting from a particular parameter choice is evaluated based on the grasp controller error function, calculated using local tactile feedback. Essentially, the process of learning an association between coarse visual features and appropriate parameters is "supervised" by tactile feedback.

In most research that explores grasp learning, the robot learns a relationship between visual or object features and the precise positions where grasp contacts must be placed. For example, Kamon, Flash, and Edelman described experiments where a robot with a parallel jaw gripper learned the relationship between features derived from a two-dimensional visual object outline and desired grasp points [4]. Saxena *et al.* learned sets of visual edge and texture features that predicted points on the object where a parallel jaw gripper may be centered so as to grasp an object [5]. Moussa proposed learning an object-centric homogeneous transform that correctly positions the gripper based on object characteristics that he calls *g-features* [6].

Rather than learning precise goal locations for grasp contacts, the work reported in this paper learns approximate grasp configurations as a function of coarse visual features. This approach relies on a grasp controller that uses tactile feedback to displace grasp contacts into precise positions [1], [2], [3]. The learning system delivers the manipulator to a "neighborhood" of good grasp configurations and the grasp controller finds a nearby precise contact configuration. The advantages of this approach are as follows. First, it decomposes the grasp synthesis problem into an initial reach to a grasp region based on qualitative visual feedback and a subsequent quantitative refinement of contact positions based on tactile feedback. This allows the vision subsystem to be focused on detecting those features relevant to grasp strategy rather than on precise positioning of the contacts. This more focused role for vision reduces the need to place the camera nearby the grasping interaction or to carefully align the cameras with the interaction. Another advantage of augmenting visual information with tactile feedback is that the tactile information can be used to provide feedback to the visual learning process. The grasp controller error function reports the exact grasp quality in terms of applied forces, rather than relying on less direct or quantitative methods of assessing grasp quality.

The focus of the current work is on learning the relationship between coarse visual features and context-appropriate controller parameterizations. A new learning algorithm, known as *schema structured learning*, is used to learn this relationship. Section II describes the reach and grasp controllers used in the current work and reviews schema structured learning. Next, Section III-A, characterizes this approach in a series of experiments where, Dexter, the UMass bimanual dexterous humanoid, learns to grasp a vertically-presented cylinder, a sphere, a vertical detergent bottle, and a horizontally-presented rectangular box. Finally, in Section III-B, the ability for grasp skills learned in this way to generalize to new objects is explored in an experiment where Dexter learns to grasp a set of training objects and tests its knowledge on a set of unfamiliar test objects

II. Learning Grasp Strategies

In this work, the robot learns which parameterizations of reach and grasp controllers maximize the probability of a successful grasp as a function of grasp context. First, this section describes the set of viable parameterizations of reach and grasp controllers. Next, based on general visual characteristics of the object, a new learning algorithm, *schema structured learning*, is used to determine which reach-grasp parameterization is most likely to lead to grasp success.

A. Reach and Grasp Controllers

1) Reach*:* The goal of the reach controller is to move the grasp contacts to a configuration from which the grasp controller will be able to realize a good grasp by making only small displacements. It is referenced with respect to the visually determined object pose and is decomposed into two component primitives: a reach-to-position control primitive, π_{rx}, and a reach-to-orientation control primitive, $\pi_{r\theta}$. When position control primitive, π_{rx}, operates alone, then the manipulator moves toward a designated position while leaving orientation unspecified. When these two primitives are concurrently combined using the subject-to operator, $\pi_{r\theta} \triangleleft \pi_{rx}$, the resulting controller moves a set of manipulator contact points to a designated position and orientation relative to the object. (For more information on the subject-to operator, see [7], [3].)

The reach control primitives are parameterized by a set of manipulator contact points, γ_y, and position and orientation offsets, κ_x or κ_θ, relative to the visual centroid and orientation of the major axis. The γ_y contact points are moved to a configuration with the specified position and orientation offsets. The reach-to-position control primitive, $\pi_{rx}|_{\gamma_y}^{\gamma_y}(\kappa_x)$, moves the centroid of the contacts in γ_y to a position κ_x along the object major axis. Let $\sigma_{\mathbf{x}} \in R^3$ be the Cartesian object centroid and let $\sigma_{m1} \in R^3$ be a vector pointing along the object major axis from the object centroid to the end of the object major axis. The reach-to-position control primitive moves the centroid of the γ_y contact points to the reference position,

$$x_{ref} = \sigma_{\mathbf{x}} \pm \kappa_x \sigma_{m1},$$

where $\kappa_x \in [0, 1]$ is the position of x_{ref} as a fraction of the total length of the object major axis. The reach-to-orientation control primitive, $\pi_{r\theta}|_{\gamma_y}^{\gamma_y}(\kappa_\theta)$, orients the set of contacts in γ_y with respect to the object major axis. This primitive is only defined for two or three contact points, $|\gamma_y| \in \{2, 3\}$. If $|\gamma_y| = 2$, then $\pi_{r\theta}$ orients the two contacts such that a line that passes through both contacts has the specified offset angle, $\kappa_\theta \in [0, \frac{\pi}{2}]$, with the object major axis. If $|\gamma_y| = 3$, then $\pi_{r\theta}$ orients the three contacts such that the normal of the plane formed by the three contacts forms the specified offset angle with the object major axis.

Sets of reach controller parameterizations are defined as follows. Π_{rx} is defined to be the set of allowed parameterizations of the reach-to-position controller,

$$\Pi_{rx} = \{\pi_{rx}|_{\gamma_y}^{\gamma_y}(\kappa_x) : \gamma_y \subseteq \Gamma, \kappa_x \in [0, 1]\}$$

where Γ is a set of allowed contact resources. $\Pi_{rx\theta}$ is the set of allowed parameterizations of the composite controller that specifies both position and orientation,

$$\Pi_{rx\theta} = \{\pi_{r\theta}|_{\gamma_y}^{\gamma_y}(\kappa_\theta) \triangleleft \pi_{rx}|_{\gamma_y}^{\gamma_y}(\kappa_x) : \gamma_y \subseteq \Gamma, \kappa_\theta \in \left[0, \frac{\pi}{2}\right], \kappa_x \in [0, 1]\}.$$

2) Grasp*:* Grasp controllers displace contacts toward good grasp configurations using feedback control [1], [2], [3]. 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 [3].

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

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

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 [8]. 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 [9]. The grasp controller, $\phi_g|^{\sigma_g(\gamma_y)}_{\tau_g(\gamma_y)}$, is parameterized by a set of contact resources, γ_y, that are used to synthesize a grasp. The set of feasible grasp controller parameterizations is,

$$\Pi_g = \{\phi_g|^{\sigma_g(\gamma_y)}_{\tau_g(\gamma_y)} : \gamma_y \subseteq \Gamma\},$$

where γ_y may include either physical contacts or *virtual* contacts. In the case of a virtual contact, multiple physical contacts are considered, for the purposes of grasp synthesis, to occupy a single net position and apply a single net force.

B. Application of Schema Structured Learning to Grasp Synthesis

This work applies schema structured learning to the problem of determining which reach and grasp controller parameterizations maximize the probability of grasp success as a function of a coarse visual approximation of the object. Schema structured learning takes a description of a generalized solution, known as an *action schema*, as input. The action schema is a generalized solution that can be instantiated in different ways. Schema structured learning executes various different instantiations and estimates the probability that they will meet action schema goals as a function of state and problem context. Schema structured learning focuses exploration on those instantiations estimated to have the greatest probability of satisfying action schema goals. For a more detailed description of schema structured learning, see [10], [3].

In its application to grasp synthesis, coarse grasp context is established by general visual features. The robot learns which reach and grasp controller parameterizations minimize expected grasp error and maximize the probability of lift success. At the start of learning, the robot executes random parameterizations of the reach and grasp controllers and observes the results. After each grasp, the outcome of executing each controller for the particular visual features is stored. On subsequent trials, this experience is used to calculate the expected grasp error and probability of successfully lifting the object for various different reach-grasp parameter choices. As learning progresses, more and more experience accrues and the ability of the robot to select high quality grasp configurations improves.

1) Expected Grasp Quality as a Function of Coarse Visual Parameters: Before reaching, the object is visually characterized in terms of coarse visual features. In the current work, these features are the parameters of the ellipsoid that most closely matches the visual "blob" that corresponds to the object. In order to calculate the ellipsoid parameters, the object is first 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, the eigenvalues and eigenvectors of the covariance matrix describing the blob in each image plane are calculated. Essentially, this step characterizes the object as an ellipsoid, as illustrated in Figure 2. Finally, the spatial Cartesian location of the centroid, $\sigma_{\mathbf{x}}$, and vectors pointing along the object major and minor axes (σ_{m1} and σ_{m2}, respectively) are calculated.

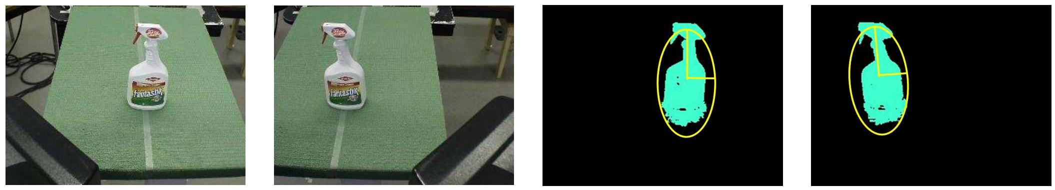
These vectors are used to calculate the parameters upon which estimates of expected grasp error and the probability of lift success are conditioned. These parameters include blob position, $\sigma_{\mathbf{x}} \in R^3$, major axis length, $\sigma_l \in R$, major axis elevation angle, $\sigma_\phi \in \left[0, \frac{\pi}{2}\right]$, and eccentricity, $\sigma_e \in R$. These parameters encode the *context* of the grasp synthesis problem and are used in predicting the reach controller parameters, κ_x, κ_θ, and the set of grasp contacts, γ_y, that minimize grasp error and maximize the probability of lift success.

The expected grasp error and probability of subsequent lift success as a function of context and controller parameters can be approximated in a variety of ways. If context and controller parameters are discrete quantities, then these probabilities may be approximated by a multinomial distribution. If they are continuous quantities, then several approximation methods may be used. If the distribution is assumed to be known *a priori*, then a parametric method may be used. Otherwise, a non-parametric, lazy-learning approach may be more appropriate. This paper's experiments used k-nearest neighbor to approximate the expected error and the probability of transport success as a function of problem context.

2) Sampling the Parameter Space: After visually characterizing the ellipsoid parameters that describe the object, schema structured learning evaluates the expected grasp error and probability of transport success for a set of different reach-grasp parameter choices. In the case of a small and discrete parameter space, it is possible to evaluate each reach-grasp candidate individually. However, note that the reach controller parameters, $\kappa_{\mathbf{x}}$ and κ_θ, constitute a real-valued parameter space that must be sampled. Three possible sample strategies are: sampling from a regularly-spaced grid, sampling randomly from a uniform distribution over the parameter space, and sampling randomly from the estimated distribution. In sampling from the estimated distribution, the quantity to be evaluated (for example, grasp error) is converted into a probability distribution where the most valuable regions of parameter space are given the highest probabilities. By sampling from this distribution, regions of the parameter space estimated to have the highest value are the most densely sampled. At the start of learning, the robot has no relevant experience and the parameter space is sampled uniformly randomly (assuming a uniform prior). As experience accrues, the distribution corresponding to the quantity being estimated improves and the sample strategy more densely covers high-value regions of the parameter space. Eventually, the bulk of the sample set becomes focused on high-value peaks in the parameter space.

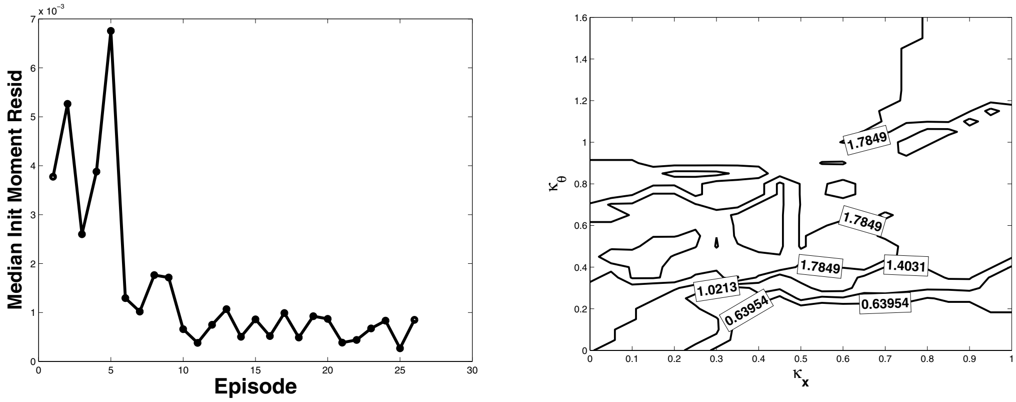
III. Experiments

Two experiments were conducted to evaluate this approach to grasp learning. The first experiment evaluated the approach on a set of four known objects for which a good ellipsoid fit exists: a vertically presented cylinder, a 16cm diameter sphere, a vertical detergent bottle, and a horizontal rectangular



(a) (b) (c) (d)

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.



(a) (b)

Fig. 3. Dexter learned to grasp a vertical cylinder using a three-contact grasp. (a) shows the median (over eight learning runs) moment residual error (grasp error) after the reach executed but before the grasp controller as a function of grasp number. (b) is a contour graph that shows the expected grasp error as a function of the reach controller parameters, κ_x and κ_θ.

object. Results characterize the average speed of learning and the grasp skills that are learned as a result for each of the four objects. The second experiment evaluated how well grasp strategies learned for particular objects generalized to new objects. In this experiment, the system trained on a set of five objects and tested on 19 new objects. The results show that the grasp knowledge learned using the training objects measurably improves performance on the new test objects when compared with the performance of random reach-grasp parameterizations.

All experiments were performed using Dexter, the UMass bi-manual humanoid robot [11]. 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. Experiment 1: Learning to Grasp Different Objects

1) Vertical Cylinder: Dexter learned to grasp a vertically presented cylinder 10cm in diameter and 20cm high in a series of reaches and grasps. On each reach, Dexter selected a reach controller parameterization from the set $\Pi_{rx} \cup \Pi_{rx\theta}$. If the selected reach controller left orientation unspecified (*i.e.* a member of Π_{rx} was selected), then only the parameter $\kappa_x \in [0,1]$ was specified. If a reach controller was an element of $\Pi_{rx\theta}$, then both $\kappa_x \in [0,1]$ and $\kappa_\theta \in \left[0, \frac{\pi}{2}\right]$ were specified. After reaching to the object, Dexter executed the grasp controller. Each trial terminated after the grasp controller converged to a good grasp configuration or was prematurely terminated by the human monitor to prevent collisions with the table or object.

Figure 3(a) shows median moment residual grasp error after executing the reach controller, but before executing the grasp controller, for eight learning runs as a function of grasp number. In this experiment, Dexter was constrained to attempt to grasp using three fingers. The drop in median moment residual error as a function of grasp number indicates that this approach is able to learn to select reach controller parameterizations that lead to low grasp errors quickly.

Figure 3(b) is a contour graph that shows expected composite grasp error as a function of κ_x and κ_θ (the parameters of the $\pi_{r\theta} \triangleleft \pi_{rx}$ reach controller). Recall that κ_x is a proportional distance along the object major axis between the center of the major axis and either end of the major axis. κ_θ is the angle between the object major axis and the normal of the plane that passes through the three contacts. This graph shows the grasp error that the system learned (over the course of 20 attempted reaches and grasps) to expect as a function of the pose that Dexter reached toward. In order to create the contour plot of

Figure 3(b), the two components of grasp error, force residual error, ϵ_{fr}, and moment residual error, ϵ_{mr}, were combined using the following weighted sum:

$$\epsilon_g = \epsilon_{fr} + 150\epsilon_{mr}. \quad (2)$$

Figure 3(b) shows that ϵ_g was minimized when Dexter reached to a configuration where the normal of the plane containing the contacts is approximately parallel to the object major axis (near $\kappa_\theta = 0$). In this graph, executions of the grasp controller that failed to converge were assigned error values of $\epsilon_{fr} = 4$ and $\epsilon_{mr} = 0.05$.

Although the contour graph of grasp error shown in Figure 3(b) is learned over relatively few reaches and grasps, it is similar to the true function. Figure 4(a) shows the weighted sum of grasp error for three-contact grasps as a function of κ_x and κ_θ after five times as much experience (106 grasp experiences). Note that, grasp error is still minimized when the normal of the plane of defined by the three contacts is nearly parallel with the object major axis.

2) 16cm Diameter Sphere: The same learning process was tested on a 16cm diameter sphere. In a series of 60 reaches and grasps, Dexter executed reach controller parameterizations drawn from the set $\Pi_{rx} \cup \Pi_{r\theta x}$, followed by an execution of the three-contact grasp controller. In this implementation, the sphere was perceived to have a longer vertical extent than actually existed because shadows cast by the object were perceived by the vision subsystem to be part of the object itself. This effect caused the vision system to consistently perceive the spherical object to have a short, vertically directed major axis.

Although the sphere was consistently perceived to be vertically oriented, Dexter learned that manipulator orientation relative to the object had little effect on expected grasp error. Figure 4(b) is a contour plot showing the expected grasp errors as a function of κ_x and κ_θ. It shows that low-error grasps exist for the 16cm sphere at a large range of orientations – between approximately $\frac{\pi}{4}$ and 0 radians. This result contrasts with the contour plot from the vertical cylinder in Figure 4(a) where low-error grasps exist only in configurations where the normal of the plane of the contacts is nearly parallel to the major axis, *i.e.* when κ_θ is close to zero.

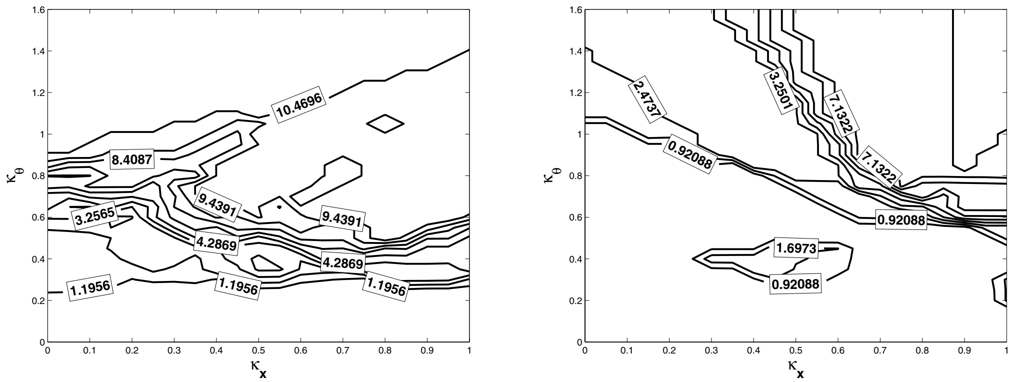
3) Vertical Detergent Bottle: In the same way, Dexter learned which reach-grasp parameterizations were associated with low grasp errors for the vertically presented detergent bottle illustrated in Figure 6(b). Dexter explored different reach controller parameterizations in 56 reaches and grasps where the two-contact grasp controller was executed after each reach. Expected grasp error as a function of the reach controller parameters, κ_x and κ_θ, are illustrated in the contour graph of Figure 5(a). Dexter learned that the smallest grasp controller errors are associated with manipulator orientations where a line passing through the contacts is nearly perpendicular to the object major axis (*i.e.* when κ_θ is near $\frac{\pi}{2}$). This reflects the grasp knowledge that the line of opposition between the two contacts must roughly be perpendicular to the major axis.

4) Horizontal Rectangular Object: Dexter also learned to grasp a horizontally oriented eccentric object (presented at an arbitrary orientation in the horizontal plane). Note that, because the reach controllers are referenced to the visually-perceived object position and orientation, it should be unnecessary to learn how the grasp horizontal objects if the system has already learned to grasp vertical eccentric objects. However, one important difference exists that changes the way these two objects must be grasped – the magnitude of the moment exerted by gravity when the object is grasped and lifted at one of its ends. When a horizontal eccentric object is grasped and lifted at one end, the distance between the grasp point and the CG causes gravity to exert a moment. This does not affect vertically presented eccentric objects because the direction of the gravitational force is not perpendicular to the major axis. Since this effect can cause the object to slip out of the grasp when the object is lifted, Dexter must learn to grasp horizontal eccentric objects near the center.

In a series of 50 grasps where Dexter attempted to grasp and lift a rectangular horizontal eccentric object using different reach controller parameterizations, the system learned that positions near the center of mass were associated with the highest probability of a successful lift. After lifting the object, the grasp was only considered to be a success if the moment induced by gravity was beneath a specified threshold. The results are illustrated in the contour graph of Figure 5(b). This graph shows the expected probability of grasping the object and successfully lifting it as a function of the reach controller parameters, κ_x and κ_θ. Note that this quantity is different from the grasp error of Equation 2 shown in previous contour plots. The graph shows that the probability of grasp and lift success is maximized when the contacts are oriented perpendicular to the object and positioned near the object's center (*i.e.* when κ_x is near 0 and κ_θ is near $\frac{\pi}{2}$.)

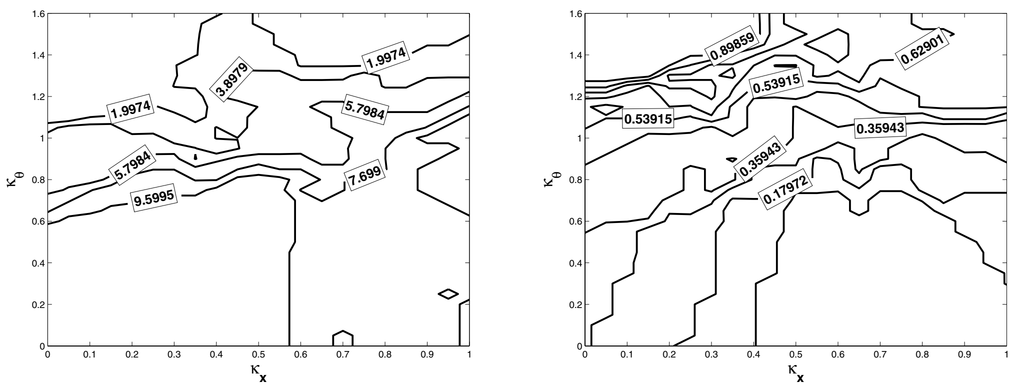
B. Generalizing to New Objects

The previous experiments where Dexter learned to grasp specific objects begs the question regarding whether these grasp skills can generalize to new objects. Whether this is possible depends on how object and object pose is represented to the system. Grasp skills will generalize well when objects and object poses that afford similar grasp strategies are given similar representations. This paper proposes representing objects in terms of coarse blob parameters. While this representation does not capture complex detail of object shape, this paper proposes that it does capture "first order" properties that correlate to basic grasp strategies. Even though the estimate of an object's size and eccentricity cannot be used to precisely place grasp contacts, for simple objects that are well approximated by an ellipsoid, they can be used to select a grasp strategy and to position the contacts in a region around likely good grasp configurations. Even for objects that do not "neatly" correspond to an oriented ellipsoid, it may be possible, in future work, to describe grasp strategies for these more complex objects in terms of strategies that have been learned for "constituent" simpler objects.



(a) (b)

Fig. 4. Contour graphs showing expected grasp error for (a) a vertical cylinder after experiencing 106 reaches and grasps and (b) a 16cm diameter sphere after experiencing 60 reaches and grasps.



(a) (b)

Fig. 5. Contour graphs showing (a) expected grasp error for the vertical detergent bottle shown in Figure 6(b) and (b) the expected probability of grasp and lift success for the horizontal box shown in Figure 6(e).

(a) (b) (c) (d) (e)

Fig. 6. The five training objects used in the generalization experiment.

The extent to which grasp strategies generalize between objects with similar blob parameters was tested in an experiment where Dexter was trained to grasp the five objects illustrated in Figure 6 and tested on the set of 19 new objects illustrated in Figure 7. For each test object, 16 reaches and grasps were executed – eight using the experiences acquired from the five training objects and eight without this experience. On each trial, a parameterization of the reach controller was executed, followed by the two-contact grasp controller, a controller that applied the necessary grasping forces, and a transport controller that lifted the object. During the eight executions that tested performance without experience, Dexter essentially selected random reach controller parameterizations from the set $\Pi_{rx} \cup \Pi_{r\theta x}$. Note that for two of the training objects (the detergent bottle in 6(b) and the horizontal eccentric box in 6(e)), the reach-grasp parameterizations learned have already been described in Section III-A.

During testing, the objects were placed in approximately the

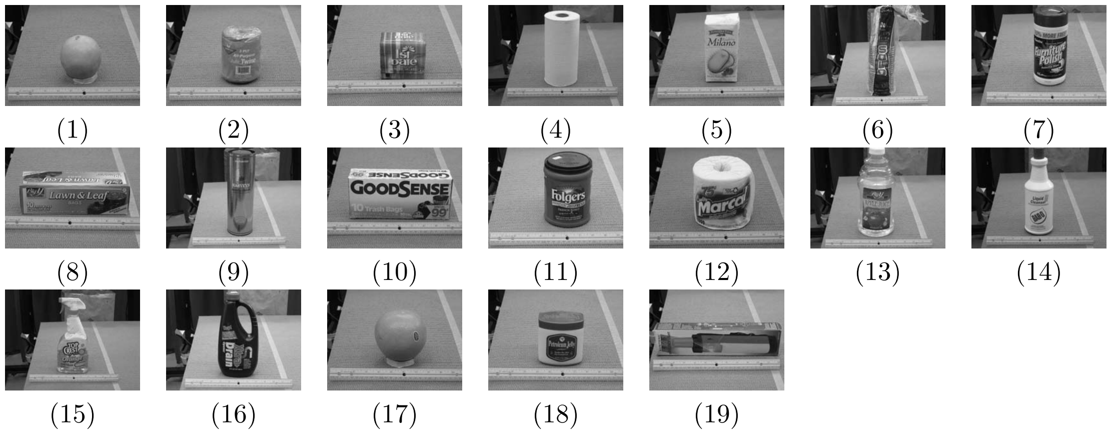


Fig. 7. The 19 test objects used in the generalization experiment.

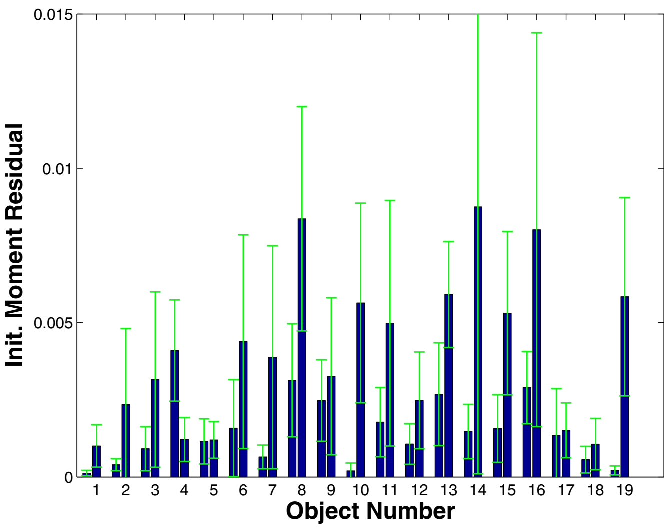


Fig. 8. Generalization: results show that experience grasping a few training objects improves the robot's ability to grasp objects that it has never seen before. The pairs of bars on the horizontal axis show grasp error with (the leftmost bar in each pair) and without (the rightmost bar in each pair) learning experience for each of the 19 test objects. The error bars show a 95% confidence interval around the mean.

same position on the table. The three horizontally presented eccentric objects (objects 8, 10, and 19) were placed at arbitrary orientations in the horizontal plane. The two vertical objects with dissimilar non-principle axes (objects 15 and 16) were always presented in the orientation shown in the Figure 7. The training experiences acquired for each of the five training objects were stored as a function of the blob parameters – major axis length, major/minor ratio (eccentricity), and elevation angle. When presented with a new object, schema structured learning accessed experiences of objects with similar blob parameters and used this information to make grasp decisions for the new object.

Figure 8 illustrates the results. The pairs of bars on the horizontal axis correspond to moment residual error (grasp error) at the beginning of grasp controller execution with and without learning for each of the 19 test objects shown in Figure 7. This is the moment residual error after completing the reach to the object, but before executing the grasp controller. The rightmost bar in each of the 19 pairs shows the mean initial moment residual averaged over eight grasps that did not benefit from the skills learned on the training set. The leftmost bar in each pair shows the mean initial moment residual over the eight grasps that did use the training data. The error bars give a 95% confidence interval around the mean. Since the confidence intervals for many of the objects overlap, the statistical significance of the results for each object was analyzed using a two-sample t-test. Table I shows the t statistic and p-value for each object. The p-value is the probability that learning did not improve grasp performance. Objects 1, 8, 10, 13, 15, and 19 have values for p less than 0.05, indicating that there is a more than 95% probability that learning has improved performance for these objects. If the requirement is lowered to 0.10 (90% percentile), all of the objects except for 5, 9, 17, and 18 show improved grasp performance.

Taken over all objects, the average improvement in grasp performance is significant. Figure 9(a) shows the initial moment residual with and without learning averaged over all 19 objects. The figure shows that after having trained on the set of five objects, when presented with a new (but related) object, the system can be expected to select an instantiation of the reach controller that leads to an initial moment residual of 0.0015N-m with a 95% confidence interval of less than 0.0005N-m. Without learning, Dexter can be expected to do almost three times worse, reaching to an initial moment residual of 0.0042N-m with a 95% confidence interval of 0.0008N-m. The same trend exists when the performance of grasping in terms of the probability of successfully holding and lifting the object is considered. Figure 9(b) shows the probability of a successful lift averaged over all 19 objects with and without learning. When the identity of the object to be grasped is unknown, Figure 9(b) shows that the probability of successfully lifting the object is much better when the robot leverages its previous experience with the training objects.

Object	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
t value	2.5	1.5	1.5	3.2	0.1	1.4	1.7	2.5	0.5	3.3	1.5	1.6	2.7	1.6	2.6	1.6	0.2	1.1	3.4
p value	.01	.07	.08	.01	.46	.09	.05	.01	.30	.01	.08	.06	.01	.06	.01	.07	.42	.16	.01

TABLE I

GENERALIZATION: t VALUES AND p VALUES THAT CALCULATE THE STATISTICAL SIGNIFICANCE OF THE IMPROVEMENT IN INITIAL MOMENT RESIDUAL ERROR FOR EACH OF THE 19 OBJECTS.

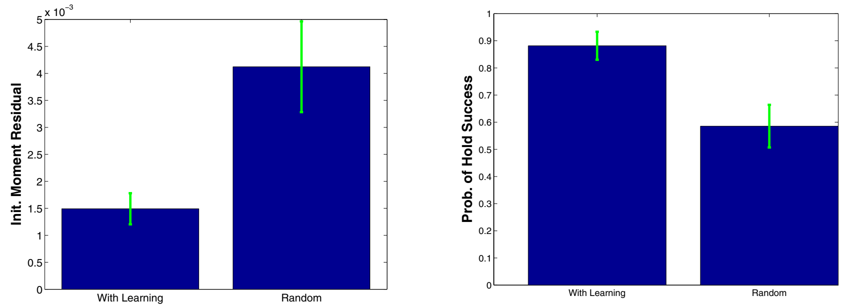


Fig. 9. Generalization: (a) shows the initial moment residual with and without learning averaged over all 19 objects. (b) shows the average probability of successfully lifting each object with (the leftmost bar) and without (the rightmost bar) training experience. In both plots, the error bars show 95% confidence intervals.

IV. 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. Based on visual information, the reach controller moves the grasp contacts into a neighborhood around a good grasp. Then the grasp controller uses tactile feedback to place the contacts in a precise grasp configuration. A strategy is proposed for learning through trial-and-error which parameterizations of the reach and grasp controllers are appropriate for different objects and object poses. This grasp knowledge is organized in terms of ellipsoidal parameters that describe the object that was grasped. Each object encountered by the system is characterized in terms of the ellipsoid that most closely matches the visually segmented object. New objects with similar ellipsoidal parameters are assumed to be graspable in similar ways. Experimental results show that this approach is capable of learning, for specific objects, which reach and grasp controller parameterizations are likely to be successful. In addition, results show that it is possible to learn reach-grasp skills based on experience with a limited set of objects and successfully apply these skills to new objects.

ACKNOWLEDGMENT

The work described in this paper was completed while the first author was a student at the University of Massachusetts. 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] R. Platt, "Learning and generalizing control-based grasping and manipulation skills," Ph.D. dissertation, University of Massachusetts, September 2006.

[4] 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.

[5] A. Saxena, J. Driemeyer, J. Kearns, C. Osondu, and A. Y. Ng, "Learning to grasp novel objects using vision," in *International Symposium on Experimental Robotics*, 2006.

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

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

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

[9] 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.

[10] 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.

[11] 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.