Going beyond the perception of affordances: Learning how to actualize them through behavioral parameters

Emre Ugur[1,2,3], Erhan Oztop[2,1,4] and Erol Şahin[3]

[1] Biological ICT, National Institute of Information and Communication Technology, Kyoto, Japan
[2] Cognitive Mechanisms Labs., Advanced Telecommunications Institute International, Kyoto, Japan
[3] KOVAN Research Lab., Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
[4] School of Engineering Science, Osaka University, Osaka, Japan
Emails: emre@atr.jp, erol@ceng.metu.edu.tr, erhan@atr.jp

***Abstract*— In this paper, we propose a method that enables a robot to learn not only the existence of affordances provided by objects, but also the behavioral parameters required to actualize them, and the prediction of effects generated on the objects in an unsupervised way. In a previous study, it was shown that through self-interaction and self-observation, analogous to an infant, an anthropomorphic robot can learn object affordances in a completely unsupervised way, and use this knowledge to make plans in its perceptual space. This paper extends the affordances model proposed in that study by using parametric behaviors and including the behavior parameters into affordance learning and goal-oriented plan generation. Furthermore, for handling complex behaviors and complex objects (such as execution of precision grasp on a mug), the perceptual processing is improved by using a combination of local and global features. Finally, a hierarchical clustering algorithm is used to discover the affordances in non-homogenous feature space. In short, object affordances for object manipulation are discovered together with behavior parameters based on the monitored effects.**

I. INTRODUCTION

In the postnatal age of 7-10 months, the infant explores the environment actively. By observing the effects of her hitting, grasping and dropping actions on objects, she can learn the dynamics of the objects [1]. The infant in this stage has already acquired a number of manipulation behaviors and is able to detect different properties of objects such as shape, position, color, etc. Using her motor skills, the infant interacts with the environment and observes the changes she creates via her perceptual system, accumulating knowledge about the relationships between objects, actions and the effects.

Within this developmental stage, the infant not only learns *what* type of affordances are offered by the object, but also learns *how* he can actualize them. For instance, she learns not only that a milk bottle is graspable, but also at which angle her hand should approach the bottle to successfully make the grasp. At this period, she demonstrates different modes of grasping such as power-grasp which relies on synergistic control of the hand as a whole, and precision-grasp that requires delicate distal finger control. It is not clear whether the two types of grasps develop from a single rudimentary grasping behavior or develop independently. However it is known that infants in that age do not have the complete adult level visuo-motor grasp execution ability [2], thus the control of grasp behavior develops with the perception of the affordance graspability.

During the recent years, studies inspired by ideas in developmental psychology have increased considerably [1], [3]. In these studies, the agent typically acquires the ability to make predictions about the effects it can create through active exploration of the environment. For example, [4] proposed methods for the self-discovery of the affordances, where the effect categories were found through unsupervised clustering in the effect space. [5] used probabilistic networks that capture the stochastic relations between objects, actions and effects. These networks allow bi-directional relation learning and prediction. Although these systems gained the ability to predict the effect to be generated, they cannot predict more than one step ahead, which prohibits complex planning. In [6] on the other hand, after learning, the robots could make multi-step predictions using transition rules and hence were able to demonstrate complex planning. Their approach is different from ours since sensorimotor experience of the robot was used to associate the predicates of the AI rules.

This paper builds upon our previous work [7] which addresses how symbolic planning operators, as opposed to the symbols used in planning, can be grounded in the continuous sensory-motor experiences of a robot from a developmental point of view. Our approach is inspired from the notion of affordances [8], for which we provided a computational framework in [9]. The current work extends our previous work by (1) going from discrete fixed behaviors to continuous behaviors and by (2) addressing a more complex behavior, namely grasping. In the previous work [7], we learned the effect of actions within the framework of [9] with the assumption that the behaviors are fixedi, i.e. the behaviors have no free parameters. Here we show how this assumption can be removed. The current work also addresses learning to grasp in two modes: precision and power grasping. Grasp learning is a complex task [2]; here we adopt a minimalist representation for the grasp actions requiring two parameters, namely the target position and the approach direction. We choose to have the former to be determined by the object location and orientation uniquely. The approach angle, on the other hand, represents the freedom in grasping and used

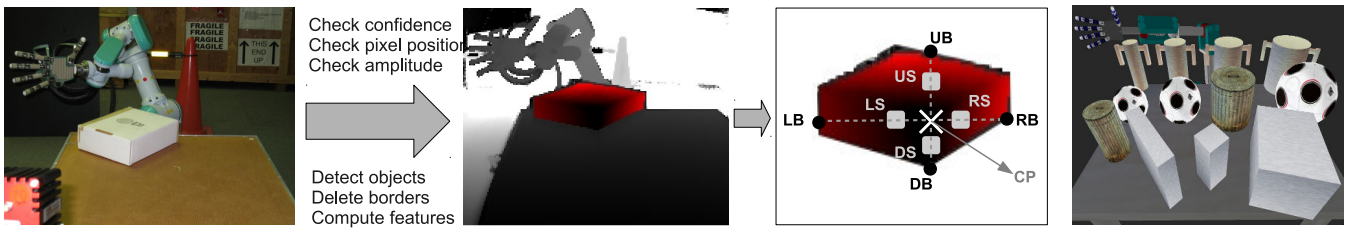


Fig. 1. In (a), the 23 DOF hand-arm robotic platform, infrared range camera (on the bottom-left) and one object that is used in this study are shown. In (b) the range image obtained from the range camera and the detected object are shown. The pixels and surface patches that are used in feature computation. The range image is scanned in four different directions starting from Closest Pixel (CP, shown by cross). Four neighbor rectangular surface patches and four border pixels are detected. U (up), D (down), L (left), and R (right) stand for four directions. Thus LS and LB means *left surface patches* and *left border*, respectively. Surface patches in different directions contain fixed number of (5x5=25) pixels at CP's neighborhood. A snapshot from the robot simulator is shown in (d) with a number of sample objects used in training. Note that the size and orientation of the objects are randomly set during training.

by our learning system to discover the grasp actions that are suitable for the given object.

II. Affordances and Robot Control

The notion of *affordances* was proposed by J.J. Gibson, to refer to the action possibilities offered to the organism by its environment [8]. For example, a horizontal and rigid surface affords walk-ability, and a flat surface at a certain height affords sit-tability. This notion emphasizes the complementarity of a robot and its environment and claims that affordances are determined by both the properties of the objects and the capabilities of the organism. A small cobblestone may afford hide-ability to a mouse, while affording throw-ability to us.

Recently, we proposed a formalism for using affordances as a framework at different levels of robot control ranging from perceptual learning to planning [9]. The proposed formalism agrees with the Gibsonian view that affordances are relations within the agent-environment system, but it also extends this view by arguing that these relationships can also be represented in the agent (a.k.a. robot). Specifically, the formalism defines affordances as general relations that pertain to the robot-environment interaction and claimed that they can be represented as a triple which consist of the initial percept of the object, the behavior applied and the effect produces. For instance, the lift-ability affordance is a relation between the properties of an object, the behavioral capabilities of the robot and the type of effect produced by the lift behavior. In this paper, we used this framework to propose a developmental method that enables a robot to learn the symbolic relations that pertain to its interactions with the world and show that they can be used in planning.

III. EXPERIMENTAL FRAMEWORK

An anthropomorphic robotic system (Fig. 1 (a)), equipped with a range camera, and its physics-based simulator is used as the experimental platform (Fig. 1 (d)). The robot platform consists of a five fingered 16 DOF Gifu robot hand and 7 DOF PA-10 robot arm. For robot perception, SwissRanger SR-4000 infrared range camera, with 176×144 pixel array, 0.23° angular resolution and 1 cm distance accuracy was used. Along with the range image, the camera also provides grayscale image of the scene that enables us to differentiate the robot hand from objects.

The simulator (Fig. 2, 3), developed using the Open Dynamics Engine (ODE) library, is used during the exploration phase. The range camera is simulated by sending a 176×144 ray array from camera center with 0.23° angular intervals.

A. Interactions

What type of interactions the robot can perform on the objects depend on the diversity of its behavior repertoire. In this work, five different behaviors, that are assumed to be learned in a previous developmental stage, are used to manipulate the objects in the environment. These behaviors are triggered with different mechanisms based on the internal and external sensors. We postulate that manipulation behaviors are executed over object's closest point (CP) to the robot. Thus, if an object is detected on the table, the position of the closest point (CP), computed from the range camera, is used to reach to and interact with the object by the behaviors triggered by external sensors.

How the object are affected from the execution of the same behavior, on the other hand, depends on the free parameters of these behaviors. For simplicity, each behavior is modulated with one parameter, α. The 5 behaviors and their modulation strategy is as follows:

Open-hand(α): : The robot rotates its wrist in α angle and opens its hand.

Move-hand(α): : The robot moves its hand 10 cm in α *direction*.

Push-object(α): : The robot pushes the object for 10 cm approaching from α *direction*[1].

Power-grasp(α): : The hand approaches wide-open from α *direction* to the CP of the object. When palm-touch sensor is activated or the hand reaches the desired position (CP), all the fingers are closed and the hand is lifted.

Precision-grasp(α): : The hand approaches from α *direction* to the CP of the object. Different from power-grasp, only thumb and index fingers are used to make a precision

[1]During object manipulation the robot hand is moved only in horizontal plane above the table, thus *direction* parameter can also be represented by one angle.

Fig. 2. The execution of power-grasp behavior and the final object range image. The arrow shows the corresponding approach direction (α).

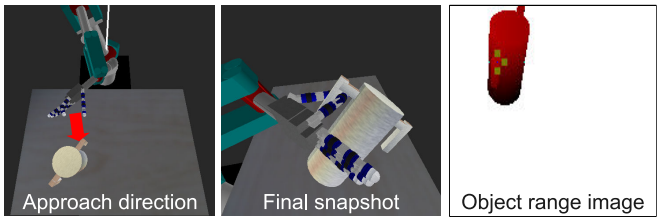


Fig. 3. The execution of precision-grasp behavior and the final object range image. The arrow shows the corresponding approach direction (α).

grasp when the tip of these fingers reach CP. The hand is lifted after the fingers are closed.

B. Objects

The robot interacts with four types of objects; namely boxes, cylinders, spheres, and objects with handles, all in different size and orientations (Fig. 1 (d)). During the execution of its behaviors with different parameters, the robot may experience interactions with objects and face with different consequences. For instance when the hand pushes boxes or upright cylinders, the objects will remain on the table, but if it pushes spheres the objects will roll down the table. As another example, the same box can be grasped from one approach direction while cannot be grasped from other directions. Note that in order to avoid robot arm - camera collision, the camera is placed on the other side of the table. On the other hand, the robot interacts with the closest point (CP) of the object and the closest point is generally out of view of the camera. Thus, only symmetric objects, which provide mirrored but same information from robot and camera views, are used in experiments.

C. Perception

a) Object Detection: The first step of pre-processing is to filter out the pixels whose confidence values are below an empirically selected threshold value. Then the pixels outside the region of interest are filtered out. As a result, the remaining pixels of the range image would belong to one or more objects that are segmented by the Connected Component Labeling algorithm [10]. In order to reduce the effect of camera noise, the pixels at the boundary of the object are removed, and the Median and Gaussian filters with 5×5 window sizes are applied (see Fig. 1 (b) for a sample range image). Finally, a feature vector for each object is computed using the positions of the corresponding object pixels as detailed in the next paragraph.

b) Object feature vector computation: The perceptual state of the robot at time t is denoted as $[\boldsymbol{f}_{o_0}^{t,()}, \boldsymbol{f}_{o_1}^{t,()}..]$ where $\boldsymbol{f}$ is a feature vector of size 25, and the superscript $^{()}$ denotes that no behavior has been executed on the object yet. Six channels of information are gathered and encoded in a feature vector for the object.

Behavior execution on the objects are performed through interaction with objects' closest point (CP). Thus, the interaction results are affected by the properties of the CP and its local neighborhood. A number of pixels and surface patches, related to CP, are detected by scanning the range image in four different directions as shown in Fig. 1 (c). Then,

- the position of CP (3 features),
- the distance of CP to each border pixel (4 features),
- the distance of CP to the center of each surface patch (4 features),
- the mean normal vector for each surface (12 features),
- the visibility of the object (1 binary feature), and
- the touch sensor on the hand (1 binary feature)

are included into the feature set.

c) Effect feature vector computation: For each object, the effect created by a behavior is defined as the difference between its final and initial features:

$$\boldsymbol{f}_{\text{effect},o_i}^{(b_j)} = \boldsymbol{f}_{o_i}^{(b_j)} - \boldsymbol{f}_{o_i}^{()}$$

where $\boldsymbol{f}_{o_i}^{(b_j)}$ represents the final feature vector computed for object o_i after the execution of behavior b_j.

IV. LEARNING OF AFFORDANCE RELATIONS

The exploration phase, conducted only in simulation, consists of episodes, where the robot interacts with the objects, and monitors the changes. The data from an interaction is recorded in the form of $< \boldsymbol{f}_{\text{effect}}^{b_j}, \boldsymbol{f}^{()}, b_j(\alpha) >$ tuples, i.e. (object, effect, behavior) instances. Here, α is the parameter of the behavior b_j used for interaction, $f^{()}$ and $f_{\text{effect}}^{b_j}$ denote the initial object feature vector and the difference between final and initial feature vectors, respectively.

The learning process consists of two steps: the unsupervised discovery of effect categories, and the training of classifiers to predict the effect categories from object features. The learning process is applied separately for each behavior as detailed below.

Effect category discovery: In the first step, the effect categories and their prototypes are discovered through a hierarchical clustering algorithm. In the lower level, channel-specific effect categories are found by clustering in the space of each channel, discovering separate categories. In the upper level, the channel-specific effect categories are combined to obtain all-channel effect categories using the Cartesian product operation. Finally, the effect categories that occur rarely are automatically discarded together with their members. This hierarchical clustering method is superior to simple one-level clustering method, since the results of one-level clustering are sensitive to the relative weighting of the effect features that are encoded in different units (e.g. continuous position features vs. binary visibility feature).

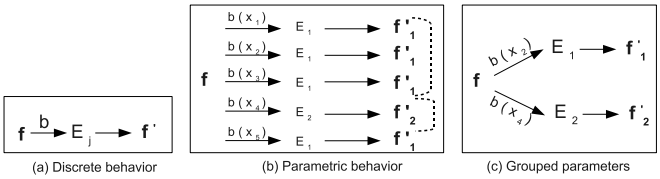
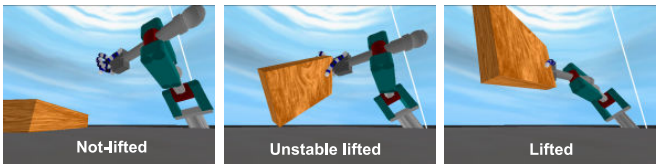


Fig. 4. Given object features ($\boldsymbol{f}$) and behavior-id (b), the effect category (E_j) and the next state ($\boldsymbol{f'}$) can be predicted by using the corresponding *svmPredictor()* and prototype features. (a) and (b) shows the next state prediction using discrete and parametric behaviors, respectively. A grouping and averaging mechanisms is used to choose the most reliable behavior parameters that transform the current object perceptual state to one of the possible states the corresponding behavior can transform.



(a) $\alpha = -35^\circ$ (b) $\alpha = -15^\circ$ (c) $\alpha = 25^\circ$

Fig. 5. The interaction results for 3 different cases from Fig. 7 are shown. Object angle is always kept as -45° but the approach angle α is changed.

TABLE I

EFFECT CATEGORY PROTOTYPES DISCOVERED FOR POWER-GRASP.

Only significant changes are given in the table. The comments are provided for the effect prototypes and are not used during experiments.

Effect id	Visibility	Position (x,y,z)	Touch	Comment
Effect 1	0	+3cm,+2cm,+2cm	0	Not-lifted
Effect 2	0	+3cm,**+13cm**,+3cm	**+1**	Lifted
Effect 3	0	+3cm,+2cm,+2cm	**+1**	Unstable lifted
Effect 4	**-1**	+3cm,+2cm,+2cm	0	Disappeared

After discovering the effect categories and assigning each feature vector in the set of $\{\boldsymbol{f}^{b_j}_{\text{effect}}\}$ to one of the effect categories ($E_{b_j,id}$), the prototype effect vectors ($\boldsymbol{f}^{b_j}_{\text{prototype},id}$) are computed as the average of the category members.

Learning effect category prediction: In the second step, classifiers are trained to predict the effect category for a given object feature vector, a behavior id and behavior's parameter by learning the $(\boldsymbol{f}^{()}, \alpha) \rightarrow E_{b_j,id}$ mapping. Specifically, we used a Support Vector Machine (SVM) classifier with Radial Basis Function (RBF) kernel to learn this mapping for each behavior b_j, where $(\boldsymbol{f}^{()}, \alpha)$ is given as the input, and the corresponding $E_{b_j,id}$ as the target category.

V. BEHAVIOR PARAMETER SELECTION FOR GOAL-ORIENTED AFFORDANCE USE

The trained SVM classifiers allow the robot to predict the *effect category* that is expected to be generated on an *object* by a *behavior* controlled with a particular *parameter*:

$$E^{\text{predicted}}_{b_j,id} = svmPredict^{b_j}(\boldsymbol{f}^{()}, \alpha).$$

The predicted next percept of the object can be found as:

$$\boldsymbol{f'}^{(b_j(\alpha))} = FM^{b_j}(\boldsymbol{f}^{()}, \alpha) = \boldsymbol{f}^{()} + \boldsymbol{f}^{b_j}_{\text{prototype},id^{\text{predicted}}}$$

Effectively, this corresponds to a forward model (FM) that returns the next perceptual state of the object. By successively applying this model, the robot can predict the perceptual state of the object for any number of sequentially executed behaviors. This multi-step prediction ability has already been proven to be useful in satisfying goals that were encoded in perceptual space with discrete behaviors in [7] .

Predicting the next state of the object for any discrete behavior is straightforward since given initial object features, the SVM classifier will predict only one effect category and FM will give only one next state as shown in Fig. 4 (a).

On the other hand, one non-discrete behavior can create many different effects on the same object when controlled with different parameters. The next state predictions also depend on the behavior parameter since it is an input to $svmPredict()$, thus different next state predictions can be obtained when whole parameter space of the behavior is considered as shown in Fig. 4 (b). Still, the number of effect categories is fixed for each behavior and the possible next states are limited with this number. As a result, the problem can be transformed to 'finding the most reliable behavior parameter to reach a possible next state'. For this purpose, (1) a grid search is done in continuous parameter space; (2) behaviors which transform the current state to the same state are grouped together; (3) the largest group for each next different state is found; and (4) the mean parameter value in each group is selected as the best parameter that transforms the current state to the corresponding next state. Fig. 4 (b,c) illustrates this method in a simple example.

VI. EXPERIMENTS

In the experiments, a table with 100×70 cm^2 surface area was placed with a distance of 40 cm in front of the robot, as shown in Fig. 1. At the beginning of each exploration trial, one random object of random size $[8cm - 40cm]$ was placed on the table at random orientation. For all behaviors, 2000 interactions were simulated with random parameters and the resulting set of relation instances were used in learning. The X-means algorithm was used to find channel-specific effect categories, and Support Vector Machine (SVM) classifiers were employed to learn effect category prediction.

A. Discovered effect categories for grasp behaviors

For the *power-grasp* behavior, 4 clusters were found to represent whole effect space as shown in Table I. Large objects could not be lifted resulting in *not-lifted effect*. Small objects could be lifted so the height is increased and touch sensor is activated as shown in prototype of *lifted effect*. In some cases, the grasp was not stable, so the object slided from robot's hand during lifting but remained in contact with the hand, creating *unstable-lifted effect* (Fig. 5 (b)). In this effect, the vertical position of the object was not increased (significantly), however the touch sensor remained activated. The *disappeared* effect was created by the spheres that roll away during interaction.

TABLE II

EFFECT CATEGORY PROTOTYPES DISCOVERED FOR *precision-grasp*.

Effect id	Visibility	Position (x,y,z)	Touch	Comment
Effect 1	0	+6cm,-1cm,+4cm	0	Not-lifted
Effect 2	0	+5cm,**+10cm**,+2cm	**+1**	Lifted
Effect 3	**-1**	+6cm,-1cm,+4cm	0	Disappeared

For the *precision-grasp* behavior, 3 effect categories were obtained as shown in Table II. Because the robot inserted one of its fingers through the aperture of the handle, the grasps were more stable once the object is hold.

B. Effect prediction in power grasp behavior

After the discovery of effect categories, the mapping from the initial object features to these categories was learned for each behavior b_j ($Predictor^{b_j}()$) by multi-class Support Vector Machines (SVMs). The Libsvm software package was used with optimized parameters of the RBF kernel through cross-validated grid-search in parameter space. Different sets of 1000 simulated interactions were used in training and for testing. At the end, 72% accuracy was obtained in predicting the correct effect categories for *power-grasp* behavior. The low accuracy is due to the difficulty in predicting *unstable-lifted* effect category since it corresponds to the critical point between success and failure in liftability. When this category is discarded from the sample set, the prediction accuracy in predicting the three categories is increased to 85% in average.

We analyzed the relevance of the features in affordance prediction for the *power-grasp* and *precision-grasp*. For this purpose, we used Schemata Search [11] which computes the relevance of a feature based on its impact on the prediction accuracy. The Schemata Search is a greedy iterative method that starts with the full feature set (R_0), and shrinks it by removing the least relevant feature (based on its impact on prediction accuracy) in each iteration.

Fig. 6 (a) and (b) gives the prediction accuracy results with different feature sets, with and without *unstable-lifted* effect. When the feature relevance is examined, behavior parameter (α) is among the most relevant features as presented. The other relevant features represent CP's object-relative properties and CP's local surface angles. For example, *distance to right border* and *distance to left border* encodes the location of CP with respect to object and left/right surface normals represent the shape of the CP's local neighborhood.

We systematically analyzed the success in effect prediction by comparing real and predicted effect categories using a fixed size box which is graspable from only one side. It is rotated in 10° intervals and in each object orientation, *power-grasp* behaviors were executed with varying (reachable) approach direction angles from -70° to 40°. The real effect categories obtained during these interaction were shown in Fig. 7 with different colors. Predicting the relation between object-angle and approach-angle, which determines the liftability of the objects, is non-trivial as the robotic hand is not a simple gripper. There is hardly any symmetry

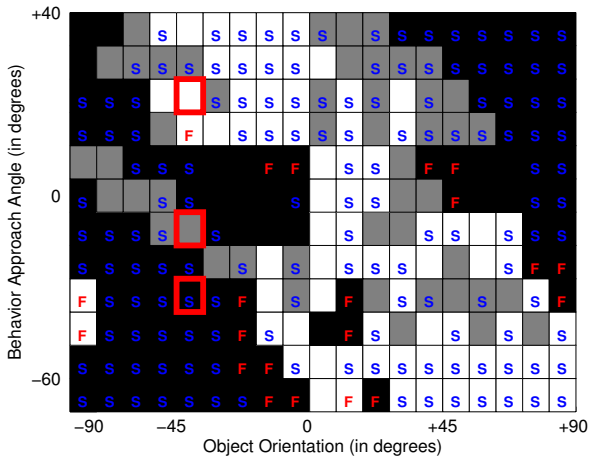


Fig. 7. The comparison of real and predicted effect categories for different object orientations and power-grasp approach directions. The color of the regions corresponds to observed real effect categories; black: *Not-lifted effect*, white: *Lifted effect*, and gray: *Unstable lifted effect*. The 'S' and 'F' labels corresponds to prediction success and failure. If the prediction or real effect category is *unstable-lifted*, then the corresponding box is not labeled. The cases marked with bold red boxes are shown in Fig. 5.

(a) *Power-grasp* ($\alpha = 5^\circ$) (b) *Power-grasp* ($\alpha = -25^\circ$)

Fig. 8. Objects in different orientations were grasped with different approach angles.

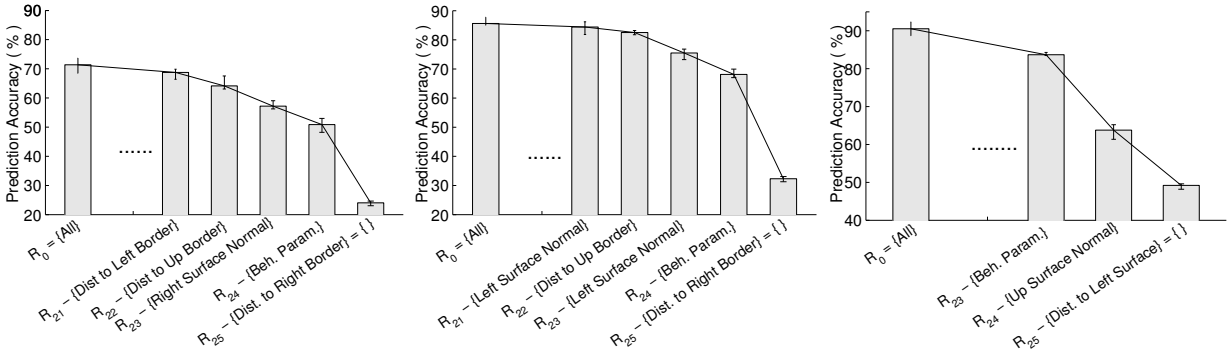
between these two components (e.g. while objects at 60° were lifted by *power-grasp*(-20°), objects at -60° could not be lifted by *power-grasp*(20°). Furthermore, there are many 'gray' regions which corresponded to *unstable-lifted effect* that are distributed between *lifted* and *dragged* regions. Our method was able to predict many effect categories correctly, however failed to predict some that reside in critical border.

C. Real Robot Results

The results obtained in the simulator were partially verified on the real robot platform. For this purpose, the effect category prediction system was transferred to the real robot. A box shaped object and an object with a handle were used to assess the ability in prediction of *lift effect* with *power-grasp* and *precision-grasp* behaviors, respectively.

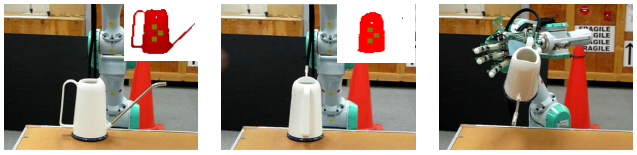
The box shaped object was placed in two different orientations as shown in Fig. 8. As a result, the behaviors that were predicted to lift the objects from their narrow side were parameterized with different angles.

The watering can was placed in two different orientations: In the first orientation, the closest point (CP) was on its



(a) Power-grasp behavior (4 categories) (b) Power-grasp behavior (3 categories) (c) Precision-grasp behavior (3 categories)

Fig. 6. The prediction accuracies of the classifiers that are computed using different feature sets. The feature set is increased by one feature in each iteration by adding the most successful one. The left-most and right-most bars in each plot show the results obtained using all and no features, respectively. Error bars on prediction accuracies indicate the best, median, and worst classifiers found by 10-fold cross-validation. Many features, when discarded from the training set, did not change the prediction accuracy significantly. Thus their prediction accuracies are not shown, and they are represented with '.....'.



(a) No precision grasp (b) Precision grasp [-90,+90] (c) Result of precision grasp

Fig. 9. The object is correctly predicted not to be liftable in (a). The same object when rotated is predicted to become liftable with any *precision-grasp* behavior. Thus, it is approached with 0° and lifted up.

main body; and in the other one, the CP was on the handle (Fig. 9). The robot computes the features based on CP, so the results were different. In (a), no precision grasp was predicted to lift the object, where in (b) precision grasps from all directions were predicted to lift the object since the handle was reachable from all directions. When the average of these directions were used as the final parameter, the object was approached from behind and lifted up. The movie for this behavior is available at
`http://www.emreugur.net/movies/icra2011/`.

VII. CONCLUSION

In this paper, we proposed a method that allows a robot not only to discover *what* type of affordances are offered by the objects but also to learn *how* to actualize them. After robot's exploration, the effect of behavior parameters over discovered affordances were learned in relation with the object features and the generated effects. In this context, we proposed a method to select the behavior parameters to reach desired goals. This enabled the robot to predict the objects' next perceptual state based on the current object features and the behavior parameters. This prediction ability was used to satisfy particular goals, i.e. to reach desired final states.

The proposed method is able to not only predict the type of effect that will be generated by a behavior for a certain type of parameter value, but also the change to be generated on the object as a result of execution. This property allows us to use these relations for making multi-step plans [7].

VIII. ACKNOWLEDGMENTS

Emre Ugur acknowledges the financial support of TÜBİTAK (The Scientific and Technical Research Council of Turkey). This research was supported in part by Global COE Program "Center of Human-Friendly Robotics Based on Cognitive Neuroscience" of the Ministry of Education, Culture, Sports, Science and Technology, Japan. It was was also partially funded by the European Commission under the ROSSI project (FP7-21625) and the TÜBİTAK through project 109E033.

REFERENCES

[1] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: a survey," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.

[2] E. Oztop, N. Bradley, and M. Arbib, "Infant grasp learning: a computational model," *Experimental Brain Research*, vol. 158, pp. 1354–1361, 2004.

[3] A. Stoytchev, "Some basic principles of developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 122–130, Aug. 2009.

[4] S. Griffith, J. Sinapov, M. Miller, and A. Stoytchev, "Toward interactive learning of object categories by a robot: A case study with container and non-container objects," in *Proc. of the 8th IEEE Intl. Conf. on Development and Learning (ICDL)*, June 2009, pp. 1–6.

[5] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor maps to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[6] J. Modayil and B. Kuipers, "The Initial Development of Object Knowledge by a Learning Robot." *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 879–890, Nov. 2008.

[7] E. Ugur, E. Sahin, and E. Oztop, "Affordance learning from range data for multi-step planning," in *Proceedings of The 9th International Conference on Epigenetic Robotics, EpiRob'09*, 2009, pp. 177–184.

[8] J. Gibson, *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986.

[9] E. Şahin, M. Çakmak, M. R. Doğar, E. Ugur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, pp. 447–472, 2007.

[10] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision, Volume I*. Addison-Wesley, 1992.

[11] A. Moore and M. Lee, "Efficient algorithms for minimizing cross validation error," in *Proceedings of the 11th International Conference on Machine Learning*. Morgan Kaufmann, 1994, pp. 190–198.