Unsupervised Learning of Object Affordances for Planning in a Mobile Manipulation Platform

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

[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] Osaka University, Osaka, Japan
Email: emre@atr.jp

Abstract—**In this paper, we use the notion of affordances, proposed in cognitive science, as a framework to propose a developmental method that would enable a robot to ground symbolic planning mechanisms in the continuous sensory-motor experiences of a robot. We propose a method that allows a robot to learn the symbolic relations that pertain to its interactions with the world and show that they can be used in planning. Specifically, the robot interacts with the objects in its environment using a pre-coded repertoire of behaviors and records its interactions in a triple that consist of the initial percept of the object, the behavior applied and its effect, defined as the difference between the initial and the final percept. The method allows the robot to learn object affordance relations which can be used to predict the change in the percept of the object when a certain behavior is applied. These relations can then be used to develop plans using forward chaining. The method is implemented and evaluated on a mobile robot system with limited object manipulation capabilities. We have shown that the robot is able to learn the physical affordances of objects from range images and use them to build symbols and relations that can be used in making multi-step predictions about the affordances of objects and achieve complex goals.**

I. Introduction

There exists a representational gap between the continuous sensory-motor experiences of a robot and the symbolic planning operators of Artificial Intelligence. The mapping of the symbols used in these operators onto the sensory-motor readings of the robot's continuous world is typically referred as part of the symbol grounding problem[1] and has been studied since the days of STRIPS. These studies[2] typically assume that the relations that bind the pre-coded symbols (such as pre-conditions and effects of an operator) are given, and aim to learn the mapping from these symbols to the continuous sensory-motor readings of the robot. Recently, it has been argued that symbols "are not formed in isolation" and that "they are formed in relation to the experience of agents, through their perceptual/motor apparatuses, in their world and linked to their goals and actions"[3].

In this paper we are interested in 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 view, which takes its inspiration from the notion of affordances developed in cognitive science, argues that the learning of relations that encode the interactions of the robot with its world implicitly encapsulates the learning of symbols. Moreover, we claim that such relations enable the robot to make predictions about the world state and build plans to achieve complex goals.

In the postnatal age of 7-10 months, the infant interacts with the environment and observes the changes created via her perceptual system building the knowledge of the relationships between objects, actions and the effects [4]. This process is effectively learning the affordances[5], action possibilities provided by the objects in the environment of the infant. The learning at this stage is largely performed in a goal-free fashion through self-exploration and self-observation [6]. After approximately 9 months of age, the infants start using the learned object-action-effect relations in a goal-directed to generate simple plans that may involve multiple steps [7]. The proposed learning system shares crucial elements with the development of infants of 7-10 months age.

A. Affordances and Robot Control

The theory of affordances that was introduced by Ecological Psychologist J.J. Gibson [5] states that the organisms do not need to recognize the action-free meanings of the objects and make complex inferences over these meanings in order to act on them. For example we do not identify the objects with their action-free labels such as chairs, couches or stones when we need to throw them or sit on them. Instead, we look for a specific combination of the object properties taken with reference to us and our actions in order to detect their 'throwability' or 'sittability' affordances. Since affordances tightly couple the actions of the organism with its environment, and emphasize the existence and necessity of an action-oriented perception view, recently the concept has started to be used in robot control.

Recently, we proposed a formalism for using affordances as a framework at different levels of robot control [8]. 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 defined affordances as general relations that pertain to the robot-environment interaction and claimed that they can

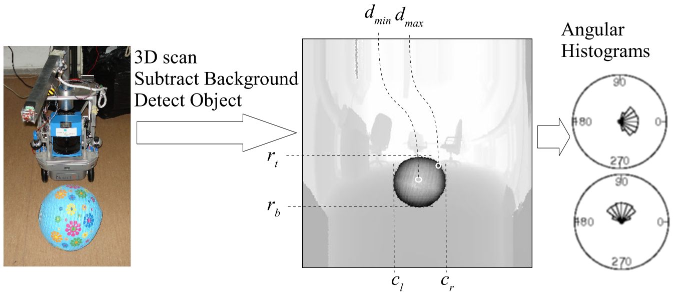

Fig. 1. On the left, the robot and a spherical object shown. On the right, the range image obtained from the 3D scan is given. The subtracted background and other objects are blurred. Distance, relative position and shape related features are shown.

be represented as a triple which consist of the initial percept of the object, the behavior applied and the effect produced. 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.

II. Experimental Framework

A medium-sized ($45cm \times 33cm \times 47cm$), differential drive mobile robot, equipped with a 3D laser range finder, and its physics-based simulator, is used as the experimental platform. The 3D laser scanner is based on a SICK LMS 200 2D laser scanner, rotated vertically with an RC-servo motor. The 3D laser scanner has a horizontal range of 180°, and is able to sweep a vertical range of $\pm 82.8°$ to produce a 720×720 range image. A crane arm is mounted on top of the robot with 3 degrees of freedom and an electromagnetic gripper to manipulate magnetizable objects. The arm can rotate around itself, move the gripper back-and-forth in a range of $55cm$, and lift its magnet up and down.

The robot is equipped with five **move behaviors** and a **lift behavior**. The move behaviors (*move-forward*, $move_{\pm 30°}$, and $move_{\pm 60°}$) rotate the robot as specified by the type of the behavior and drives it forward for $40cms$. The robot is also endowed with a closed-loop lift behavior, which is triggered by an object region in the range image to lift the object whose relative position can be computed from the range image.

The robot interacts with three types of objects; namely boxes, cylinders and spheres, at different size and orientations. During the execution of its move behaviors, the robot may experience collisions with objects and face with different consequences. For instance, when the robot collides with boxes or upright cylinders, it would come to a stop as a result of the physical interaction. However, when the robot collides with a sphere, the sphere would roll away not blocking the robot's movement. The robot may or may not get blocked when it collides with lying cylinders depending on the relative orientation of the cylinder. The lift behavior would succeed in lifting an object, if the object is within the arm length of the crane and has a flat top (assuming that all objects are magnetizable). In this sense, all boxes and cylinders are liftable, whereas spheres and lying cylinders are not.

A. Perception

The robot perceives the world through its 3D range scanner. First, the range image is down-scaled to 360×360 for noise reduction and is subtracted from the *background image* that was obtained from an empty environment. 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 [9]. For each object, o, a feature vector, $\boldsymbol{p}_o$ is computed (see Fig. 1). The perception of the robot at time t is denoted as $[\boldsymbol{p}^t_{o_1} \dots \boldsymbol{p}^t_{o_m}]$, where $\boldsymbol{p}_o$ is the feature vector of object o, and m is the number of objects segmented. $\boldsymbol{p}_o$ is a vector of size 44 and is represented as follows:

$$\boldsymbol{p}_o = [d_{min}, d_{avg}, d_{max}, a, r_t, r_b, c_l, c_r, \varphi_1 \dots \varphi_{18}, \theta_1 \dots \theta_{18}]$$

where $d_{min}, d_{avg}, d_{max}$ denotes the minimum, average and maximum range values, a is the area measured in pixels, r_t, r_b, c_l, c_r are the indexes of the top and bottom rows, and the left and right columns of the bounding box, and φ_i and θ_j represent the frequency histogram of normal vector angles in latitude and longitude. Histogram encodes the distribution of local surface normal vectors of the objects ([10]).

For each object, the effect created by a behavior is computed as the difference between the final and initial features:

$$\boldsymbol{\xi}^b_o = \boldsymbol{p'}_o - \boldsymbol{p}_o$$

where $\boldsymbol{\xi}^b_o$, $\boldsymbol{p'}_o$ and $\boldsymbol{p}_o$ represents the effect, final and initial feature vectors, and b represents the behavior executed. Each interaction of the robot is used to create a *relation instance* of the form $(\boldsymbol{\xi}^b_o, \boldsymbol{p}_o, b)$.

III. The Learning of Affordance Relations

During the interaction phase, the robot interacts with the environment and creates *relation instances* of the form $(\boldsymbol{\xi}^b_o, \boldsymbol{p}_o, b)$. Then, for each behavior b the relation instances are grouped into a set: $\mathbf{I} = \{(\boldsymbol{\xi}^b_o, \boldsymbol{p}_o, b)\}$. The effects of each behavior, that is $\{\boldsymbol{\xi}^b_o\}$, are grouped using k-means clustering and for each cluster an *effect-id* is assigned. The associated *effect prototype* for the cluster is defined as the mean of the cluster, denoted as $\boldsymbol{C}^b_i$. Hence, for a given $\boldsymbol{\xi}$, the corresponding *effect-id* can be found as:

$$\textit{effect-id}_{\boldsymbol{\xi}} = \underset{1 \leq i \leq k}{\operatorname{argmin}} \|\boldsymbol{\xi} - \boldsymbol{C}^b_i\| \tag{1}$$

where $1 \leq i \leq k$ is the cluster index.

The relationship between the objects and the effects created by a given behavior is learned by a classifier using the data set:

$$\mathbf{T^b} = \{(\boldsymbol{p}_o, \textit{effect-id}_{\boldsymbol{\xi}^b_o})\}$$

where $\boldsymbol{p}_o$ is given as the input, and the corresponding *effect-id* as the target category. Specifically, in order to learn this mapping for each behavior b using this training set, Support

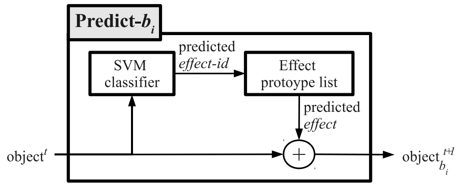

Fig. 2. The `Predict-` operator that is trained to predict the next state of an object based on the predicted effect of applying behavior b_i.

Vector Machine (SVM) classifiers with Radial Basis Function (RBF) kernels are used because they are robust in the face of noisy input and able to deal non-linear relations.

The trained SVM classifiers allow the robot to predict the type of *effect* a behavior is expected to generate when applied on a given object perceived as $\boldsymbol{p}_{o'}$ using:

$$\textit{effect-id}^{\text{predicted}}_{o'} = svmPredict(\boldsymbol{p}_{o'}, b).$$

The predicted percept of the object after the application of the behavior can then be computed as (see Fig. 2):

$$\boldsymbol{p}^{t+1}_{o',\{b\}} = \boldsymbol{p}^{t}_{o'} + \boldsymbol{C}^{b}_{\textit{effect-id}^{\text{predicted}}_{o'}}$$

IV. Planning Using Learned Affordance Relations

The learned affordance relations can be used as operators for planning.

a) States: A state is represented as the set of objects perceived or expected to be perceived after execution a number of behaviors in t steps:

$$S^{t}_{\{b^1\ldots b^{t-1}\}} = [\boldsymbol{p}^{t}_{o_1,\{b^1\ldots b^{t-1}\}} \cdots \boldsymbol{p}^{t}_{o_m,\{b^1\ldots b^{t-1}\}}]$$

where o_m corresponds to the m$^{\text{th}}$ perceived object, and $\boldsymbol{p}^{t}_{o_m,b^1\ldots b^{t-1}}$ is the expected percept for object m after execution of the behavior sequence $\{b^1\ldots b^{t-1}\}$.

b) Actions: The pre-coded behaviors; namely the five move behaviors and the lift behavior, constitute the actions. Different from standard techniques, the actions do not have any pre-conditions and their description does not include predefined state transition rules. All actions are applicable in all states, where the next state depends on the learned effect prediction operators summarized in Fig. 2.

$$S^{t}_{\{b^1\ldots b^{t-1}\}} \xrightarrow{b^t} S^{t+1}_{\{b^1\ldots b^{t}\}}$$

c) Goals: A goal is specified as a partial state, in terms of values of some object features within states. The user can define a goal based on feature values of any object, of a particular object or the combination of both. For example, the state that includes an object feature vector with $d_{min} < 0.1m$ will satisfy the goal of *approach any object*. As another example, the goal of *pick-up a particular object* is satisfied in a state, where the bottom-most row feature value of the corresponding object is large ($r_b > 180$) in the range image.

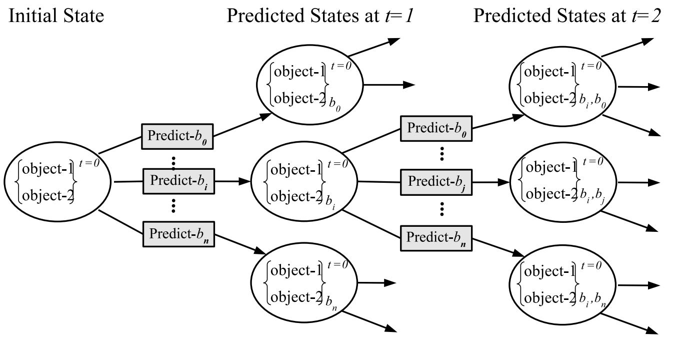

Fig. 3. The breadth-first construction of the plan tree. States include one or more objects whose next states are predicted based on the operators in Fig. 2.

d) Plan generation: Forward chaining is used to generate totally ordered plans starting from the initial state (see Fig. 3). This process can be viewed as the breadth-first construction of a plan tree where the branching factor is the number of behaviors. The next states are computed using the prediction operator in Fig. 2. If the state in any time step satisfies the goal, the sequence of the behaviors which lead the initial state to the goal is accepted as a potential plan.

V. Experiments

The learning experiments are conducted in a physics based simulator where the robot is verified against the real robot in [10]. One random object o (among ▢ , ○ , ▯ , ▭) is placed in $[-90°, +90°]$ of robot's frontal area, in a random orientation and size $[20cm - 40cm]$. The robot makes 3D scans before and after executing one of its behaviors (b) to compute the object ($\boldsymbol{p}_o$) and effect (($\boldsymbol{\xi}^{b}_{o}$) feature vectors. For the lifting behavior 1000 interactions are simulated, whereas for the move behaviors 3000 interactions are simulated. The resulting set of relation instances $\mathbf{I}$ are then used in training.

A. The learning of lift behavior

The set of effects ($\boldsymbol{\xi}^{lift}_{o}$) are split into two clusters using k-means. After clustering phase completed, each object in the training set is assigned to an *effect-id*, based on the class to which the created effect $\boldsymbol{\xi}^{lift}_{o}$ belongs to (Eqn. 1). Fig. 4 shows the effect classes of these entities together with the shape and position information for 2-cluster case. The objects assigned to class $+$ are the ones with flat top and close proximity to the robot. On the other hand, close objects with curved tops (spheres and lying cylinders) and all distant objects are assigned to a separate class ('$\bullet$'). Hence, we can conclude that the robot learned to distinguish successful lift actions.

After assigning each object to an effect-class, in order to learn the mapping between initial percept of the objects and the corresponding effects, an SVM classifier is trained. The parameters of SVM training with RBF kernel are optimized in a grid search using LibSvm software package. $c = 0.03$ and $\gamma = 32000$ are found as optimum cost of SVM and width of Gaussian respectively. After training is completed, prediction accuracy of the SVM model is tested on a distinct

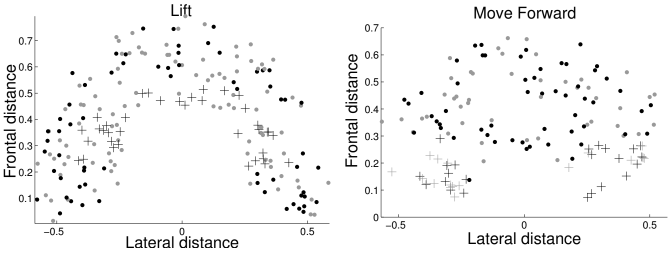

Fig. 4. Interpretation of effect classes obtained with unsupervised clustering for *lift* and *move-forward* behaviors. Each environment in interaction phase includes only one object, and each marker correponds to the location of the object in a different environment. Black markers represent boxes and upright cylinders; gray markers represent spheres and lying cylinders. • and + illustrates clustering results of the corresponding objects.

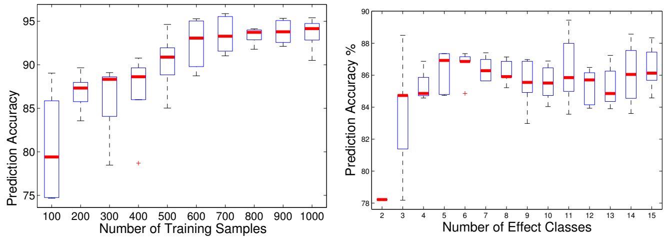

Fig. 5. **Left:** Performances of SVM classifiers trained with different number of samples in predicting lift effect classes (liftability). **Right:** The accuracy of liftability prediction in two-step planning. The boxes shows the distribution of prediction accuracy obtained in testing. The box is bounded by lower and upper quartile values, the line in the box refers to the median, and the whiskers show the extent of the data.

set of relation instances, where the result of lift behavior is also included as ground truth. Fig. 5-left plots the accuracy of the liftability prediction for different number of training samples. As results indicate, the SVM classifiers trained with more than 700 samples have performance over 90%. Additional training does not increase the performance.

B. The learning of move behaviors

The same type of learning is also applied to the data obtained from the five move behaviors. $c = 512$ and $\gamma = 0.125$ are found as optimum cost of SVM and width of Gaussian respectively. However, for simplicity, we will only discuss the results obtained from the move-forward behavior. Fig. 4 shows the effect categories for different objects together with the shape and position of those objects for 2 clusters. Independent of their shapes, objects located within $0.3m$ are within class $+$, and all distant objects are within class •. Additionally, some • objects in front of the robot are closer than $+$ objects. These observations show that the clustering process makes a distinction on effects based on whether the object disappears from the view of the robot or not.

Using the same training data, we varied the number of clusters being used to cluster the effect data, and measured the prediction accuracy . As the number of clusters increase, the clustering process also incorporates the differences in shapes of the objects. The accuracy of prediction of effect classes based on the objects are also examined and found to be over 90% for number of clusters upto 10.

C. Two-step planning

In this set of experiments, we evaluated the prediction accuracy of the robot to perceive the liftability of an object that are randomly placed within the $1m$ range of the robot. The robot applied the `Predict`-*move-forward* and `Predict`-*lift* operators (Fig. 2) to the initial percept of the object and using the final predicted percept of the object determine whether it's liftable or not. The predicted effect is then compared with the actual effect obtained by executing *move-forward* and *lift* behaviors.

Fig. 5-right plots the accuracy of liftability prediction for such two-step plans with respect to the number of effect clusters being used in the training of the *move-forward* behavior. The training set contained 3000 relation instances. Two points can be made. First, the average prediction accuracy of two-step plans for liftability (around 85%) is lower than the average prediction accuracy obtained from '1-step' plans. This is an expected result since as the objects get further away, the resolution of their perception degrades reducing the accuracy. Second, the number of effect clusters to be used in the training of the *move-forward* behavior should be greater than 2 to achieve a good prediction accuracy. This is probably due to the fact that the use of only 2 prototypes does not provide the necessary resolution to the *move-forward* behavior that can be propagated for planning.

D. Case study: Bringing an object on top of another

In all the experiments reported so far, a single object is presented to the robot during evaluation, as has been done during the training phase. In this experiment, we put the robot into an environment containing multiple objects and specify the goal as the conjunction of two conditions..

In this experiment, the robot is asked to lift an object and go towards a button (pre-defined object). The robot is free to select the object to lift. The goal is defined over desired future entities based on the predicted outcomes following the execution of planned action sequence. The goal for lift is to obtain an outcome for any object, where the bottom part of the predicted outcome range image should be high, ie. $r'_b > 180$ where r'_b is the 'bottom-row' feature of final feature vector. The goal for approach is defined as obtaining close proximity to a pre-defined object. The mean distance of the predicted final feature vector of that object, after the plan is executed should be small, ie. $d_{mean} < 0.1m$. Thus, the overall goal is to obtain an outcome of any object which satisfies lift goal and an outcome of the pre-defined object which satisfies approach goal, respectively.

A plan for an environment that includes 5 objects is presented in Fig. 6, where the robot is required to lift an(y) object and approach to another object (shown as a button). As shown, the generated plan is composed of three steps: $< move_{30}; lift(o_3); move_{-60} >$. We can make three observations. First, the robot is able to predict the liftability of object 3, before approaching. Moreover, the two cylindrical objects,

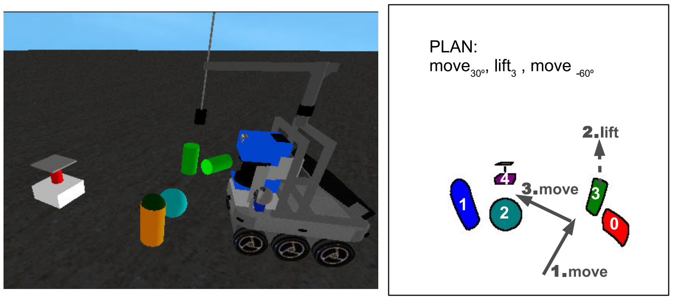

Fig. 6. The goal is to 'bring an object on top of the button'. On the left, 5 objects (including the button) are placed in the environment. On the right, the detected objects and the detected parts of the gripper arm in the range image are shown together with the generated plan. The object numbers in the range image are assigned automatically in the perception process, where button is numbered as 4. The robot cannot lift any object except the cylindrical shaped standing object on the right. However in order to lift it, the robot should approach to it by executing $move_{30°}$ action. After the $lift_3$ step, $move_{-60°}$ behavior is predicted to drive the robot towards the goal object.

one of which is lying on its side, and the other with a non-flat top, are correctly predicted to be non-liftable. Third, note that although object 1 is a cylinder with a round top, a novel object that was not used in training, the robot was able to predict that is was not liftable.

E. Case study: Novel objects in real-world

The plan generation is also tested in real world for liftability. The learned affordance relations and effect prediction methods are directly transferred to the real robot and its plan generation ability is tested. The environment contained six objects: A desktop world globe with a base, a box shaped power supply, an irregular hexagonal shaped metal piece, a triangular prism-shaped desk calendar, a can lying on top of another upright can (not visible), and an upside-down small pot as seen in Fig. 6. The segmented range image and the shortest plans to lift each of them (if there exists) is shown on the right hand side of the figure. The following plans are made. The pan is within the reach of the crane arm, and hence it can be directly lifted. The robot is required to move in order to pick-up the power supply or metallic piece. No plan is generated to lift the globe with base and the triangular prism since they did not have flat tops. However, it should be noted that the robot made an incorrect plan to lift the lying cylinder on its left. This is probably due to the fact that the robot predicted the effect of move$_{-60°}$ wrongly to achieve liftability[1].

VI. RELATED WORK

Learning of object affordances have been studied in [11] where the objects in the environment were differentiated using

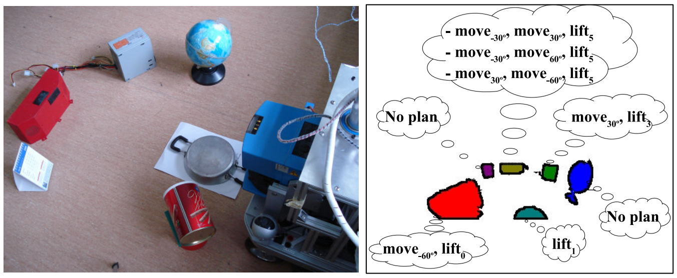
- move$_{-30°}$, move$_{30°}$, lift$_5$
- move$_{-30°}$, move$_{60°}$, lift$_5$
- move$_{30°}$, move$_{-60°}$, lift$_5$
No plan
move$_{30°}$, lift$_3$
No plan
move$_{-60°}$, lift$_0$
lift$_1$

Fig. 7. On the left, photograph of the environment where six real world objects are placed in front of the robot. On the right, the object regions detected in the range image are shown together with different plans generated for lifting. The bottom parts of the nearest two objects are not perceived and not seen in the range image since the laser beams are blocked by the robot base.

their colors only, and no learning was conducted to discover the real distinctive features of the objects. In another work, [12] proposed a general probabilistic model based on Bayesian networks to learn the relationship between actions, objects, and effects through interaction with environment. In [13] the robot learned tool affordances by building an adaptive hierarchical outcome model for each behavior based on the observed outcome. Although all these studies investigated the problem of predicting the future effects based on discovered affordance relations, none of them used learned affordances in planning.

Traditional AI planners work on a symbolic domain, but robot control and perception usually occurs on a much lower level. In order to bridge this gap [14] used self-organizing maps to cluster low-level perception data, and then mapped one perceptual state to the next outcome state. Predicting outcome (the next state) directly rather than predicting effect have some drawbacks on flexibility of the system. While there is no need to represent the non-changing features (for example shape) where the robot predicts only the effect in the environment, such features must also be taken into account in a system predicting the outcome. In another study, [15] proposed to learn and ground high-level domain structures as precondition and effects of robot actions. But in this study the high-level domain structures were pre-defined by the human programmer, rather than being generated through the experience of the robot. The concept of affordances have also been used in relation to planning, although these studies do not use an explicit formalism or representation for affordances. [16] makes a robot learn the *liftability* of an object based on the parameters of a pre-specified sequence of actions. But since the plan in this case was fixed, the learned knowledge could not be extended to generic sequence of actions. In [17], the robot learned goal-free action structures, in relation with the objects and the effects of these actions on the objects. Object concepts and these structures were then used in generating a sequence of actions in a goal-directed manner.

The studies summarized above can be grouped in two categories. In one group, the transition rules were defined as actions linked by logical precondition and postcondition predicates. Their approach is different from ours since sensorimotor experience of the robot was used to associate the predicates

[1] Due to a mechanical breakdown in our crane, we were not able to test these plans on the real robot. Although this is unfortunate, we do not believe that it undermines the validity of the results presented due to two reasons. First, the learning of initial percepts as well as the effects being produced, takes place on the range images, and that we used range images produced in real-world to test the learned relations. Second, by their very definition (as described in the last paragraph of the first page), the behaviors are assumed to be implemented in a closed-loop manner. Any failure (or success) in their execution would be due to their particular implementation and does not provide any implications for the learning method proposed in this paper.

of the transition rules. The other group learns the relation between objects, actions and effects independent of symbolic predicates, however these systems cannot predict more than one step ahead, which prohibits complex planning.

VII. DISCUSSION

In this paper, we proposed a method that allows a robot to learn symbolic relations that pertain to its interactions with the world and showed that they can be used in planning. Specifically, we used the notion of affordances, proposed in cognitive science, to develop a mapping from the sensory-motor experiences of the robot to symbols and relations that can be used in planning. We have shown that a mobile robot can learn the physical affordances of objects from range images and use them to build symbols and relations that can be used in making multi-step predictions about the affordances of objects and achieve complex goals. We followed a similar method to learn affordances provided by the objects to a robot hand in [18], where the camera and robot body was fixed and only manipulation related affordances were studied.

Although from its outset, one may feel that the work presented in this paper is merely Good-Old-Fashioned-AI, re inventing the STRIPS-like approach to planning, it's not. In STRIPS-like approaches, the operators appear in symbolic form, and these symbols are assumed to be manually grounded in the raw sensory data. This has been acknowledged as one of the major shortcomings of the STRIPS-like approach. The work presented in this paper, on the other hand, starts with raw sensory data and builds relations that can be used in generating plans. As discussed in the previous section, our approach is different from other learning approaches on this topic, in that it learns relations, as opposed to symbols, and also provides a means to using them in making plans.

In spite of the non-trivial learning and planning ability that our system exhibits, it has room for improvement. First, in a general setting the effects should be associated with action parameters. In case, for simplicity we defined multiple actions (e.g. move_{-30°, move_{-30°) for different instantiations of the same action with different parameters. One other issues is that the exploration in our system, is not guided by intrinsic motivations (e.g. curiosity) which are thought to guide infant exploration [19], [20]. Our earlier online learning study has also showed that such a strategy cuts down the exploration time [21]. Probably the most valuable improvement to our work would be to integrate the stochastic nature of robot-object interaction as bi-directional relations while being faithful to developmental stages and not sacrificing the planning ability demonstrated by our system. Lastly, the predictions on object features created during plan generation can be used to monitor the plan execution and to check whether the change in the state is as the one that was predicted in the plan, to decide whether execution was successful or not.

VIII. ACKNOWLEDGMENTS

This work was partially funded by the European Commission under the ROSSI project (FP7-216125). 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. We would like to acknowledge that the ideas presented in this paper partially shaped through discussions with Mehmet R. Dogar and Maya Çakmak.

REFERENCES

[1] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, pp. 335–346, 1990.

[2] V. Klingspor, K. Morik, and A. D. Rieger, "Learning concepts from sensor data of a mobile robot," *Machine Learning*, vol. 23, no. 2-3, pp. 305–332, 1996.

[3] R. Sun, "Symbol grounding: A new look at an old idea," *Philosophical Psychology*, vol. 13, no. 149–172, 2000.

[4] 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.

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

[6] J. Provasi, C. Dubon, and H. Bloch, "Do 9- and 12-month-olds learn means-ends relation by observing?" *Infant Behavior and Development*, vol. 24, no. 2, pp. 195–213, Feb. 2001.

[7] P. Willatts, "The Stage IV infant's solution of problems requiring the use of supports," *Infant Behavior and Development*, vol. 7, pp. 125–134, 1984.

[8] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, 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.

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

[10] E. Ugur and E. Şahin, "Traversability: A case study for learning and perceiving affordances in robots," *Adaptive Behavior*, vol. 18, no. 3-4, 2010.

[11] P. Fitzpatrick, G. Metta, L. Natale, A. Rao, and G. Sandini, "Learning about objects through action -initial steps towards artificial cognition," in *Proc. of ICRA 03*, 2003, pp. 3140–3145.

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

[13] J. Sinapov and A. Stoytchev, "Detecting the functional similarities between tools using a hierarchical representation of outcomes," in *7th IEEE International Conference on Development and Learning*, 2008.

[14] J. Pisokas and U. Nehmzow, "Experiments in subsymbolic action planning with mobile robots," in *International Symposium on Robotics and Automation*, 2002.

[15] C. Geib, K. Mourão, R. Petrick, N. Pugeault, M. Steedman, N. Krueger, and F. Wörgötter, "Object action complexes as an interface for planning and robot control," in *Workshop: Towards Cognitive Humanoid Robots at IEEE RAS Int Conf. Humanoid Robots*, 2006.

[16] S. Hart, R. A. Grupen, and D. Jensen, "A relational representation for procedural task knowledge," in *AAAI*, 2005, pp. 1280–1285.

[17] J. Modayil and B. Kuipers, "Autonomous development of a grounded object ontology by a learning robot," in *Proc. of AAAI*, 2007, pp. 1095–1101.

[18] E. Ugur, E. Sahin, and E. Oztop, "Affordance learning from range data for multi-step planning," in *Proc. of The 9th Int. Conf. on Epigenetic Robotics*, 2009, pp. 177–184.

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

[20] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic Motivation Systems for Autonomous Mental Development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[21] E. Ugur, M. Doğar, M. Çakmak, and E. Şahin, "Curiosity-driven learning of traversability affordance on a mobile robot," in *Proc. of IEEE Intl. Conf. on Development and Learning (ICDL 07)*, London, UK, 2007, pp. 13–18.