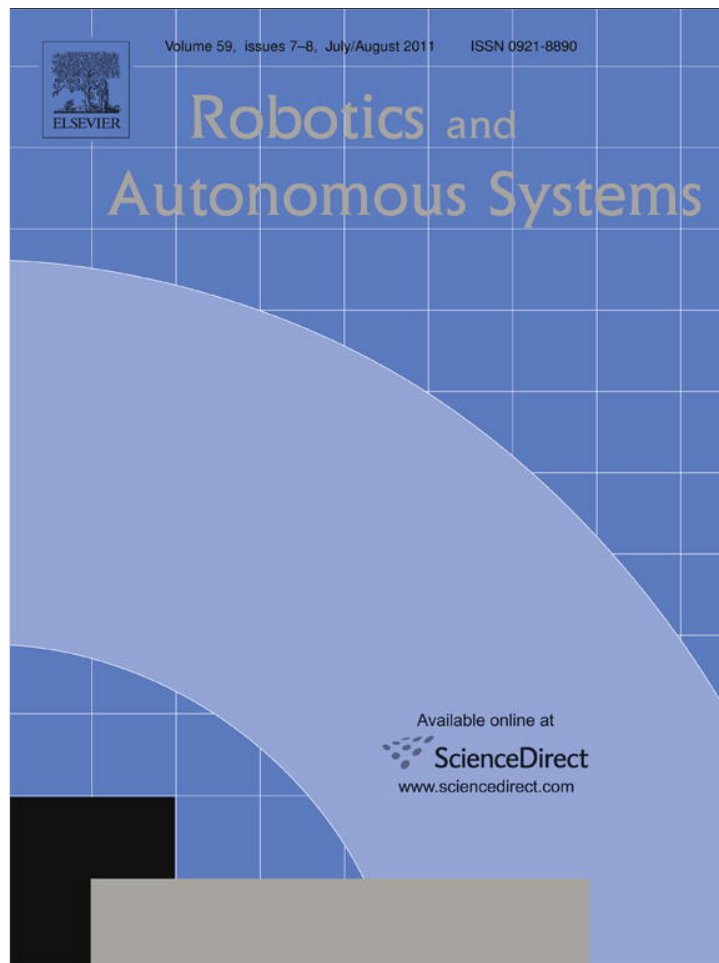


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Goal emulation and planning in perceptual space using learned affordances

Emre Ugur^{a,b,c,*}, Erhan Oztop^{a,b,d}, Erol Sahin^c^a NICT, Biological ICT Group, Kyoto, Japan^b ATR, Cognitive Mechanisms Labs., Kyoto, Japan^c Middle East Technical University, Department of Computer Engineering, KOVAN Research Lab., Ankara, Turkey^d Osaka University, School of Engineering Science, Osaka, Japan

ARTICLE INFO

Article history:

Received 13 August 2010

Received in revised form

24 January 2011

Accepted 9 April 2011

Available online 17 April 2011

Keywords:

Affordance

Developmental robotics

Sensorimotor learning

Manipulation

Perception

Cognitive robotics

ABSTRACT

In this paper, we show that through self-interaction and self-observation, an anthropomorphic robot equipped with a range camera can learn object affordances and use this knowledge for planning. In the first step of learning, the robot discovers commonalities in its action-effect experiences by discovering effect categories. Once the effect categories are discovered, in the second step, affordance predictors for each behavior are obtained by learning the mapping from the object features to the effect categories. After learning, the robot can make plans to achieve desired goals, emulate end states of demonstrated actions, monitor the plan execution and take corrective actions using the perceptual structures employed or discovered during learning. We argue that the learning system proposed shares crucial elements with the development of infants of 7–10 months age, who explore the environment and learn the dynamics of the objects through goal-free exploration. In addition, we discuss goal emulation and planning in relation to older infants with no symbolic inference capability and non-linguistic animals which utilize object affordances to make action plans.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

For a growing infant, a major problem is to make sense of the incoming sensorimotor data by learning what changes she can generate in the environment. Only after this problem is overcome, the infant starts making plans and executes them for achieving goals; for example, pulling the table cloth to reach a toy that is otherwise unreachable. It is plausible to think that earlier planning takes place in the perceptual domain of the infant, which is later augmented by symbolic planning capability as the infant forms symbolic representations through her interaction with the environment.

In this article, we consider the former phase of this developmental progression in a robotics context, where the robot learns its visuomotor capabilities by interacting with its environment. We are content that by adopting such a developmental approach, adaptive and human-like robotic systems can be synthesized. In the last decade, with similar views in mind, various developmental stages have been studied, modeled and transferred to

robots. These stages correspond to acquisition of skills at different levels and ages, ranging from the emergence of motor patterns before birth [1] and the development of pattern generators for crawling [2] to language learning [3] (see [4] for a comprehensive review).

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 [4]. 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. This process effectively corresponds to the learning of the affordances [5] provided by the environment. The learning in this stage is largely performed in a goal-free fashion through self-exploration and self-observation [6–9]. After approximately 9 months of age, the infant starts using the learned object-action-effect relations in a goal-directed way anticipating a desirable change in the environment and behaving accordingly [10–12]. This behavior ranges from recalling action-effect mappings to making simple plans that may involve multiple steps [13]. Goal emulation, a form of imitation characterized by the replication of the observed end effect [9], starts after this period, and infants become skilled at imitating unseen movements after 12

* Corresponding address: Department of Communication and Cognitive Cybernetics, ATR Cognitive Mechanisms Laboratories 2-2-2 Hikaridai, Seika-cho, Sorakugun, Kyoto 619-0288, Japan. Tel.: +81 774 95 2403; fax: +81 774 95 1236.

E-mail addresses: emre@atr.jp (E. Ugur), erhan@atr.jp (E. Oztop), erol@ceng.metu.edu.tr (E. Sahin).

months of age [14]. According to Elsner and Hommel [15], infants learn to use anticipation for goal-directed actions in two phases. In the first phase, they execute random actions in the environment, self-monitor the changes, and learn the action–effect associations in a bi-directional way. Later, in the second phase, they start to control their actions by predicting the effects they can create.

In a similar vein, in Stage 1, our manipulator robot experiences a goal-free self-exploration and self-monitoring phase where it discovers the affordances provided by the environment and learns how to use these affordances to predict the next perceptual state after the execution of a given behavior. In Stage 2, the robot emulates goals presented in its sensory space by generating multi-step plans based on the learned affordance and prediction capabilities. The computational elements of our system are derived from the affordance framework [16] that captures the active perception–action cycle of a learning agent from a computational point of view.

In the remainder of this section, we present a brief overview of the affordance concept, its computational interpretation, and its use in the field of robotics. The following section introduces the experimental platform and details on how the affordance framework was instantiated in our case, i.e. within a manipulator robot system. Sections 3–6 present the robotic analogues of the two phases of the aforementioned infant development process together with the learning and execution results obtained. Section 7 discusses our system in the light of human infant and chimpanzee studies, and presents potential improvements in a robotics context.

1.1. Affordances: a computational view

The concept of affordances was introduced by Gibson to explain how inherent “values” and “meanings” of things in the environment can be directly perceived and how this information can be linked to the action possibilities offered to the organism by the environment [5]. The definition of the term often depends on the field it is used in; in fact Gibson himself gave differing definitions over the course of his publications [17]. In general, the question ‘what does this mug afford for me?’ can be equated with ‘what type of actions can I apply on this mug?’. One clear fundamental notion of the affordance concept is that object recognition is not a necessary step for interacting with objects. That is, a specific combination of object properties with respect to the agent and its action capabilities are enough to detect the affordances of a given object (and act on it). Although it is not the classical engineering approach of identify and then act, this strategy appears to be the one employed by our brains. It is known that the cerebral cortex processes visual information in at least two channels, the so-called dorsal and ventral pathways. The ventral pathway appears to be responsible for object identification, whereas the dorsal pathway is mainly involved in perception for action [18–21]. These data suggest that an agent does not necessarily need to possess object recognition capabilities to learn about its environment, and use this knowledge for making plans.

Placing the concept of affordance on a general computational ground is difficult due to its elusive and multi-facet nature. Recently, Şahin et al. [16] have proposed a computational interpretation of the affordance concept that was shown to be effective for mobile robot control [22–24]. In this study, we adopt this framework, and build upon our preliminary work in robot hand control [24]. One key feature of this framework is that the affordances are defined from the viewpoint of the acting agent (see Şahin et al. [16] for alternatives). An affordance is a learned relation between an effect obtained when a behavior is applied to an entity. Hence, affordance is the relation between the pair (entity, behavior) and the (effect). For instance, the liftability

affordance is represented as the existence of at least one relation between the lifted effect,¹ an object and a behavior of the robot. ‘Entity’ denotes the environmental relation obtained via perception of the environment and self. Entity is a high-level term that can encapsulate the perceptual representation of an agent at different complexity levels, ranging from raw sensory data to the features extracted from the environment. Behavior represents the physical embodiment of the agents interaction. It is an internal representation that defines a unit of action that can often take parameters for the initiation and online control. As in the entity definition, the level of complexity is not a part of the definition; therefore, a simple joint rotation, as well as a grasping action directed to an object can be considered as behaviors. Finally, an effect is defined as the change generated in the environment due to a behavior execution.

In this work, the entities are encoded as continuous valued feature vectors representing the objects that the agent can interact with. In other words, the environment consists of only the objects, excluding the surrounding and the state of the robot itself. The behaviors are represented as open-loop control units with multiple parameters (e.g. push-forward (x, y, z)). The effects are taken as the vectorial differences between the entity representations before and after behavior execution. With this setting, in order to learn the affordances, our robot goes into interactions with the environment using its actuation capabilities, and monitors the environment and the changes that were created in it. In each interaction, the initial perception of the environment generates a feature vector (f_o) corresponding to an entity before behavior execution. Then the robot executes one of its behaviors, b_i , in its repertoire. After the execution of the behavior, the robot, once more, perceives the environment and obtains the final feature vector. Finally, effect, e_o , created is found by taking the difference between final and initial feature vectors.

1.2. Related work

During the recent years, studies inspired by ideas in developmental psychology have increased considerably [25,26,4,27]. These studies typically use exploration, learning and embodiment to enable robots learn about their environment via exploration with minimal expert knowledge. In the rest of this section, we review related studies (see Table 1 for a summary) and discuss our contribution to the field.

The pioneering studies that used motor babbling as a means of exploration for learning of affordances include Metta and Fitzpatrick [28], Fitzpatrick et al. [29] and Stoytchev [30]. For example, Fitzpatrick et al. [29] studied the learning of rollability affordance by executing different actions on different objects and observing their effects. Stoytchev [30] investigated tool affordances by discovering tool–behavior pairs that give the desired effects. In these studies, the learning of the association between the visual features of the objects/tools and their effects (i.e. affordances) were not addressed; therefore, the learned affordance knowledge could not be generalized to novel objects/tools.

Later this issue was addressed by Ugur and Şahin [22], Erdemir et al. [31], and Fritz et al. [32], where the relations between the visual features of entities and their affordances were learned. However, in these studies the target affordance categories (e.g. liftability and traversability) were pre-defined, and learning was performed in a supervised way based on the success criteria defined over the effect of each action. Thus, the affordances were

¹ Note that ‘lifted effect’ is a label used to describe the sensory change here; in the agent’s world this just corresponds to an internal representation which is not necessarily assigned to any label.

Table 1
Summary of the related robotics literature. The *Fixed objects* group does not use to object properties. The *Supervised* and *Unsupervised* groups learn one-directional mapping from object/environment properties to given and discovered effect categories, respectively. The *Probabilistic* group learns the probabilistic bi-directional relations and the *High-level rules* group assumes the existence of AI inference rules. The *Planning in perceptual space* group's learning is similar to the *Unsupervised* group, but can make multi-step predictions. *Planning in sensory-motor (SM) space* corresponds to the neuro-dynamical system where the robot learns in sensorimotor space without utilizing object affordances. Initial, effect, and final can encode visual perception or proprioception of the robot.

		Initial percept representation	Effect/final percept representation	Learning	Learning method
Fixed objects	Fitzpatrick et al. [29] Stoytchev [30]	Pre-defined-ids	Continuous Binary	initial → effect	Probabilistic inference Table-lookup
Supervised	Ugur and Şahin [22] Erdemir et al. [31] Fritz et al. [32]	Continuous	Binary	initial → effect	SVM GMM Decision tree
Unsupervised	Sinapov and Stoytchev [33] Griffith et al. [34]	Continuous	Discrete (<i>X</i> -means)	initial → effect	Decision tree
	Cos-Aguilera et al. [35]	Discrete (GWR)	Discrete (GWR)		
Probabilistic	Demiris and Dearden [36] Montesano et al. [38] Hart et al. [37]	Discrete (<i>X</i> -means) Continuous	Continuous Discrete (<i>X</i> -means) Binary	(initial) ↔ (action) ↔ effect	Bayesian Belief Nets. Bayesian Nets. Relational Dep. Nets.
	High-level rules	Petrick et al. [39] Wörgötter et al. [40] Modayil and Kuipers [41]	Binary Continuous Continuous		
Planning in SM space	Arie et al. [42]	Continuous	Continuous	initial → final	Recurrent Neural Nets.
Planning in perceptual space	Pisokas and Nehmzow [43]	Discrete (SOM)	Discrete (SOM)	initial → final effect	List of links
	Current work	Continuous	Discrete (<i>X</i> -means)		SVM

not discovered by pure exploration and the robot only learned to predict the effects designed by the programmer.

Sinapov and Stoytchev [33], Griffith et al. [34], and Cos-Aguilera et al. [35] proposed the self-discovery of the affordances, where the effect categories were found through unsupervised clustering in the effect space. Furthermore, using these categories, the mappings of *object* → *effect categories* were learned. Thus, the robot was able to make predictions to choose actions that would fulfill a desired environment change.

All the aforementioned studies were deterministic and relied on one-directional mappings. Demiris and Dearden [36], Hart et al. [37], and Montesano et al. [38] used probabilistic networks that capture the stochastic relations between objects, actions and effects. These networks allowed bi-directional relation learning and prediction. For example in [38], after training Bayesian networks, the robot could predict the object categories when effects and actions were given, or it could predict the effect categories when objects and actions were given. One drawback of this approach was that the object categories were created by unsupervised clustering in feature space without any reference to the interaction experience of the robot. Cognitive development in humans suggest that actions and the effects created by them are used together to parse the perceptual space into categories that may be called 'objects' (entities). Therefore, it is not the case that the object categories exist in the environment and their relations with the effects and actions are learned; but rather the effects and actions define the object categories. In our work, we will follow this full action based perception view by categorizing the object feature space based on the effects. This is central to the affordance concept.

In all of the studies mentioned above, the agent acquires the ability to make predictions about the effects it can create through active exploration of the environment. However, due to the 'effect' representation adopted in these studies, the systems described cannot predict more than one step ahead, which prohibits complex planning.

In [39–41] on the other hand, after learning, the robots could make multi-step predictions using transition rules and hence

were able to demonstrate complex planning. The transition rules were defined as actions linked by logical pre-condition and post-condition predicates. This approach is different from the previous ones since sensorimotor experience of the robot was used to associate the predicates of the transition rules. These conditions were pre-defined binary functions of sensor readings in [40], where the robot learned to combine these conditions in the form of pre-conditions and effects through human assistance. Petrick et al. [39] used pre-defined or pre-learned high-level object and environment properties as the predicates of the transition rules. On the other hand, Modayil and Kuipers [41] discovered these predicates from low-level sensory readings during a goal-free exploration and learning phase. Although objects could be categorized based on their shapes in the sensory level, this information was not used in effect prediction. Moreover, only position features were used to learn "simple affordances of the object" [41, p. 886]. In short, in these approaches, the learned affordances were either simple or acquired through supervision. In addition, the mapping of these architectures to developmental psychology is not straightforward as logical inference mechanisms were assumed to be available to the learning agent.

A biologically plausible model was proposed in [42], where the robot was able to plan novel continuous motions that corresponded to multi-step behavior patterns that were observed during training. A multi-component recursive neural network was used to learn the object-robot dynamics and generate plans in the continuous sensory-motor space. This approach had no prior assumptions in terms of perceptual and behavior representation since raw retinal image and arm joint angles were used in system training and generation of novel actions. However, this system was trained using only one object, and did not focus on affordance learning.

In our study, we will follow a similar approach to the studies [33–35] that were discussed above (*Unsupervised* group in Table 1). The main novelty of our approach is the encoding of the effects and objects in the same feature space. In contrast, in the other studies

the effect representation were context and task dependent, and therefore did not correspond to the object feature space. Having the effects and objects encoded in the same space will provide the ability to predict the next perceptual state by adding the current features to the predicted effect features. This will enable the robot to make plans (without using high-level AI rule techniques) based on the structures that are learned in a completely bottom-up manner during its interaction with the environment.

From the planning viewpoint, Pisokas and Nehmzow [43] can be considered as the closest to our approach, where the robot learned the environment dynamics in its perceptual space and made multi-step action plans to achieve goals in a locomotion task. However, there are important differences that sets our work apart. In [43], first, the initial percept space was categorized in an unsupervised manner, i.e. irrespective of the interaction experience of the robot (as was the case for [38]). Second, the robot learned the *initial* \rightarrow *final* mapping, whereas our system learns the *initial* \rightarrow *effect* mapping which provides better generalization. For example, in our case, pushing a box located on the table will always generate the same effect regardless of its position (unless, of course the object is at the edge). Yet, at the same time, it will be possible to obtain the final percept (i.e. the predicted position of the box). Generalization of the knowledge obtained via exploration is a critical issue when the world of the agent becomes more complex, i.e. when the number of actions and the type of environments that can be experienced becomes large. An adaptive agent needs to utilize its resources parsimoniously, and needs to be able to predict in situations that it never encountered before.

In summary, the points that set our work apart from the existing ones are (1) multi-step planning, (2) categorization of the perceptual space based on actions and their effects, (3) generalization of the knowledge obtained through exploration.

2. System realization

2.1. Robot system

An anthropomorphic robotic system, equipped with a range camera, and its physics-based simulator is used as the experimental platform (Fig. 1). The robot platform consists of a five fingered 16 DOF robot hand² and a 7 DOF robot arm.³ For robot perception, an infrared range camera,⁴ 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 greyscale image of the scene and a confidence value for each pixel.

The simulator (Fig. 2(a)), 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. For each ray, the first contact with any surface is retrieved using ODE functions, distance between the contact point and ray origin point is used as the range value, and a Gaussian noise with $\mu = 0$, $\sigma^2 = 0.2$ is added to account for camera noise.

2.2. Perception

Object detection. The first step of pre-processing is to filter out the pixels whose confidence values are below an empirically selected

threshold value. The robot's workspace consists of a black table, so region of interest is defined as the volume over the table, and black pixels are filtered out as the range readings from black surfaces are noisy. As a result, the remaining pixels of the range image are taken as belonging to one or more objects. These objects are segmented by the Connected Component Labeling algorithm [44] which differentiates object regions that are spatially separated by a preset threshold value (2 cm in the current implementation). In order to reduce the effect of camera noise, the pixels at the boundary of the object are removed, and Median and Gaussian filters with 5×5 window sizes are applied. The detected objects on the range image of a sample setup is shown in Fig. 1(b). Finally, a feature vector for each object is computed using the 3D positions obtained from depth values of the corresponding object pixels as detailed in the next paragraph.

Algorithm 1 Object Detection

isConfident(p): true if confidence[p] \geq confidence-threshold
 isOnTable(p): true if position[p] is on table
 isBright(p): true if amplitude[p] \geq amplitude-threshold
 setObjectPart(p): pixel p is assigned as object part

- 1: **for** each pixel p (from 0 to 174×144) **do**
 - 2: **if** (isConfident(p)) **and** (isOnTable(p)) **and** (isBright(p)) **then**
 - 3: setObjectPart(p)
 - 4: **end if**
 - 5: **end for**
 - 6: Find distinct objects with Connected Component Labeling
 - 7: Remove pixels on object boundaries
 - 8: Apply Median and Gaussian filters to object pixels
-

Object feature vector computation. The perception of the robot at time t is denoted as $[\mathbf{f}_{o_0}^{t,0}, \mathbf{f}_{o_1}^{t,0} \dots]^5$ where \mathbf{f} is a feature vector of size 43, and the superscript ⁰ denotes that no behavior has been executed on the object yet. Three channels of information are gathered and encoded in a feature vector for each object o_i (Fig. 2(b)). The first channel consists of *object-visibility* feature which encodes the knowledge regarding the existence of the object. The second channel corresponds to the distance perception of object's borders. Here, the points with minimum and maximum values along longitudinal, lateral and vertical axes are used as 6 position-related features. The third channel encodes the shape-related features, where the distribution of the local surface normal vectors are used. Specifically, histograms of normal vector angles along the latitude and longitude are computed and used as follows.

The normal vector of the local surface around each point is calculated using the positions of the two neighbors in the range image:

$$\mathbf{N}_{r,c} = (\mathbf{p}_{r-n,c} - \mathbf{p}_{r,c}) \times (\mathbf{p}_{r,c-n} - \mathbf{p}_{r,c})$$

where \mathbf{p} represents 3D position, n corresponds to the neighbor pixel distance and is here set to 5. In spherical coordinates, the unit length 3D normal vector is represented by two angles, polar (θ) and azimuthal (φ) angles that encode information along latitude and longitude, respectively. The polar angle (θ) corresponds to the angle between x - z plane and the normal vector, whereas φ is the angle between z -axis and the normal vector's orthogonal projection on x - z plane. After polar and azimuthal angles are computed for each pixel, two histograms are computed in θ and φ using a 20° bin size. Finally, the angular histograms represent the 36 shape-related features.

² Gifu Hand III, Dainichi Co. Ltd., Japan. <http://www.kk-dainichi.co.jp/e/gifuhand.html>.

³ PA-10, Mitsubishi Heavy Industries.

⁴ SwissRanger SR-4000 <http://www.mesa-imaging.ch/>.

⁵ Note that t and o_i are sometimes omitted in the rest of the text in order to ensure easy readability of the notation.

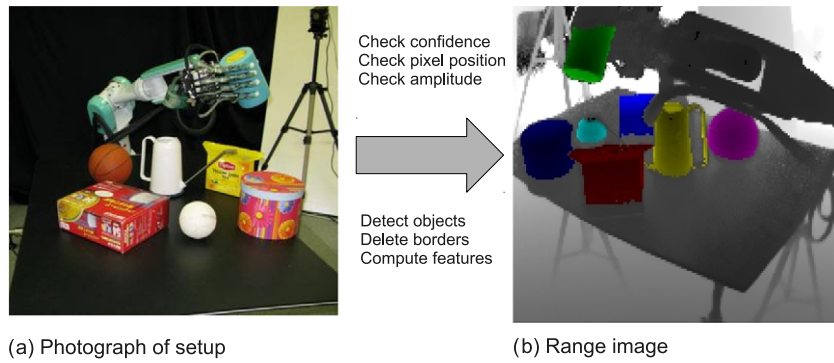


Fig. 1. In (a), the 23 DOF hand-arm robotic platform, infrared range camera (on the top-right) and the objects that are used in this study are shown. In (b), the range image obtained from the range camera and the detected objects are shown where the range is encoded in greyscale and in color for the environment and objects, respectively.

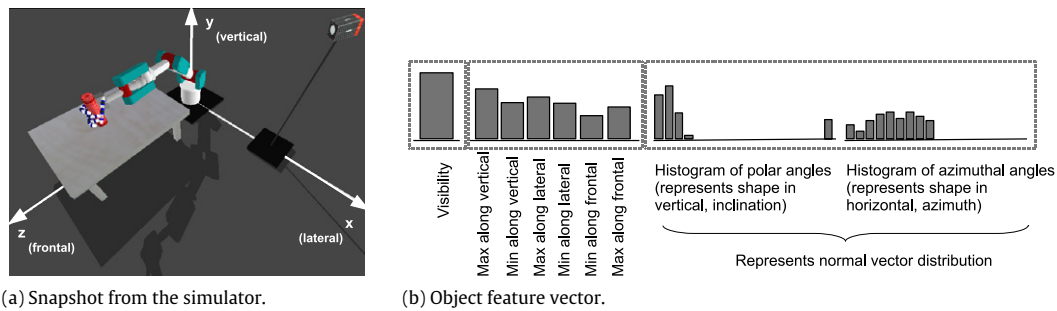


Fig. 2. (a) The robot arm grasps and lifts a cylindrical object in the physics-based simulator. The coordinate system is also illustrated. (b) The 43-dimension feature vector computed for the object in the robot's hand in Fig. 1 is given. It is composed of 1 visibility, 6 positions and 36 shape-related features whose values correspond to the height of the bars in normalized form.

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

$$f_{effect, o_i}^{(b_j)} = f_{o_i}^{(b_j)} - f_{o_i}^{(0)}$$

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

2.3. Interaction

The robot interacts with the objects using three *push* behaviors and one *lift* behavior. The object position computed from the range camera is used as argument for the behaviors to enable the robot interact with objects placed in different positions. The hand is initially wide-open for all behaviors, is clenched into a fist during *push-forward* execution, and remains open for the other *push* behaviors. For *push-forward*, *push-left*, and *push-right* behaviors the robot hand is brought to the rear, right and left side of the object, respectively. Then, the hand moves toward the object center, pushing the object in the appropriate direction. After behavior execution, the hand is placed to a 'home' position. In the *lift* behavior, the robot hand is placed at the back-right diagonal of the object first, then moved toward the object while the fingers are closed to grasp the object. After the fingers come to a halt, the hand is lifted vertically.

The robot interacts with three types of objects: boxes, cylinders and spheres of different sizes and orientation. During the execution of *push* behaviors, the robot observes the consequences of its actions. For instance, when the robot pushes a box (☐) or an upright cylinder (⊞), the object is dragged during the execution of the behavior and stand still at the end of the action. However, when the robot pushes a sphere (⊙), the object rolls away and falls down the table. The *lift* behavior would succeed in lifting an object, if the object is within the arm length of the robot and small

enough to fit into the robot hand. However the consequences of the *lift* behavior execution is not limited to having lifted the objects and can be complex. For example, some spheres may roll out of the view after an attempt to grasp and lift, while large boxes will be pushed away but still remain in the view after the *lift* behavior execution.

Algorithm 2 Exploration phase

- 1: **for** each trial k (from 1 to m) **do**
- 2: Reset robot joint angles
- 3: Put a random object in random position, size, and orientation
- 4: **while** isObjectVisible() **and** isObjectPositionChanged() **do**
- 5: Perceive the environment and compute initial feature vector $f^{(0)}$
- 6: Execute a random behavior b_i where $0 \leq i \leq 3$
- 7: Perceive the environment and compute effect feature vector $f_{effect}^{b_i}$
- 8: Put $(f_{effect}^{b_i}, f^{(0)}, b_i)$ into repository.
- 9: **end while**
- 10: **end for**

2.4. Exploration

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 $(f_{effect}^{b_i}, f^{(0)}, b_i)$ tuples, i.e. (object, effect, behavior) instances (Algorithm 2). Here, b_i is the behavior used for interaction, $f^{(0)}$ and $f_{effect}^{b_i}$ denote the initial object feature vector and the difference between final and initial feature vectors, respectively. Note that o_j notation is omitted since the robot interacts with a single object at a time in this phase.

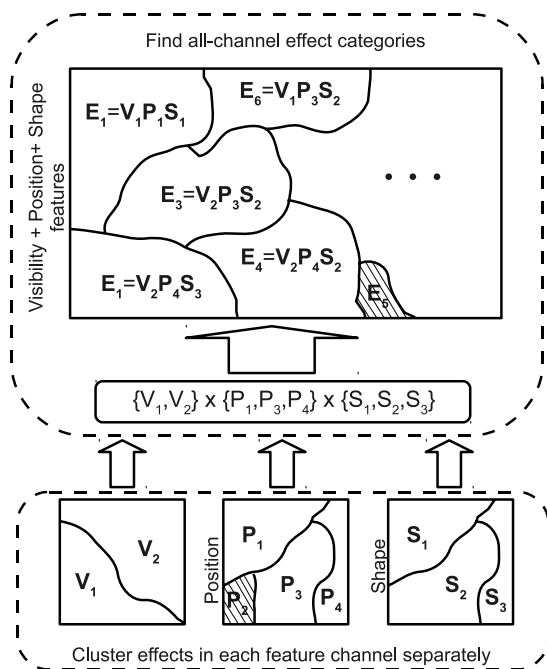


Fig. 3. The proposed hierarchical clustering method to discover effect categories. Channel-specific and all-channel effect categories are shown on lower and upper levels, respectively.

3. Stage 1: learning affordances

The data collected as tuples during the exploration phase are stored in a repository

$$\{(\mathbf{f}_{effect}^{bj}, \mathbf{f}^0, b_i)\}$$

and is used by the robot to learn the affordances of objects. 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 (Fig. 3). In the lower level, channel-specific effect categories are found by clustering in the space of each channel, discovering separate categories for visibility, position and shape. In the upper level, the channel-specific effect categories are combined to obtain all-channel effect categories using the Cartesian product operation. In Fig. 3, where a hypothetical example is depicted, the effect category $E_1 = V_1P_1S_1$ stands for $E_1 = V_1 \wedge P_1 \wedge S_1$ and contains the effect feature vector instances which are classified as $V_1, P_1,$ and S_1 when only the corresponding feature channel is considered, respectively. Finally, the effect categories that occur rarely (indicated in the figure as shaded regions) are automatically discarded together with their members. The proposed hierarchical clustering method is superior to simple one-level clustering method, since the result of one-level clustering is sensitive to the relative weighting of the effect features that are encoded in different units (e.g. continuous position features vs. binary visibility feature). Additionally, the performance of the clustering process is optimized by running the clustering algorithm multiple times and selecting the best clusters based on their utility in the second step of learning.

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

the robot in a more compact way, the continuous effect vectors are replaced by the effect category id's and their prototypes. The repository is thus transformed into the following form:

$$\{E_{id}^{bj}, \mathbf{f}^0, b_j\}, \{(E_{id}^{bj}, \mathbf{f}_{prototype, id}^{bj})\}.$$

Here, the first list corresponds to the set of affordance relation instances where effects are generalized and the second one corresponds to the list of (effect category id, prototype vector) pairs.

Learning effect category prediction. In the second step, classifiers are trained to predict the effect category for a given object feature vector and a behavior by learning the $\mathbf{f}^0 \rightarrow E_{id}^{bj}$ mapping. Effectively, this establishes a forward model, $Predictor^{bj}(\mathbf{f}^0)$ that returns E_{id}^{bj} for each behavior.

At the end of these two learning steps, affordance relations are encoded as:

$$\{Predictor^{bj}()\}, \{(E_{id}^{bj}, \mathbf{f}_{prototype, id}^{bj})\}$$

or

$$\{\{Predictor()\}, \{(E_{id}, \mathbf{f}_{prototype, id})\}^{bj}$$

allowing the robot to 'know' the effect of a behavior in terms of the effect category and its prototype.

4. Stage 1: results

In the experiments, a table with $100 \times 70 \text{ cm}^2$ surface area was placed in front of the robot with 40 cm distance, as shown in Fig. 1. At the beginning of each exploration trial, one random object (\square , \ominus , or \square) of random size (20–40 cm) was placed on the table at random orientation (see Algorithm 2). For all behaviors, 5000 interactions were simulated and the resulting set of relation instances were used in learning. The X-means algorithm [45] was used to find channel-specific effect categories,⁶ and Support Vector Machine (SVM) [46] classifiers were employed to learn effect category prediction.

In the rest of the section, the effect categories that were discovered using the proposed hierarchical clustering algorithm are interpreted, and the contributions of specific object features for affordance prediction are assessed.

4.1. Discovered effect categories for push behaviors

The detailed results are given for only one representative behavior, *push-right*, as all the *push* behaviors produced similar effect categories. The channel-specific effect categories discovered for the *push-right* behavior and their prototypes are shown in Fig. 4. Two categories are discovered within the visibility channel. The first category corresponds to the disappearance of the object (indicated by a change of -1 on the visibility feature) and the second category represents the effect where the object remains in the view (indicated by no change).

The changes in object position channel are represented by four distinct effect categories. The first category represents the case for no change in object position, and the third and fourth categories represent different magnitudes of object movement. The occurrence of the second effect category is very rare, i.e. the ratio of the members in this category to whole sample set is below a preset threshold (of 3%); hence this category is discarded. In the shape channel, four effect categories are discovered but one

⁶ X-means implementation in Weka data mining software is used. <http://www.cs.waikato.ac.nz/ml/weka/>.

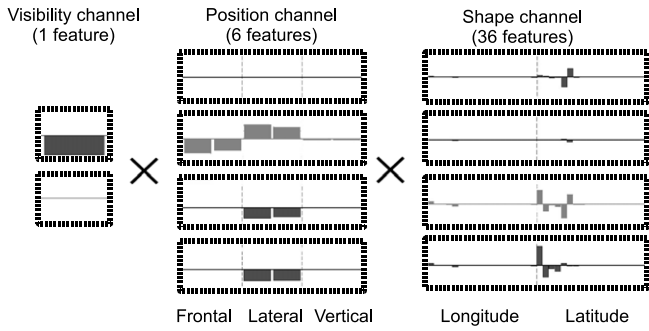


Fig. 4. Each dashed box corresponds to a channel-specific effect category discovered for the *push-right* behavior. The category prototype vectors are represented by the bars in normalized form. For example, the position of the objects, which created the last effect category in position channel, reduced along lateral axis and did not change along other axes. The light colored prototypes are discarded since the number of members was below the threshold.

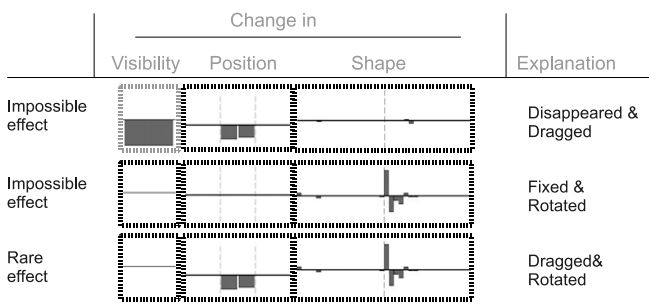


Fig. 5. Impossible or rare effect categories that are formed through Cartesian product of channel-specific categories for push-right behavior. Some of the categories can be created due to inaccuracies in simulator and some of them do occur very rarely.

of them (third category) is discarded as its ratio was below the threshold.

The all-channel effect categories are computed by taking the Cartesian product of the channel-specific effect categories. The 2 categories in the visibility and 3 categories in both the position and shape channels generate $2 \times 3 \times 3 = 18$ all-channel categories. A pruning process is applied as in the lower level, to remove the impossible and rare effects based on the number of category members.

Fig. 5 shows some of such categories that are obtained due to rare occurrence in robot's experience. The first illustrated category is physically impossible because the object disappears according to the visibility feature, and at the same time moves to a visible position based on the position feature. In the second category, object's position is not changed but it is rotated around. This is also impossible unless the object is attached to the table, which is not the case in our setup. The third category, where the object is pushed to the right and rotated, is possible but rare, as the objects are pushed from the center.

Next, we analyze the prototypes of remaining effect categories (Fig. 6).

- The *unreachable effect* (Effect-2) corresponds to the prototype where no feature change is observed. The average distance of the objects that produced an *unreachable effect* is 124.4 cm indicating their *unreachability*, given the kinematics of our robot.
- In the *disappear effect* (Effect-1), the visibility of objects drop from 1 to 0, indicating the objects falling off the table. This can happen when the objects are pushed and rolled out of the table. We found that most of the objects that fall under this category are spheres, since they are likely to roll away and fall from the table. However, boxes and upright cylinders placed on

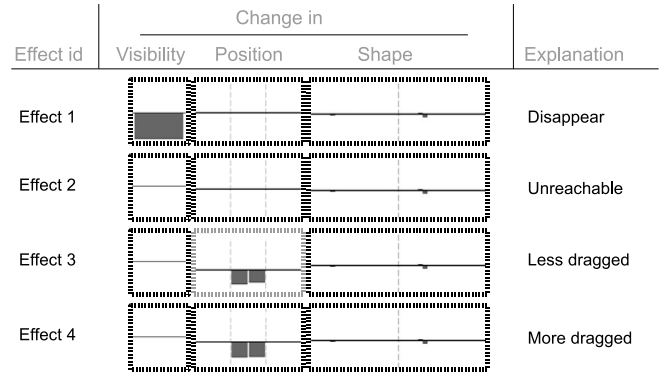


Fig. 6. The prototypes of effect categories for push-right behavior. It can be seen that *push-right* has an effect on visibility and lateral position features but not on others.

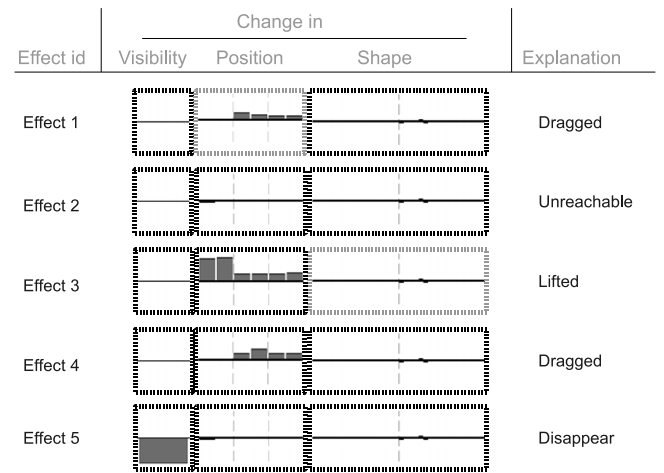


Fig. 7. The prototypes of effect categories for lift behavior. It can be seen that *lift* behavior has an effect on vertical, frontal and lateral position features as well as visibility.

the edge of the table also fall under this category as they fall off the table when pushed. The *disappear effect* (Effect-1) was also created by the objects which were elevated over the table. This happens when the robot had executed a successful *lift* behavior in the previous step. In such situations, a subsequent *push-right* behavior would open the hand causing the lifted object to drop and hence might make it disappear. Note that the disappearance of an object through dropping it (*lift* followed by *push-right*) was an unexpected emergent behavior.

- In the *less-dragged effect* (Effect-3) and the *more-dragged effect* (Effect-4), the lateral position of the objects are reduced (the objects are pushed right with respect to the robot) as a result of *push-right* behavior. These categories were created by only boxes and upright cylinders, and do not include any spheres since they always roll away when pushed.

4.2. Discovered effect categories for lift behavior

Fig. 7 shows the all-channel effect prototypes, discovered by the hierarchical clustering process for the *lift* behavior.

- The *unreachable effect* (Effect-2) corresponds to no significant changes in the feature vector since it was created by (failed) interaction with unreachable objects, similar to Effect-2 for the *push-right* behavior.
- In the *disappear effect* (Effect-5), objects became invisible after the execution of *lift* behavior. This effect was created by (1) ungraspable large spherical objects that roll away after

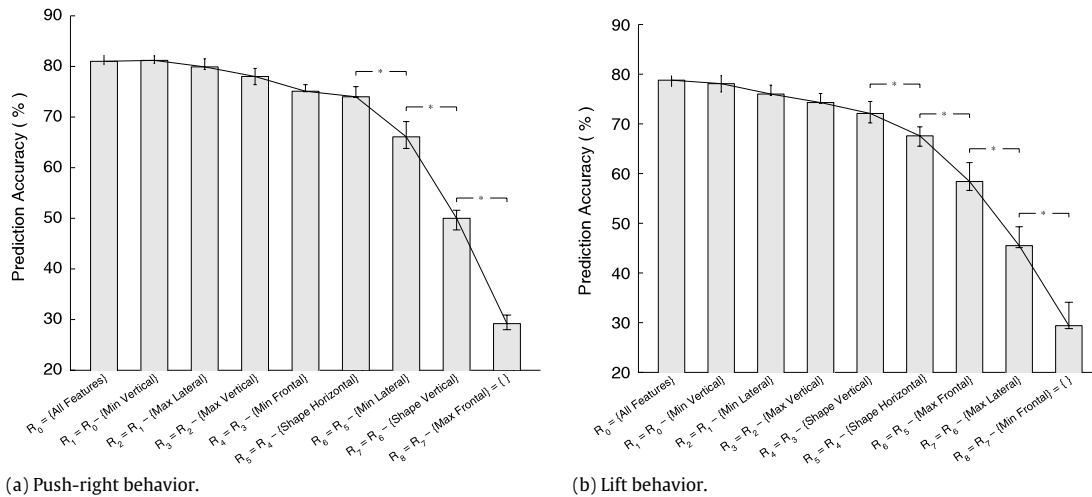


Fig. 8. The prediction accuracies of the classifiers that are computed using different feature sets. The feature set is reduced by one feature in each iteration by deleting the most irrelevant one. The left-most and right-most bars in each plot show the results obtained using all and no features, respectively. Error bars on prediction accuracies indicate the best, median, and worst classifiers found by 5-fold cross-validation. Significant changes between adjacent feature subsets from *t*-test are shown (* : $p < 0.002$).

interaction, (2) ungraspable large objects that are pushed off from the left edge of the table, and (3) the objects that were already in robot's hand due to a previous *lift* behavior execution.

- In the *dragged effects* (Effect-1 and Effect-4), the vertical position of the object remains same, but its position on the table is changed indicating a drag over the table. This effect was created by large ungraspable objects that are not rollable. The objects that create *dragged effects* were pushed on the table for different amounts and in different directions as interactions with different object types and sizes result in different collision dynamics between the hand and object.
- In the *lifted effect* (Effect-3), the elevation of the objects (represented by first two columns) increase, corresponding to the cases where objects were successfully grasped and lifted.

The *lift* behavior was designed to grasp and lift the objects. Thus, from the designer's point of view, there can be two different outcomes resulting from the execution of the lift behavior: Either the object can be grasped and lifted successfully, or it cannot be grasped, so it cannot be lifted. However, when the effects that were obtained during the robot's exploration were clustered using the hierarchical clustering algorithm, five different effect categories were generated. These results show that the effect categories should not be limited to the definition of the behavior or the intention of the behavior designer, but should be discovered through interaction.

4.3. Effect category prediction results

After the discovery of effect categories, the mapping from the initial object features to these categories is learned for each behavior b_j (Predictor $^{b_j}()$) by multi-class Support Vector Machines (SVMs). The Libsvm⁷ software package was used with optimized parameters of the RBF kernel through cross-validated grid search in parameter space. 4000 simulated interactions were used in training and a separate set of simulated 1000 interactions were used for testing. At the end, 95%, 84.3%, 82.2%, and 79.7% accuracies were obtained in predicting the correct effect categories for *push-forward*, *push-left*, *push-right*, and *lift* behaviors, respectively. The accuracy of *push-forward* is higher than other push behaviors since

it has three effect categories (compared to four effect categories in other two *push* behaviors).

We analyzed the relevance of the features in affordance prediction for the *push-right* and *lift* behaviors using the Schemata Search [47] by computing 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 whole feature set (R_0), and reduces it by removing the least relevant feature in each iteration. At each iteration (t), candidate subsets are formed by removing a different feature from R_{t-1} (remaining feature set of previous iteration), and they are evaluated by training SVM classifiers in 5-fold cross-validation. The subset with the highest mean prediction accuracy is chosen as R_t and transferred to the next iteration. The computation time is reduced by grouping the vertical (longitude) and horizontal (latitude) shape features, and treating them as single units.

Fig. 8 shows the prediction accuracies of the feature sets produced by this method. In both plots, the first bar corresponds to the prediction accuracy with the full feature set (R_0) and the last bar corresponds to the accuracy without use of any features ($R_8 = \{\}$, base condition).

The effects of these features were further investigated by performing *t*-tests contrasting the prediction accuracies of adjacent feature subsets. We found that the prediction accuracy changed significantly after the removal of features from the subsets R_5 and R_4 for *push-right* and *lift* behaviors, respectively.

- For the *push-right* behavior (Fig. 8(a)), the three most relevant features were *Min. Lateral*, *Shape Vertical*, and *Max. Frontal*. The *Shape Vertical* feature has direct relation to the rollability of objects, whereas the *Max. Frontal* and *Min. Lateral* features determine the object's position on the table and hence give information about whether the object is reachable or fallable from the edge. Note that, removing *Min. Frontal* feature from the training set did not have a significant effect on accuracy since the existence of *Max. Frontal* in that set makes *Min. Frontal* redundant.
- For the *lift* behavior (Fig. 8(b)), *Min. Frontal* is among the most relevant four features together with *Max. Frontal*. This is unlike the case in *push-right*, where either of them would suffice for successful prediction. In the *lift* behavior, these together define the size of the object, and so determine whether the object is graspable or not. The removal of *Shape Vertical* did not have significant effect on accuracy since the number of cases where the object rolled out of the view was not high. However *Shape Horizontal* feature was significant as it tells about the surface opposed by the fingers during grasping.

⁷ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

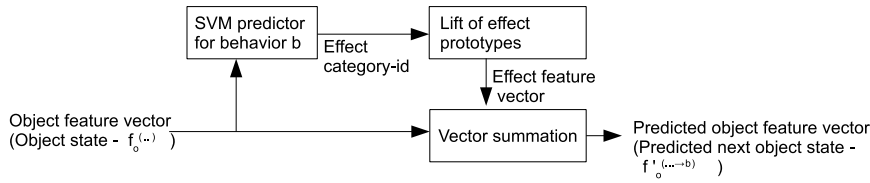


Fig. 9. Next state prediction using the general affordance relations encoded in: $\{\{\text{Predictor}()\}, \{(E_{id}, f_{\text{prototype}, id})\}\}^b$.

5. Stage 2: use of affordances in task execution

In this section, we present the methods that enable the use of learned affordances to accomplish tasks which require sequential planning. The state space search algorithms are used for this purpose, where the world state is represented in the perceptual space of the robot. Here, the world state corresponds to the list of feature vectors obtained from the objects in the environment. The initial world state can be represented as follows:

$$[f_{o_0}^0, f_{o_1}^0, \dots, f_{o_m}^0]$$

where, () denotes the zero length behavior sequence executed on the objects, and m is the maximum number of objects. If the actual number of objects is less than m , the visibility features of non-existing objects are set to 0:

$$f_{o_i}^0[0] = 0, i \leq m$$

where 0 is the index of visibility feature.

State transition occurs when the robot executes one of its behaviors on an object. Only one object is assumed to be affected at a time during the execution of a single behavior, i.e. only the state of the corresponding object is changed during a state transition. For example, if the robot executes its 3rd, 2nd, 3rd, and 1st behaviors on 1st, 1st, 2nd, and 1st objects, respectively, where $m = 3$, the resulting state will be shown as

$$[f_{o_1}^{(b_3 \rightarrow b_2 \rightarrow b_1)}, f_{o_2}^{(b_3)}, f_{o_3}^0].$$

In the previous section, the robot acquired the ability to predict the next state ($f^{(b_i)}$) based on the current state of the object f^0 using SVM classifiers (Predictor ^{b_i}) for each behavior (Fig. 9). Based on this prediction scheme, the robot can estimate the total effect that a sequence of behaviors will create and use this to predict the final object state. Thus, any goal can be encoded in the perceptual state of the robot, and a search can be done through predicting effects of different behavior sequences to reach that goal state.

Goals. The goals are represented by a set of constraints on the object features that are encoded in states. For example, the state that includes an object feature vector with $f_{o_2}^{(\dots)}[5] = [0.75 \text{ m} - 0.85 \text{ m}]$ will roughly satisfy the goal of *move the 2nd object to 0.8 m distance along the frontal axis* where 5 corresponds to the index of the feature that encodes 'minimum distance along frontal axis'. As another example, the goal of *pick-up a particular object* is satisfied in a state, where $f_{o_*}^{(\dots)}[2] = [0.35 - 0.45]$. Here, the 2nd feature corresponds to 'minimum position along vertical axis' and '*' corresponds to any object in robot's view, i.e. any object included in the world state description. If the task is to lift the 2nd object with 0.8 m frontal distance to the robot, both features are required to be satisfied.

Formally, the constraint set (goal) is composed of (object-index, feature-index, value and range) tuples $CS = \{(o_j, i, v, r)\}$. A state satisfies the goal if for all the constraints, the following inequality holds:

$$v - r \leq f_{o_j}^{(\dots)}[i] \leq v + r.$$

Goal specification. The straightforward means to set a goal is to manually decide what the constraints (features, objects, values,

and ranges) are. In this way, one can encode any goal by manually setting the desired feature value ranges for any object or objects. However, this approach requires full knowledge of the representations of the states and the meaning of all the features. In case of any changes in feature space, the goal setting procedure needs to be repeated. Furthermore, hand-tuned goal setting requires programmer intervention each time, making it time consuming and inconvenient in a world with changing tasks and goals. A more convenient way is to demonstrate an action from which the robot can automatically extract the goal and encode it in its perceptual space. This second approach is used in the next section, where the robot self-discovers the goals by observing the desired goal state of the object or objects, and then generates plans based on these.

Plan generation. This refers to finding the behavior sequence required to transform the given state into the goal state. In this study, forward chaining is used to search the state space and find a sequence. Forward chaining uses a tree structure with nodes holding the perceptual states and edges corresponding to (behavior-object) pairs. The execution of each behavior on each different object can transfer the state to a different state making the branching factor of the search tree to be *number of behaviors* \times *number of objects*. Starting from the initial state encoded in the root node, the next states for different behavior-object pairs are predicted for each state (Fig. 9). Note that object features do not change if the behavior is not executed on them, thus only one prediction is performed and one feature vector is predicted in each transition.

In order to reduce the search time, the states with minimal distance to the goal state are expanded first. The distance between states is computed using the features that appear in the constraint set. When a state reached satisfies the goal constraints, the sequence of behavior-object pairs ($\{(b_i, o_j)\}$) that transfers the initial state to that state is returned as the plan.

5.1. Control architecture

In order to test the proposed method on the real robot platform, a control architecture that supports goal emulation through automatic goal specification was implemented. The robot, the infrared range camera, and a table were placed similar to the simulated interaction environment. A closed-loop robot control architecture, which can be viewed as a 3-layer hybrid architecture ([48, Chapter 7]), was used for this purpose (Fig. 10). The *Perception Module* receives data from the range camera and computes the features of the objects, i.e. the state of the world, as described in Section 2.2. The *User Interface Module* is the means of communication with the robot. It shows the range image, the detected objects, and the features of the objects, gives a status report to the user about the plan being executed, and illustrates the search plan tree. Through the *User Interface Module* and the *Set-goal* command, the user can provide the goal environment to the robot, and he can initiate the process of goal emulation in another environment by giving *Generate-plan* and *Start-plan-execution* commands. The predictions on object features that were calculated during the planning process and stored in the nodes of the search tree can be

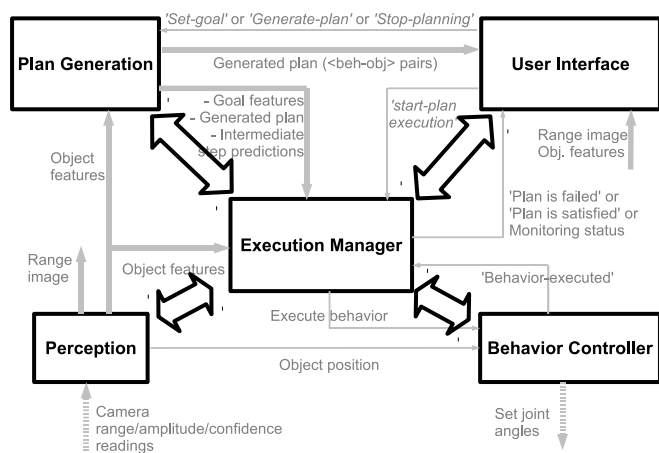


Fig. 10. Robot control architecture.

used to assess the difference between the predictions and actual perception of the environment. The *Plan Generation Module* stores the necessary knowledge ($\{\{ \text{Predictor}() \}, \{ \{ E_{id}, f_{\text{prototype}, id} \} \}^{b_j} \}$) for making predictions in the perceptual space of the robot. It stores the goal state and starts the plan generation when the *Set-goal* and *Generate-plan* commands are received. Note that both goal state and initial state of planning are provided by the *Perception Module*.

The *Execution Manager Module* is responsible for the ordered execution of behaviors and monitoring of the plan execution. It receives the plan (behavior-object pairs) from the *Plan Generation Module* and when the *Start-plan-execution* command is sent through the *User Interface Module*, the behaviors to be executed are sent one-by-one to the *Behavior Controller Module*. At each step, the *Execution Manager Module* checks whether the change in the state is as the one that was predicted in the plan, to decide whether the execution was successful or not. A mismatch detected during the execution of a behavior is reported to the *User Interface Module* causing the execution of the plan to stop. The *Behavior Controller Module* receives the behavior-id to be executed, generates a trajectory of the joint angles based on the specified behavior, object position and current joint angles, and sends it to the low-level controller of the robot arm and hand. This system is tested with several objects at varying positions in different tasks as shown in the following sections, and used to assess the effectiveness of our approach for real-world applications.

5.2. Goal setting through observation

We introduced *observation* and *imitation* phases to facilitate automatic goal setting. In the *observation* phase, the robot perceives the environment and encodes the goal based on the feature vectors obtained from the environment. In the *imitation* phase, the robot searches a sequence of behaviors that will transform the current state to the observed goal state. “What to imitate” is still an open question in developmental psychology and cognitive robotics [49,50]. Here we followed a feature-channel-specific goal emulation mechanism that prioritize some features over others.

As mentioned earlier, the states are encoded in three different feature channels. We postulated a hierarchy of importance on these features for the agent. According to this, the visibility channel is the most important one since it determines whether an object exists or not. The position channel represents the object’s location (and relation to the robot and the other objects) in the world. Finally, the shape channel gives information about the contour of the object. The robot first checks whether the object-visibility feature condition is satisfied or not. If not, it only focuses on satisfying the object-visibility condition. If it is already satisfied,

then the robot makes a plan to obtain the observed position-related features. If both object-visibility and position-related features satisfy the goal constraints, the shape-related features are chosen as the goal channel.

6. Stage 2: results

6.1. One-object imitation

Clear the table task. The goal of this task was to keep the table clear, and hence an empty table was shown to the robot in the *observation* phase. Since no object was perceived, the *object-visibility* feature was automatically set to 0. Later, during the *imitation* phase, different objects were placed on the table and the robot generated and executed plans to reduce the *object-visibility* to 0.

The snapshots taken from this experiment are shown in Fig. 11. In (1), the object was pushed and dropped from the left edge of the table using two *push-left* behaviors. In (2), a graspable object was placed at almost the same position and the robot generated a plan with *lift* and *push-left* behaviors. When these behaviors were executed, the robot lifted the object using *lift* behavior and then the object was dropped from the hand in the beginning of *push-left* behavior. The object, that was landed on the table was pushed from the edge of the table by the *push-left* behavior. In (3), the *push-left* behavior execution was predicted to drop the object from the table, however at the end of the *push-left*, the object remained on the table. The plan monitoring module detected the failure and generated a new (correct) plan (4) to roll away the object in this slightly changed configuration. In (5), when a ball was placed on the table, the *push-forward* action was executed to roll it off the table. When a large non-rollable cylinder was placed in (6), a wrong plan was generated since the diameter of the large cylinder was on the decision boundary for liftability (graspability). However, when the object’s position was slightly changed, the system was able to make a new plan (7) with four subsequent *push-right* behaviors. This experiment verifies that through interaction, the robot had learned the affordances related to physical characteristics and positions of the objects. Additionally, unsuccessful plan executions due to incorrect predictions could be corrected through the self-monitoring mechanism. Note that in order to save space, we did not include the experiments with the unreachable objects where no plan was generated, and box shaped objects that have similar movement characteristics with upright cylinders.

Move the object to a target position task. In the *observation* phase, an object lifted in the air was shown to the robot. The *observation* phase and the initial step of the *imitation* phase are shown in the upper and lower panels, respectively in Fig. 12(a)–(c). Visibility, distance and shape features were normalized and their magnitudes are shown by bars in compact form. Due to the priority-based automatic goal setting, the robot sets the goal based on position-related features and generates a plan which could transform the given object features to the observed ones. Fig. 12(d) shows the expanded nodes of the search tree, and the found plan.

The snapshots from the execution of the generated plan are shown in Fig. 13. The top panel shows the initial range images before the execution of the corresponding behaviors. The figures in the middle panel show the feature values computed from the range image. The predictions made for each feature during planning for the visibility and position feature channels are indicated by small blue boxes. The lower panel illustrates the execution of each behavior. In the end, the 7-step plan was successfully executed bringing the object approximately to the goal configuration.

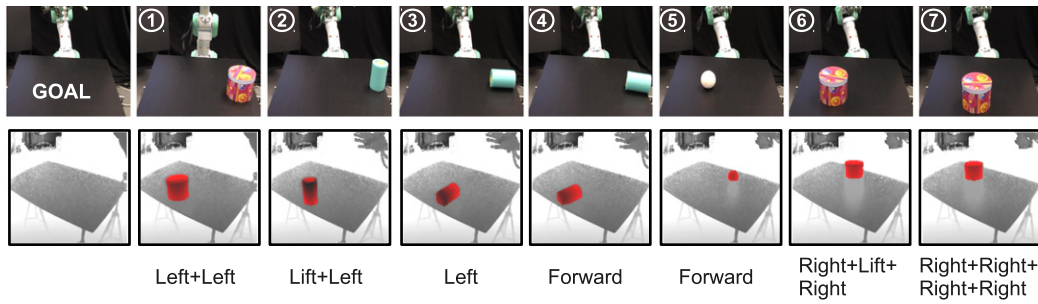


Fig. 11. Clear the table task. In the *observation* phase, an empty table is shown to the robot and the robot sets the goal *object-visibility* feature to 0. Environment snapshots, range images and generated plans are given in top, middle and bottom rows, respectively. Movies are available at <http://www.emreugur.net/movies/ras10>.

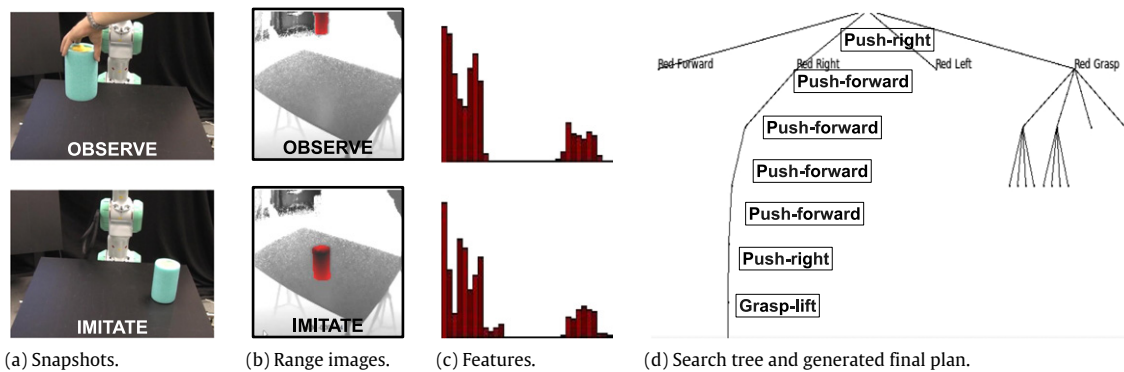


Fig. 12. Move the object to a target position task. The first panel/column corresponds to the *observation* phase and the next panels correspond to the *imitation* phase steps. The details of the computed features (third panel) are described in the text.

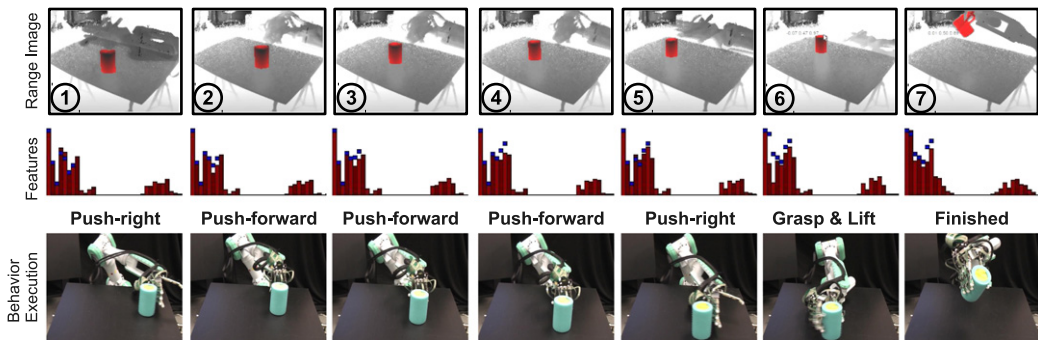


Fig. 13. The execution steps of a 7-step plan that was generated to bring the object to the observed position in Fig. 12(a) top.

6.2. Two-object imitation

We can use the learned affordances to make predictions over multiple objects under the assumption that only one object is affected by each behavior execution. For this, not the plan generation but the goal setting scheme needs to be modified for tasks involving multiple objects. In the case of two objects, the goal constraint set can be specified either absolutely or relatively. Inspired from goal emulation in biology (Section 7.1), our system sets the goals in accordance with the latter, where the relation between objects is important. The robot computes the features of the objects in the *observation* phase, gets the vectorial difference between these features and encodes this difference as the desired goal to be achieved. Setting the goal in this way is also consistent with the previous one-object imitation experiments if a second fixed object is assumed to exist (like the table or the robot's body).

The left-most panel of Fig. 14 shows a goal configuration with two objects. The top and middle feature vectors in the second panel correspond to the robot's perception in this configuration and

the bottom vector refers to the goal, computed as the difference between position features of the two objects. The right-most four panels show different situations where the robot was expected to generate plans in order to achieve the goal. In situation (1), a lying cylindrical object was placed close to the robot and a box shaped object far away. To bring these objects closer, the robot needed to either pull the box toward the cylinder or push the cylinder toward the box. The system correctly predicted that the cylinder rolls away when pushed forward and the box cannot be pulled back with the existing behaviors. Hence, no plan was generated. When the orientation of the cylinder was changed in (2), the robot predicted that the cylinder was no longer rollable, and it could be moved toward the box if pushed forward. As a result, a 4-step plan was generated with 2 *push-forward* and 2 *push-right* behaviors on the cylinder. In (3), the box was placed closer to the robot, so instead of any *push-forward* behavior, the plan consisted of two *push-right* behaviors for the cylinder and one *push-left* behavior for the box. In (4), when the upright cylinder was replaced by a sphere, i.e. a rollable object, the generated plan only included movements targeted to the box.

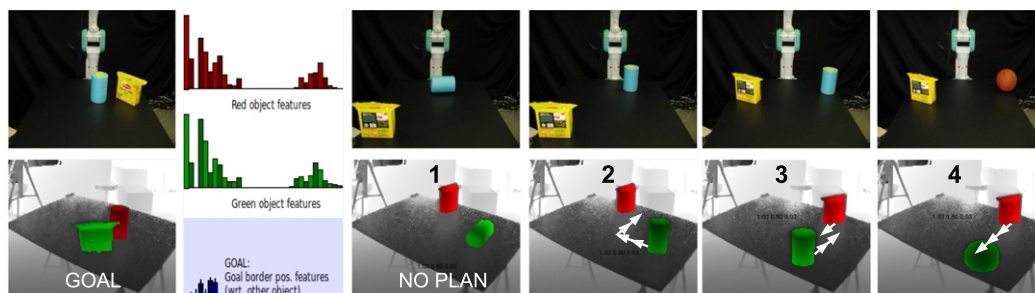


Fig. 14. Two-object imitation. The left-most panel shows the placement of objects in the *observation* phase. The second panel shows the feature vectors of objects in *observation* phase, and the difference between these vectors encoded as the goal. Right-most 4 panels show the generated plans in different setups to achieve the goal.

7. Discussion

The development of artificial agents that learn through embodied interaction with the environment is rather recent in robotics. Yet, the concepts leading to these ideas have been studied in developmental psychology for years. In the first part of this section, we discuss our robotic study in the light of cognitive science and development of human infants and chimpanzees. In addition, we discuss the representation and mechanisms required to make multi-step plans in animals with and without symbolic manipulation capabilities. In the second part, we switch our attention to the potential improvements on our study in the context of robotics.

7.1. Cognitive development

The necessity of prediction capability for goal-directed action execution and planning goes back to the 19th century. The ideomotor principle postulates that an agent must use his/her anticipation of an action's outcome to execute intentional actions [51]. Furthermore, according to this principle, these anticipations are represented as action–effect relations which are learned during the motor babbling phase through exploration of the environment [52]. Our work, among others, captures this basic observation, namely learning the effects of actions in the environment and representation of these experiences to be used in prediction and planning. In our study, the robot acquired such prediction ability in Stage 1 and utilized it in the next stage. However, by postulating stages 1 and 2 we do not claim that there is a discrete switch in the human development phases; it rather makes the analysis of learning and acting with the learned knowledge convenient. One can also relate the stages we postulate with the bifurcations that happen throughout the evolution of a dynamical system that might be representing development [53] (e.g. the switch from quadruped walking to bipedal walking). Instead of simulating a dynamical system that exhibits such bifurcation, we simulate the two modes of the dynamical system as separate stages.

An unsettled discussion is whether the goals and predictions are encoded in the perceptual space of the agent [54] or not. This is an important issue both for a robot designer and a neuroscientist searching for neural correlates of intelligent behavior. Although, most would agree that a hierarchy of predictive mechanisms, ranging from sensory to abstract, is a prerequisite for intelligent behavior, we further argue that this by itself is not sufficient. The critical issue is that the goals and the predictions can be expressed in the low-level perceptual space when needed. According to [14], this is supported by recent behavioral and neuropsychological findings. In the experiments presented with our system, the goals were specified directly in the perceptual state of the robot. However, this is not the only possibility as the prediction mechanisms we employed represent the effects of actions in two levels. One is the affordance level where discrete and abstract items are predicted (i.e. effect categories). The second is

the sensory level where the prediction takes place in the perceptual space (i.e. current perceptual state + effect category prototype). If we were to consider our robot as a ‘high-level agent’ and ask it to bring about the ‘lifted effect’ it would be able to tell whether there is, any object that affords this high-level (non-perceptually specified) goal, and if, indeed, there is such an object, it would be able to execute a behavior to bring about the desired effect.

In spite the ongoing debate on whether these anticipations are represented in the sensory-motor space or in a more abstract level, it is widely accepted that these mechanisms are used in planning [55]. According to Piaget [10], human infants start to distinguish means–ends relation at 4–8 months, and start to use these relations until around 12 months for one step goal satisfaction. It is not implausible that a limited amount of anticipation skill be hardwired through evolution in humans and other animals; however, the majority of this skill must be acquired by the organisms through interaction with the environment. Elsner and Hommel [15] and Hommel [56] argue that this ability cannot be innate and the human infant learns to use anticipation for goal-directed action execution in his/her early months of infancy. Infants use the learned action–effect relations to anticipate the results of their actions in a goal-directed way starting from 9 months [10–12]. Piaget argued that planning is only possible after the development of symbolic representation at 18–24 months, although there is evidence that younger children are able to make multi-step plans. For example, 9-month-olds are shown to generate a multi-step plan to reach a toy, by first removing the obstacle, then pulling the towel and grabbing the object placed on it [13,57]. In our system, the prediction ability was demonstrated for multi-step planning without going through a gradual development. This, however, could be easily emulated by restricting the planner to plans of depth one and gradually increasing the allowed depth in the search for plans to achieve the desired progression. One can speculate that the inability of early infants to make complex plans is due to an immature working memory needed for planning. As the infant grows, the increasing memory that is available to the planning may allow complex plans. This argument regards the planning mechanism as fixed but requiring more memory as the sought plans become more complex; however this may not be the case as different planning mechanisms may coexist and develop along with the infant's cognitive development [58]. It is largely unknown whether symbolic manipulation ability is necessary for complex planning, as Piaget argued. Our stance is that, computationally, there is no such necessity; even though a symbol manipulation machinery would be an invaluable tool for planning and other cognitive skills. Especially, this would be more advantageous as the plans shift from physical to social domain. In the presented system, the plans are performed in the perceptual domain which allow the robot to naturally interact with objects which it did not experience before. For example, it would be able to make a bottle disappear from the table, even though it has no idea what a bottle is and has never seen it before.

One feature we introduced for specifying the goals automatically in our system has interesting relations to the so-called 'goal emulation' in cognitive psychology. The term can be defined as an observer's learning that a particular goal can be achieved and setting about achieving it by its own [9]. Goal emulation is different from other social learning mechanisms such as mimicry and imitation and somewhat puzzling. Most animal mimicry is restricted to goal emulation, which is generally regarded as a simpler task than imitation. However, human infants who can imitate are unable to use goal emulation to learn new skills. Children show mimicry before 12 months of age but only start to pay attention to goals of the demonstrator only after that period [6,7,9]. 17-month-olds can use observed actions or their own action repertoire to achieve the observed goal depending on the context [59], 18-month-olds can understand 'intended' goals of a demonstrator trying but failing to achieve his goal [60], and 18-month-olds can learn tool use by observation. However, goal emulation, i.e. executing a sequence of behaviors after observing only the goal state, develops rather late [61,6] and only after 18-month-old infants are able to emulate action sequences for novel goals [62,63]. For example, in [64,63], 27-month-old infants were able to execute a 3-step plan to construct a rattle from two cups and a ball but 21-month-olds were not. The puzzling findings for human infant goal emulation could be due to the lack of motivation or the insufficient affordance experience with the toys used in the experiments. Many animals are known to make multi-step plans involving different types of objects. For example in [65], chimpanzees stack four boxes on top of each other to reach a banana that was hung out of their reach. They were also able to climb on a long stick instead of stacking the boxes when that stick was available in the environment. However, the same chimpanzees were not able to generate plans with those objects when the objects resided in another (accessible) room even if chimpanzees had explored that room recently. This observation led [66] to conclude that non-linguistic animals use object affordances to make plans, and they start reasoning from the immediate environment (initial state) to reach the goals, i.e. they use forward chaining. Furthermore, the plan generation can be successful if they have learned the affordances of the objects before [67].

In light of the above discussion, we can argue that our robot system when run in an automatic goal setting mode is more similar to a chimpanzee rather than a human infant, as the goal is more important than the means for a chimpanzee. Although, chimpanzees utilize social learning mechanisms to develop various tool use skills, unlike humans, they are less sensitive to demonstrator's body movements and tend to emulate the goal more than to imitate the demonstrator [68]. In fact movements that do not have apparent targets such as another object or a body part has little imitation potential for chimpanzees [69]. Having said this, one should note that in the current implementation, unlike infants and chimpanzees, robot's development was not completely autonomous. The robot did not have a programmed will to do actions (say to feed itself). Instead we substituted such ecologically meaningful goals with the designer's goal setting, say, the goal of 'cleaning the table'. If one wishes, this can be equated to say, fulfilling the hunger of a robot.

One final similarity of our system's working with a chimpanzee's cognitive abilities is that chimpanzees have difficulties when manipulating objects in different multiplicities. It may be speculated that this could be due the lack of symbolic planning ability of chimpanzees. This is analogous to the case in our system: our system benefits from having the planning done in perceptual space in terms of generalization and robustness; but it faces difficulty in encoding goals in the same representation for different number of objects. In the current implementation, we overcame this by introducing a special goal setting mechanism inspired

from the observation (unpublished video related to [69]) that when chimpanzees are asked to imitate an action involving two objects (put an object in a bowl), they appear to reproduce the spatial relation of the objects rather than the absolute spatial configuration shown to them (by holding the object and the bowl in both hands and bring them together in the air instead of on table).

Although our affordance-based robot control system learned affordances that allowed the robot to make plans to fulfill a given goal, the relation of these affordances to other higher level cognitive processes, such as recognition is an open question. For example, when one wishes to augment the robot with object recognition capability, would the acquired affordances be of any help? Neisser [70,71] tried to place affordances and direct perception into a complete cognitive system model and tried to link them with other cognitive processes. According to him, Gibson was right in stating that the meanings of the environment are directly available. Invariance attuned detectors are used for this purpose. However, he claimed, the Gibsonian view of affordances of perception is inadequate, since "it says so little about perceiver's contribution to the perception act" [70, p. 9]. Instead, he suggested a perceptual system where a cyclic activity, continuous over time and space, occurs. This cycle "prepares the perceiver to accept certain kinds of information... At each moment the perceiver is constructing anticipations of certain kinds of information, that enable him to accept it (information) as it becomes available" [70, p. 9]. Since every natural object has an infinite number of affordances, this cycle could also be employed to prepare the perceiver to search for particular affordances at each moment, and attune specific detectors to perceive these affordances.

In this study, we simplified the development by keeping the body of the robot and its motor skills fixed and focusing on the adaptation of the perception and visuomotor learning. However, the perceptual and motor development is tightly coupled in biological systems. Thelen's dynamical systems view takes Gibsonian perception-action concept and places it in a more biologically plausible developmental perspective [53,72]. She proposed a machinery derived from dynamical systems theory, and attempted to explain development, perception, and action (and to some extent decision making) in a single framework. In this framework, multiple physical and biological variables such as muscle power and body mass are fundamental factors for movement development that significantly affects perception since they are inseparable and reciprocally developed. The learning and shifts between developmental phases can be modulated in a more natural way if such variables are included in robot's behavior and skill development.

We close this section by noting that although Piaget's requirement for symbolic manipulation ability for complex planning might be too strict, higher cognitive abilities, including multi-object and memory based planning requires the development of symbolic planning mechanisms irrespective of whether the symbols manifest themselves as linguistic constructs or not.

7.2. Potential improvements

Discrete pre-coded behavior repertoire. In this work, we assumed that the robot has a basic behavior repertoire that was assumed to be learned before.⁸ Furthermore, these fixed robot behaviors were encoded as discrete and non-parametric actions for simplicity. For example, push-left and push-right behaviors were regarded

⁸ For computational convenience we sidestepped the early learning phase of an infant. Ideally, one would like to simulate development from birth, and probably even before that [1].

as independent primitives whose affordances were learned independent of each other. Thus, a rolling effect experience created on a ball by push-left behavior cannot be generalized to the other push directions. One improvement on our system would be to group similar behaviors such as push-left and push-right under a generic parametric behavior; for example a push behavior with approach-direction parameter would encapsulate the desired repertoire. Then, accordingly, the affordance learning framework needs to be modified such that learning establishes the mapping from (object features, behavior parameters) to effect categories. In [73], a more low-level and integrative approach was adopted where the behaviors were represented as learned dynamical systems with adjustable parameters. At the core of this approach is a recurrent neural network that allows sensorimotor prediction as well as producing next state information driving the joints of the robot. This system, by experiencing multiple push actions on different objects, could learn multiple sequences of robot and object motions and encode this visuomotor experience in the same model. The different object-robot dynamics are captured in, so-called, parametric bias variables. The robot could reconstruct and generate novel sequences through adjusting these variables. After defining a goodness measure, the robot could find the most suitable parametric bias values to obtain the desired performance measure. With this approach, the robot could generate robust action executions by finding parametric bias values that would lead to reliably predictable movements and effects. In particular, [73] showed that after training, the robot could choose motions that would result in consistent and reliable object rolling motions depending on the orientation of the objects.

Perceptual features. The perceptual features used in this study, were inspired from the properties of primate visual processing pathway that are relevant for affordance detection [74]. In particular, edge detection (visual area V1), depth processing (visual area V3) and surface and axis representations (CIP—Caudal Intraparietal Area) are critical subprocesses of the dorsal visual pathway of the primate cortex leading to the affordance representation in AIP—the anterior intraparietal area [74]. As not all of these features may be necessary for a given specific action, a selective feature processing mechanism (perceptual economy) is a parsimonious solution for evolution and the robot designer. For example, the surface shape feature that determines the rollability affordance of an object may not be important for grasping as long as the object is not too big to prohibit proper grasping. Although, our system exhibits perceptual economy as shown in Section 4.3, manual design of the features albeit inspired from biology might not capture the ideal information for the robot and the learning environment as the robot is inevitably different from a biological entity. Thus, it may be worthwhile to pursue the development of a robot that will take the incoming raw image, and automatically discover object features while simultaneously learning affordance relations. Indeed [75,76] studied automatic discovery of object features using a dynamical systems approach, where the robot self-organized motion and shape-related features observing the incoming raw image from its camera. For example, concerning shape features, the robot discovered stability and roundness of the objects by observing the effect of its push actions.

Multi-object affordances. The manipulation affordances were learned by interacting with one object at a time. In the early stages of development, infants also appear to have a similar strategy [77, p. 199]. When they are engaged in playing with an object, they often show no interest to other objects. However in the later phases of development, they start playing with multiple objects and are very much interested in their interaction. We can speculate that for human infants, learning the object-object affordance relations is the subsequent stage after single-object affordance learning. Our robot could generate and execute plans

with multiple objects under the assumption that only one object is affected by the behavior execution. This assumption is too strict in real life and should be relaxed in the future. However, this will invoke more complex perceptual processing that need to be addressed in a developmental setting such as object consistency, and object continuity (e.g. early age infants fail to perceive two objects stacked together as two separate entities).

Environment. In this study, the robot learned the affordances of three different types of objects, namely boxes, cylinders, and spheres. The feature representation used in perception has, naturally, significant effect on the scalability of the proposed prediction mechanism to novel real-world objects. One of the features we used, namely the normal histograms of object surfaces provides a decent level of generalization as we showed in the mobile robot traversability study [22]. With this feature, the robot was able to generalize the learned prediction ability of objects that were not experienced before. For example, after interacting with only cylinders lying on the ground, it was able to predict that boxes and upright cylinders are not rollable, and spheres are rollable. Furthermore, with this prediction capability the robot was able to navigate in a room cluttered with real-world office objects. Nevertheless, we think that the utilized feature representation provides limited generalization capability when objects are non-convex and consist of multiple parts (e.g. a mug). To address this limitation, an attention system that focuses on affordance bearing sub-parts of objects could be developed.

Deterministic prediction. Our system uses deterministic learning algorithms, which have limited power in capturing uncertainty of the environment and the robot state. Thus, 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 the developmental stages of human infants and not sacrificing the planning ability demonstrated by our system.

8. Conclusion

In this paper, we have shown that through self-interaction and self-observation, an anthropomorphic robot and a range camera system can learn the object affordances in an unsupervised way. The proposed learning system share crucial elements such as goal-free exploration and self-observation with infant development. After learning, the robot can make plans to achieve desired goals and also emulate end state of demonstrated actions. The plans are based on affordance prediction capability and may involve multiple objects. Furthermore, the system can monitor the plan execution and take corrective actions using the perceptual structures employed in learning.

In the first step of learning, the robot discovers commonalities in action-effect experiences by finding effect categories caused by its actions. For this purpose, the robot uses a novel hierarchical clustering algorithm that was developed for dealing with non-homogeneous feature spaces. This algorithm, first clusters effects into channel-specific categories and then takes their Cartesian product to obtain all-channel effect categories. Predictors for each behavior are then trained to map object features into effect categories using nonlinear classifiers. Using the category prototypes, the robot can make predictions about the next perceptual state of the object acted upon enabling it to make multi-step plans for achieving goals represented as constraints defined over the object features.

The key aspect of our approach is that, the agent learns about its environment by discovering the effects it can generate through its actions, and forms forward models [78] that enable it to predict the changes in the environment in terms of discrete effect categories as well as low-level sensory changes. Predicting the 'change in state'

rather than the 'next state' provides better generalization, and, at the same time, allows 'next state' prediction so that multiple steps into the future can be predicted facilitating multi-step planning. Finally, by representing the environment in relation to the effects, our agent 'understands' the world in regards to its own action capabilities, fully adhering to the action based perception view.

Acknowledgments

Emre Ugur acknowledges the financial support of TÜBİTAK (The Scientific and Technical Research Council of Turkey) through the National Scholarship Programme for Ph.D. Students. 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. This work was also partially funded by the European Commission under the ROSSI project (FP7-ICT-21625) and the TÜBİTAK through project 109E033. We thank Y. Shimizu and B. Moore for their comments on the manuscript. We would also like to thank our anonymous reviewers for their comments and suggestions.

References

- [1] Y. Kuniyoshi, S. Sangawa, Early motor development from partially ordered neural-body dynamics: experiments with a cortico-spinal-musculo-skeletal model, *Biological Cybernetics* 95 (2006) 589–605.
- [2] L. Righetti, A. Ijspeert, Design methodologies for central pattern generators: an application to crawling humanoids, in: G.S. Sukhatme, S. Schaal, W. Burgard, D. Fox (Eds.), *Proceedings of Robotics: Science and Systems*, MIT Press, 2006, pp. 191–198.
- [3] J. Hörnstein, L. Gustavsson, F. Lacerda, J. Santos-Victor, Multimodal word learning from infant directed speech, in: *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE Press, St. Louis, MO, USA, 2009, pp. 2748–2754.
- [4] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, C. Yoshida, Cognitive developmental robotics: a survey, *IEEE Transactions on Autonomous Mental Development* 1 (2009) 12–34.
- [5] J. Gibson, *The Ecological Approach to Visual Perception*, Lawrence Erlbaum Associates, 1986.
- [6] J. Provasi, C. Dubon, H. Bloch, Do 9- and 12-month-olds learn means-ends relation by observing? *Infant Behavior and Development* 24 (2001) 195–213.
- [7] M. Carpenter, K. Nagell, M. Tomasello, Social cognition, joint attention, and communicative competence from 9 to 15 months of age, *Monographs of the Society for Research in Child Development* 63 (1998).
- [8] B. Elsner, G. Aschersleben, Do I get what you get? Learning about the effects of self-performed and observed actions in infancy, *Consciousness and Cognition* 12 (2003) 732–751.
- [9] S.C. Want, P.L. Harris, How do children ape? Applying concepts from the study of non-human primates to the developmental study of imitation in children, *Developmental Science* 5 (2002) 1–13.
- [10] J. Piaget, *The Origins of Intelligence in Children*, International University Press, New York, USA, 1952.
- [11] M. Tomasello, *The Cultural Origins of Human Cognition*, Harvard University Press, Cambridge, 1999.
- [12] P. Willatts, Development of means-end behavior in young infants: pulling a support to retrieve a distant object, *Developmental Psychology* 35 (1999) 651–666.
- [13] P. Willatts, The Stage IV infant's solution of problems requiring the use of supports, *Infant Behavior and Development* 7 (1984) 125–134.
- [14] B. Elsner, Infants' imitation of goal-directed actions: the role of movements and action effects, *Acta Psychologica* 124 (2007) 44–59.
- [15] B. Elsner, B. Hommel, Effect anticipation and action control, *Journal of Experimental Psychology* 27 (2003) 229–240.
- [16] E. Şahin, M. Çakmak, M.R. Doğan, E. Ugur, G. Üçoluk, To afford or not to afford: a new formalization of affordances toward affordance-based robot control, *Adaptive Behavior* 15 (2007) 447–472.
- [17] K. Jones, What is an affordance? *Ecological Psychology* 15 (2003) 107–114.
- [18] J.C. Culham, K.F. Valyear, Human parietal cortex in action, *Current Opinion in Neurobiology* 16 (2006) 205–212.
- [19] M.A. Goodale, Action without perception in human vision, *Cognitive Neuropsychology* 25 (2008) 891–919.
- [20] M.A. Goodale, A.D. Milner, Separate visual pathways for perception and action, *Trends in Neurosciences* 15 (1992) 20–25.
- [21] L.G. Ungerleider, M. Mishkin, Two cortical visual systems, in: *Analysis of Visual Behavior*, MIT Press, Cambridge, MA, 1982, pp. 549–586.
- [22] E. Ugur, E. Şahin, Traversability: a case study for learning and perceiving affordances in robots, *Adaptive Behavior* 18 (2010).
- [23] M.R. Doğan, M. Çakmak, E. Ugur, E. Şahin, From primitive behaviors to goal-directed behavior using affordances, in: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2007, pp. 729–734.
- [24] E. Ugur, E. Şahin, E. Oztop, Affordance learning from range data for multi-step planning, in: *Proceedings of the 9th International Conference on Epigenetic Robotics, EpiRob'09*, Lund University Press, Venice, Italy, 2009, pp. 177–184.
- [25] J. Weng, J. Mc Clelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, E. Thelen, Autonomous mental development by robots and animals, *Science* 291 (2001) 599–600.
- [26] M. Lungarella, G. Metta, R. Pfeifer, G. Sandini, Developmental robotics: a survey, *Connection Science* 15 (2003) 151–190.
- [27] A. Stoytchev, Some basic principles of developmental robotics, *IEEE Transactions on Autonomous Mental Development* 1 (2009) 122–130.
- [28] G. Metta, P. Fitzpatrick, Better vision through manipulation, *Adaptive Behavior* 11 (2003) 109–128.
- [29] P. Fitzpatrick, G. Metta, L. Natale, A. Rao, G. Sandini, Learning about objects through action-initial steps towards artificial cognition, in: *Proc. of ICRA 03*, IEEE, 2003, pp. 3140–3145.
- [30] A. Stoytchev, Behavior-grounded representation of tool affordances, in: *Proceedings of IEEE International Conference on Robotics and Automation, ICRA, IEEE, Barcelona, Spain, 2005*, pp. 18–22.
- [31] E. Erdemir, C.B. Frankel, K. Kawamura, S.M. Gordon, S. Thornton, B. Ulutas, Towards a cognitive robot that uses internal rehearsal to learn affordance relations, *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008) 2016–2021.
- [32] G. Fritz, L. Paletta, M. Kumar, G. Dorffner, R. Breithaupt, R. Erich, Visual learning of affordance based cues, in: S. Nolfi, G. Baldassarre, R. Calabretta, J. Hallam, D. Marocco, J.-A. Meyer, D. Parisi (Eds.), *From Animals to Animats 9: Proceedings of the Ninth International Conference on Simulation of Adaptive Behaviour, SAB*, in: *LNAI*, vol. 4095, Springer-Verlag, Berlin, Roma, Italy, 2006, pp. 52–64.
- [33] J. Sinapov, A. Stoytchev, Detecting the functional similarities between tools using a hierarchical representation of outcomes, in: *7th IEEE International Conference on Development and Learning*, IEEE, 2008, pp. 91–96.
- [34] S. Griffith, J. Sinapov, M. Miller, 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, IEEE, Shanghai, China, 2009*, pp. 1–6.
- [35] I. Cos-Aguilera, L. Canamero, G.M. Hayes, Using a SOFM to learn object affordances, in: *In Proceedings of the 5th Workshop of Physical Agents, Girona, Catalonia, Spain*.
- [36] Y. Demiris, A. Dearden, From motor babbling to hierarchical learning by imitation: a robot developmental pathway in: *Fifth International Workshop on Epigenetic Robotics*, Lund University, 2005, pp. 31–37.
- [37] S. Hart, R. Grupen, D. Jensen, A relational representation for procedural task knowledge, in: *Proceedings of the National Conference on Artificial Intelligence, AAAI Press, 2005*, pp. 1280–1285.
- [38] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, Learning object affordances: from sensory-motor maps to imitation, *IEEE Transactions on Robotics* 24 (2008) 15–26.
- [39] R. Petrick, D. Kraft, K. Mourão, N. Pugeault, N. Krüger, M. Steedman, Representation and integration: combining robot control, high-level planning, and action learning, in: P. Doherty, G. Lakemeyer, A. Pobil (Eds.), *Proceedings of the 6th International Cognitive Robotics Workshop*, 2008, pp. 32–41.
- [40] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, B. Porr, Cognitive agents a procedural perspective relying on the predictability of object-action-complexes, *OACs, Robotics and Autonomous Systems* 57 (2009) 420–432.
- [41] J. Modayil, B. Kuipers, The initial development of object knowledge by a learning robot., *Robotics and Autonomous Systems* 56 (2008) 879–890.
- [42] H. Arie, T. Endo, T. Arakaki, S. Sugano, J. Tani, Creating novel goal-directed actions at criticality: a neuro-robotic experiment, *New Mathematics and Natural Computation* 5 (2008) 307–334.
- [43] J. Pisokas, U. Nehmzow, Experiments in subsymbolic action planning with mobile robots, in: *Adaptive Agents and Multi-Agent Systems II*, in: *Lecture Notes in Artificial Intelligence*, Springer, 2005, pp. 80–87.
- [44] R.M. Haralick, L.G. Shapiro, *Computer and Robot Vision*, vol. I, Addison-Wesley, 1992.
- [45] D. Pelleg, A. Moore, X-means: extending k-means with efficient estimation of the number of clusters, in: *Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann, 2000, pp. 727–734.
- [46] V.N. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, 1998.
- [47] A. Moore, M. Lee, Efficient algorithms for minimizing cross validation error, in: R. Greiner, D. Schuurmans (Eds.), *Proceedings of the 11th International Conference on Machine Learning*, Morgan Kaufmann, 1994, pp. 190–198.
- [48] R.R. Murphy, *An Introduction to AI Robotics*, The MIT Press, 2000.
- [49] Y. Nagai, K. Rohlfing, Can motionese tell infants and robots what to imitate, in: *Proceedings of the 4th International Symposium on Imitation in Animals and Artifacts, AISB*, 2007, pp. 299–306.
- [50] C. Nehaniv, K. Dautenhahn, Like me?-measures of correspondence and imitation, *Cybernetics and Systems* 32 (2001) 11–51.
- [51] W. James, *The Principles of Psychology*, Dover Publications, New York, 1890.
- [52] G. Pezzulo, G. Baldassarre, M. Butz, C. Castelfranchi, J. Hoffmann, From actions to goals and vice-versa: theoretical analysis and models of the ideomotor principle and tote, in: *Anticipatory Behavior in Adaptive Learning Systems*, Springer Verlag, 2007, pp. 73–93.

- [53] E. Thelen, Motor development as a foundation and future of developmental psychology, *International Journal of Behavioral Development* 24 (2000) 385–397.
- [54] A.G. Greenwald, Sensory feedback mechanisms in performance control: with special reference to the ideomotor mechanism, *Psychological Review* 77 (1970) 73–99.
- [55] B. Hommel, E. Birgit, Acquisition, representation, and control of action, in: *Oxford Handbook of Human Action*, Oxford University Press, New York, 2008, pp. 371–398.
- [56] B. Hommel, Acquisition and control of voluntary action, in: *Voluntary Action: Brains, Minds, and Sociality*, Oxford University Press, Oxford, UK, 2003, pp. 34–48.
- [57] Z. Chen, R. Sanchez, T. Campbell, From beyond to within their grasp: the rudiments of analogical problem solving in 10- and 13-month-olds, *Developmental Psychology* 33 (1997) 790–801.
- [58] R. De Lisi, A cognitive-developmental model of planning, in: S. Friedman, E. Scholnick, R. Cocking (Eds.), *Blueprints for Thinking: The Role of Planning in Cognitive Development*, Cambridge University Press, 1987, pp. 79–109.
- [59] G. Gergely, H. Bekkering, I. Kiraly, Rational imitation in preverbal infants, *Nature* 415 (2002) 755.
- [60] A.N. Meltzoff, Understanding the intentions of others: re-enactment of intended acts by 18-month-old children, *Developmental Psychology* 31 (1995) 838–850.
- [61] F. Bellagamba, Re-enacting intended acts: comparing 12- and 18-month-olds, *Development* 22 (1999) 277–282.
- [62] C.-t. Huang, T. Charman, Gradations of emulation learning in infants imitation of actions on objects, *Journal of Experimental Psychology* 92 (2005) 276–302.
- [63] P.J. Bauer, J.A. Schwade, S.S. Wewerka, K. Delaney, Planning ahead: goal-directed problem solving by 2-year-olds, *Developmental Psychology* 35 (1999) 1321–1337.
- [64] P.J. Bauer, Holding it all together: how enabling relations facilitate young children's event recall, *Cognitive Development* 7 (1992) 1–28.
- [65] W. Kohler, *The Mentality of Apes*, Harcourt Brace and World, New York, 1925.
- [66] M. Steedman, Plans, affordances, and combinatory grammar, *Linguistics and Philosophy* 25 (2002).
- [67] J.L. Gould, C.G. Gould, *The Animal Mind*, Scientific American Library, 1994.
- [68] C. Tennie, J. Call, M. Tomasello, Push or pull: imitation vs. emulation in great apes and human children, *Ethology* 112 (2006) 1159–1169.
- [69] M. Myowa-Yamakoshi, T. Matsuzawa, Factors influencing imitation of manipulatory actions in chimpanzees (Pan troglodytes), *Journal of Comparative Psychology* 113 (1999) 128–136.
- [70] U. Neisser, *Cognition and Reality: Principles and Implications of Cognitive Psychology*, W.H. Freeman and Co., 1976.
- [71] U. Neisser, Multiple systems: a new approach to cognitive theory, *European Journal of Cognitive Psychology* 6 (1994) 225–241.
- [72] L.B. Smith, Movement matters: the contributions of Esther Thelen, *Biological Theory* 1 (2006) 87–89.
- [73] S. Nishide, T. Ogata, J. Tani, K. Komatani, H.G. Okuno, Autonomous motion generation based on reliable predictability, *Journal of Robotics and Mechatronics* 21 (2009).
- [74] E. Oztop, H. Imamizu, G. Cheng, M. Kawato, A computational model of anterior intraparietal (AIP) neurons, *Neurocomputing* 69 (2006) 1354–1361.
- [75] S. Nishide, T. Ogata, J. Tani, K. Komatani, Predicting object dynamics from visual images through active sensing experiences, *Advanced Robotics* 22 (2008) 527–546.
- [76] S. Nishide, T. Ogata, J. Tani, K. Komatani, H.G. Okuno, Self-organization of dynamic object features based on bidirectional training, *Advanced Robotics* 23 (2009) 2035–2057.
- [77] R. Baillargeon, How do infants learn about the physical world?, in: D. Muir, A. Slater (Eds.), *Infant Development: The Essential Readings. Essential Readings in Developmental Psychology*, Blackwell Publishers, Inc., Malden, MA, 2000, pp. 195–212.
- [78] M. Kawato, Internal models for motor control and trajectory planning, *Current Opinion in Neurobiology* 9 (1999) 718–727.



Emre Ugur received his Ph.D., M.Sc. and B.Sc. degrees in Computer Engineering from Middle East Technical University, in 2010, 2006 and 2003 respectively. Since 2009, he has been working as a researcher at Computational Neuroscience Laboratories, Advanced Telecommunications Research Institute International, Japan. Prior to this, he worked in a number of research projects (ROSSI, MACS and Swarm-bots) funded by the European Commission. He is interested in developmental and cognitive robotics.



Erhan Oztop earned his Ph.D. from the University of Southern California in 2002. In the same year, he joined the Computational Neuroscience Laboratories at Advanced Telecommunications Research Institute International (ATR) in Japan where he worked as a researcher until 2004. Immediately after, he joined the JST ICORP Computational Project as a researcher, and later became a group leader (2004–2008). Currently, he is a senior researcher at ATR Cognitive Mechanisms Laboratories where he is leading the Communication and Cognitive Cybernetics group. He holds a visiting associate professor position at Osaka University and is also affiliated with NICT, Biological ICT group.



Erol Sahin has a B.Sc. in Electrical and Electronics Engineering from Bilkent University, Turkey, in 1991, an M.Sc. in Computer Engineering from Middle East Technical University (METU) in 1995, and a Ph.D. in Cognitive and Neural Systems from Boston University, USA, in 2000. He is working as an Assistant Professor at the Dept. of Computer Engineering at METU and is heading the KOVAN Research Laboratory. His work on cognitive systems focuses on how the notion of affordances can be used at different levels of autonomous robot control, and how the notion can be linked to mirror and canonical neurons for developing concepts that correspond to verbs and nouns in language (through the ROSSI project, Web: <http://rossiproject.eu>). Recently he has been awarded one of the free iCub humanoid robot platforms by the RobotCub project (Web: <http://robotcub.org>), to carry on his research on the topic. Dr. Sahin has also been working on swarm robotics, and has edited two books and two special issues on the topic.