Learning Object Affordances: From Sensory–Motor Coordination to Imitation

Luis Montesano, Manuel Lopes, Alexandre Bernardino, *Member, IEEE*, and José Santos-Victor, *Member, IEEE*

***Abstract*—Affordances encode relationships between actions, objects, and effects. They play an important role on basic cognitive capabilities such as prediction and planning. We address the problem of learning affordances through the interaction of a robot with the environment, a key step to understand the world properties and develop social skills. We present a general model for learning object affordances using Bayesian networks integrated within a general developmental architecture for social robots. Since learning is based on a probabilistic model, the approach is able to deal with uncertainty, redundancy, and irrelevant information. We demonstrate successful learning in the real world by having an humanoid robot interacting with objects. We illustrate the benefits of the acquired knowledge in imitation games.**

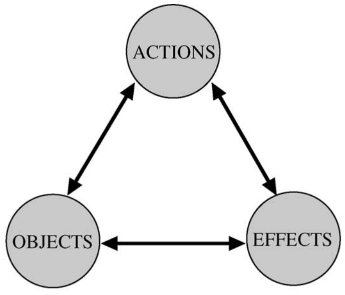
***Index Terms*—Affordances, biorobotics, cognitive robotics, humanoid robots, learning.**

I. Introduction

HUMANS can solve many complex tasks on a routine basis, e.g., by selecting, amongst a vast repertoire, the actions to exert on an object to obtain a certain desired effect. A painter knows which colors and painting technique to use to produce a certain visual impression, in the same way as a basketball player knows how to throw a ball with the exact trajectory and spin to introduce it in the basket.

In this paper, we discuss such human skills in the context of the long-term vision of building (humanoid) robots capable of acting in a complex world and interacting with humans and objects in a flexible way. What knowledge representations or cognitive architecture should such a system require to be able to act in such unpredictable environment? How can the system acquire task or domain-specific knowledge to be used in novel situations? To help answering these questions, we propose a methodology that draws inspiration from the concept of affordances introduced by Gibson in his seminal work [1]. He defined affordances as action possibilities available in the environment to an individual, thus depending on its action capabilities.

Affordances define the relation between an agent and its environment through its motor and sensing capabilities (e.g., graspable, movable, or eatable), as illustrated in Fig. 1. For instance, humans can grasp a cup or sit on a sofa, but not *vice versa*. Dogs can sit on a sofa but cannot grasp a cup.



inputs	outputs	function
(O, A)	E	Predict effect
(O, E)	A	Action recognition & planning
(A, E)	O	Object recognition & selection

Fig. 1. Affordances as relations between (A)ctions, (O)bjects, and (E)ffects that can be used to address different purposes: predict the outcome of an action, plan actions to achieve a goal, or recognize objects or actions.

From the perspective of robotics, affordances are extremely powerful since they capture the essential world and object properties, in terms of the actions the robot is able to perform. They can be used to predict the effects of an action, to plan actions to achieve a specific goal, or to select the object to produce a certain effect if acted upon in a certain way.

By extending the concept further, affordances also play an important role for interacting with other agents, since they allow the recognition of actions and can be used, for instance, in imitation [2]. By observing the actions, states and effects of other individuals (human or robots), an artificial system can retrieve a tremendous amount of knowledge [3]. Learning by imitation is one of the motivations of our affordance system and we will show how it can lead to imitation-like behaviors.

There are two points that should be stressed now regarding affordances. Firstly, one intrinsic characteristic of affordances is that they result from the (ecological) exploratory interaction between the robot and the environment, thus depending both on the world and the agent's motor and perceptual capabilities. Secondly, the concept of affordances requires a certain number of elementary actions to be defined and functional. As we shall see later, this means that the system must first know how to perform a number of actions and develop some perceptual capabilities before learning the affordances.

Manuscript received February 14, 2007; revised October 05, 2007. This paper was recommended by Associate Editor P. Dario and Editor F. Park upon evaluation of the reviewers' comments. This work was supported in part by the FCT Programa Operacional Sociedade de Informação (POSC) in the frame of QCA III and PTDC/EEA-ACR/70174/2006 project; and in part by the EU Project RobotCub (IST-004370) and Project Contact (EU-FP6-NEST-5010).

The authors are with the Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa 1049-001, Portugal (e-mail: lmontesano@isr.ist.utl.pt; macl@isr.ist.utl.pt; alex@isr.ist.utl.pt; jasv@isr.ist.utl.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2007.914848

A. Related Work

Gibson used the concept of affordance to describe the relation (including representation issues) established between a living being and its environment [1]. Gibson argues that this relation is shaped by the perceptual and motor abilities of the agent. Hence, affordances represent what the elements present in the environment afford to the agent. This very general concept was originally applied to entities such as surfaces (ground, air, water) or their frontiers. From a psychological point of view, there has been a lot of discussion to establish a definition or model of affordances (see [4] for a brief review). Other authors have shown the presence of affordances by comparing percepts among different people [5], measuring response times to tasks elicited by specific object orientations [6], or perceiving heaviness [7]. Unfortunately, there is little evidence on how humans learn affordances.

From the robotics standpoint, affordances have been mainly used to relate actions to objects. Several works use affordances as prior information. A computational, cognitive model for grasp learning in infants was proposed in [8]. The affordance layer in this model provides information that helps to perform the action. Affordances have also been used as prior distributions for action recognition in a Bayesian framework [9] or to perform selective attention in obstacle avoidance tasks [10]. Several works have investigated the problem of learning affordances and their subsequent application to different tasks. In [11], the robot learned the direction of motions of different types of objects after poking and used this information, at a later stage, to recognize actions performed by others. The robot used the learned maps to push objects so as to reproduce the observed motion. A similar approach was proposed in [12], where the imitation is also driven by the effects. However, they focus on the interaction aspects and do not consider a general model for learning and using affordances. The biologically inspired behavior selection mechanism of [13] uses clustering and self-organizing feature maps to relate object invariants to the success or failure of an action. All the previous approaches learn specific types of affordances using the relevant information extracted from their sensor channels. A more complete solution has been recently proposed in [14] where the learning procedure also selects the appropriate features from a set of visual SIFT descriptors. The work in [15] focuses on the importance of sequences of actions and invariant perceptions to discover affordances in a behavioral framework. Finally, based on the formalism of [16], a goal-oriented affordance-based control for mobile robots has been presented in [17]. Previously learned behaviors such as *traverse* or *approach* are combined to achieve goal-oriented navigation.

B. Our Approach

Learning affordances from scratch (without assuming known models) can be overwhelming, as it involves relations between motor and perceptual skills, resulting in an extremely large dimension search problem. Instead, affordances can be more appropriately defined once the robot has already learned a suitable set of elementary actions to explore the world.

TABLE I
LEARNING PHASES OF THE DEVELOPMENTAL APPROACH

Sensory-Motor Coordination	1: Learn basic motor skills
	2: Develop visual perception of objects
World Interaction	3: Perception of effects and categorization
	4: Improve motor skills
	5: Learn object affordances
	6: Prediction and planning skills
Imitation	7: Perform imitation games

We adopt a developmental approach [18], [19], where the robot acquires skills of increasing difficulty on top of previous ones. As newborns, the robot should "start" with a minimal subset of core (phylogenetic) capabilities [20] to bootstrap learning mechanisms that, through self-experimentation and interaction with the environment and other humans, would progressively lead to the acquisition of new skills.

We follow the developmental roadmap proposed in [21] and extend it to include the learning and usage of affordances in the world interaction phase. This framework considers three main stages in a possible developmental architecture for humanoid robots: 1) sensory–motor coordination; 2) world interaction; and 3) imitation (see Table I). In the sensory–motor coordination stage, the robot learns how to use its motor DOFs and the coupling between motor actions and perception. In the world interaction phase, the robot learns by exploring the effects of its own actions upon elements of the environment. In the imitation phase, the robot learns by observing and imitating other agents.

Affordances are central in the world interaction phase. At this stage, the robot has already developed a set of perceptual and motor skills required to interact with the world. We introduce a general model for affordances learned by unsupervised self-exploration that includes the effects of actions on objects. Affordances are modeled with Bayesian networks (BNs) [22], a general probabilistic representation of dependencies. We formulate the problem as a structure learning algorithm, where affordances are encoded in the probabilistic relations between actions and percepts (object features and effects). This allows us to take advantage of the general methods proposed in the machine learning community for learning, inference, and planning. Indeed, the BN provides a single framework for learning and using affordances.

We used the humanoid robot Baltazar (see Fig. 3) to validate our approach. We conducted several experiments to illustrate the capability of the system to discover affordances associated with manipulation actions (e.g., grasp, tap, and touch), applied to objects with different properties (color, size, shape). The effects of these actions consist of changes perceived in the sensor measurements, e.g., persistent tactile activation for a grasp/touch, and object motion for a tap.

Our results show how the learned network captures the structural dependencies between actions, object features, and effects. The model is able to distinguish the relevant properties of the objects and discard those that do not influence action outcomes. This "feature selection" aspect of the structure learning method is fundamental in planning because task execution is often linked to object properties and only to a lesser extent to objects themselves. The learned model is then used to predict the effects of actions, recognize actions performed by a human, and to play simple interaction games. These games are driven by the observed effects of the human action and exploit knowledge contained in the affordance network to obtain the same effects. In this sense, imitation is not limited to mimicking the detailed human actions. Rather, it is used in a goal-directed manner (emulation), as the robot may choose a very different action (when compared to that of the demonstrator) provided that its experience indicates that the desired effect can be met.

In conclusion, the main contribution of this paper is a model for learning and using affordances in the context of a developmental framework for humanoid robots. The main characteristics of the proposed model are: 1) it captures the relations between actions, object features, and effects; 2) it is learned through observation and interaction with the world; 3) it detects the features that really matter for each affordance; 4) it provides a common (seamless) framework for learning and using affordances; and 5) it allows social interaction by learning from others.

C. *Structure of the Paper*

The paper is organized as follows. Section II presents the first level of the developmental architecture, which deals with sensory–motor coordination, learning of elementary actions, and basic perceptual skills. Section III describes our approach for learning and modeling affordances using BNs. This corresponds to the second developmental stage, where learning about the world is the primary motivation. Section IV shows how several imitation-like behaviors can be formulated as decision problems over the learned affordances model. The entire approach is validated in Section V through experimental tests with our humanoid platform, illustrating the advantages of affordance-based knowledge representation. Section VI draws some conclusions and establishes directions for future work.

II. Developing Basic Skills

In this section, we present the robot phylogenetic capabilities and the skills acquired during the first stage of development, sensory–motor coordination. These motor and perceptual skills, developed prior to affordance learning, provide the abstraction layer that allows the robot to start interacting with the world.

We consider that each skill develops on top of another one, following a developmental perspective. At the bottom level, we have pre-programmed skills: simple visual segmentation and categorization abilities for characterizing objects and effects (based on color, shape, motion, and orientation) and motor capabilities (near-chaotic motion and controller structures). Although innate, they are not fully operational and they still require some learning [23]. For instance, the structure of the controller is predefined, but only after learning and with the complete development of the related visual capabilities, it becomes fully functional.

From pre-programmed skills, perception develops in order to adapt to the structure of the world. Instead of dealing directly with the raw sensor data, the robot has some filters that simplify the data processing and provide higher level information such as motion detection or color segmentation. On top of this, the system learns in an unsupervised manner regularities in the world resulting, for instance, on the occurrence of particular classes of objects (colors, shapes, sizes) and effects (changes in the sensor measurement).

In parallel, the basic motor skills allow the robot to discover the relation between its actions and its proprio-perceptions. Usually, this correspondence between perception and action is called a sensory–motor map (SMM), and it can be interpreted in terms of direct/inverse kinematics of robotic manipulators. Inverse models are used to control the robot while direct ones serve for prediction purposes.

Next, we describe the different modules that allow the robot to acquire the following capabilities: 1) basic motor skills, 2) visual perception of objects, and 3) perception of effects.

A. *Basic Motor Skills*

Newborns have a series of reflexes and responses that drive their future mastering of motor abilities. However, at birth, they are still too coarse to be functional and need to develop during the first months of life. Similarly, we consider that the robot starts with a predefined set of core motor actions $M = \{m_i\}$, whose parameters (λ, ψ) must be adjusted by self-experience, i.e., by autonomous exploration of its motor and sensory capabilities. A generic description for the model m_i is

$$\dot{\Theta} = m_i(\Theta^*, y, \lambda, \psi) \qquad (1)$$

where Θ represents the controlled variables, Θ^* is the final objective, and y's are the available proprioceptive measurements of the robot. Parameters ψ are related to the kinematics/dynamics of the robot. In particular, SMMs are used to relate observations y to robot kinematics and dynamics. Parameters λ depend on the task and serve to control its execution, i.e., desired speed, energy criteria, posture. They can be tuned during affordance learning (refer to Section V, Fig. 7), but they are frozen by the system during the initial learning phase.

In this paper, we are focusing on object manipulation actions like grasping (m_1), tapping (m_2), and touching (m_3) (see Fig. 2). We consider each of these tasks consisting of three phases: 1) bringing the hand to the field of view in an open-loop fashion; 2) approaching the object using visual servoing; and 3) actually grasping, tapping, or touching the object. The two former phases are learned by self-experience (see [21] for further details), while the latter is preprogrammed due to practical limitations of our current robotic platform.

B. *Visual Perception of Objects*

Vision is the most complex perceptual system in humans and the least developed at birth. Although infant perception and cognition have been subject of extensive research in developmental psychology, there is still not much consensus on the particular developmental stages of the young infant [24]. Recent work also suggests that infants are able to start forming perceptual categories based on correlation information at the age of four months [25].

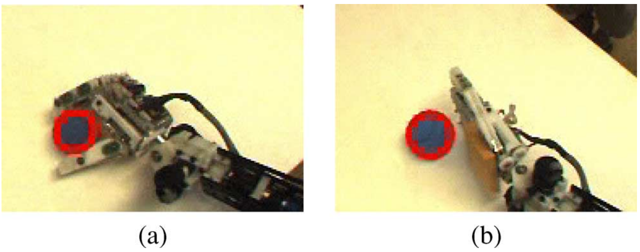


Fig. 2. Examples of actions as seen by the robot. (a) Grasping.(b) Tapping.

In this paper, we assume that the system has simple segmentation and category formation capabilities already built-in. For the sake of experimental simplicity, we have constructed a "playground" environment as shown in Fig. 3. In this environment, the robot plays with simple colorful objects over a white table, and observes other people playing with the same objects. Fast techniques like background or color segmentation are employed at this stage to allow the robot to individuate and track objects in real time. Along time, the robot collects information regarding simple visual object properties like color, shape, size, etc. Fig. 3 illustrates the robot's view of several objects, together with their color segmentation and extracted contour. After some time of interaction with the objects, the robot is able to group their properties into meaningful categories. The set of visual features employed here consist of color descriptors, shape descriptors, and size (in the image). The color descriptor is given by the hue histogram of pixels inside the segmented region (16 bins). The shape descriptor is a vector containing region-based measurements, as follows:

1) convexity—ratio between object area and convex hull area;
2) eccentricity—ratio between object minor and major axes;
3) compactness—ratio between object area and squared external contour perimeter;
4) circleness—ratio between object area and the area of the minimum enclosing circle;
5) squareness—ratio between object area and the area of the minimum enclosing rectangle.

In the category formation phase, color, shape, and size descriptors are clustered into independent categories. This allows us to make predictions on previously unseen objects, but with some properties whose affordance has already been learned.

C. Perception of Effects

In our framework, effects are defined as salient changes in the perceptual state of the agent that can be correlated to actions. For instance, upon interacting with objects, the robot may experience sudden changes of object position and velocity, and changes on tactile information related to contact. Similarly to what was carried out for object properties, effects are grouped into categories with unsupervised learning techniques. For example, after tapping an object, its velocity may be null, small, or large, depending on the object characteristics. After some time experimenting with objects and collecting information about the effects of actions on objects, the agent forms categories of effects by grouping those that are close in the sensory space. Obviously, we have to assume that the motor and perceptual capabilities of the agent are such that the same action applied to the same object will have similar effects on average. For instance, all successful grasps will have the pressure sensors persistently activated.

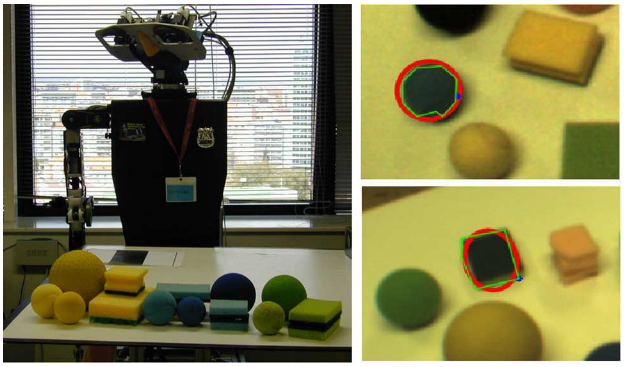


Fig. 3. Experimental setup. The Robot's workspace consists of a white table and some colored objects with different shapes (left). Objects on the table are represented and categorized according to their size, shape and color, e.g., the "ball" and "square" class (right).

All effects are processed in the same way. When the action starts, the agent observes its sensory inputs during a certain time window that depends on the action execution time and the effects duration, and records the corresponding information flow. We then fit a linear model to the temporal information and represent the observed effects by the slope and bias of the regression. For velocities (object, hand, and object–hand), the regression is made on the sequence of image velocity norms. Only the inclination is used since the bias only reflects the absolute position in the image. For the contact information, we consider only the bias (offset) of the linear regression that gives a rough measure of the duration of contact.

III. Affordance Modeling And Learning

In this section, we address the acquisition of affordances (second phase of Table I.). In the previous phase, the robot developed a set of skills that allows it to reason in a more abstract level than joint positions or raw perceptions. The robot has now available a set of actions to interact with the world and is able to detect and extract categorical information from the objects around it. We pose the affordance learning problem at this level of abstraction where the main entities are actions, objects, and effects.

We use a probabilistic graphical model known as BNs [22] to encode the dependencies between the actions, object features, and the effects of those actions (see Fig. 4). Such a representation has several advantages. It allows us to take into account the uncertainty of the real world, encodes some notion of causality, and provides a unified framework for learning and using affordances. We next describe briefly the representation, inference, and learning concepts using BNs and show how to apply them to our affordance problem.

A BN is a probabilistic directed graphical model where the nodes represent random variables $X = \{X_1, \ldots, X_n\}$ and the (lack of) arcs represent conditional independence assumptions. BNs are able to represent causal models since an arc from $X_i \rightarrow X_j$ can be interpreted as X_i *causes* X_j (see [26]). The joint distribution of the BN decomposes in the following way:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid X_{Pa(X_i)}, \theta_i) \qquad (2)$$

where $X_{Pa(X_i)}$ represents the parents of node i, i.e., the set of nodes with an arc toward X_i. The conditional probability distribution (CPD) $p(X_i \mid X_{Pa(X_i)}, \theta_i)$ of each node in the graph depends on the parents $X_{Pa(X_i)}$ and on a set of parameters θ_i. If the conditional distributions and priors are conjugate, the CPDs and marginal likelihood can be computed in closed form resulting in efficient learning and inference algorithms.

We now describe how to model the affordances using a BN and the information already learned by the robot in the previous phase. A discrete random variable $A = \{a_i\}$ models the activation of the different motor actions m_i described in Section II-A. Each action a_i is parameterized by the corresponding set of parameters λ_i as described in (1). For instance, when approaching an object to perform a grasp, the height of the hand with respect to the object or the closing angles of the hand are free parameters. It is important to note that from a sensory–motor point of view, the free parameters result in the same action. Hence, at this stage of development, the robot cannot distinguish between them, since the differences will only appear when interacting with those objects.

The object properties and effects are also modeled using discrete variables corresponding to the classes detected by the robot (see Sections II-B and Sections II-C. We denote $F_r = \{F_r(1), \ldots, F_r(n_r)\}$ and $F_o = \{F_o(1), \ldots, F_o(n_o)\}$, the descriptors extracted by each of the preprocessing modules for the robot itself and for the object o, respectively. Finally, let $E = \{E(1), \ldots, E(n_e)\}$ be the effects detected by the robot after executing an action. The set of nodes X is formed by the discrete variables A, F_r, F_o, and E, $X = \{A, F_r, F_o, E\}$.[1] The difference between object features and effects is that the former can be acquired through simple observation whereas the latter require interaction with the objects. Thus, clustering the effects correspond to the first stage of the world interaction phase and preludes the learning of the affordances.

Our final objective is to discover the relations between the random variables X representing actions, features, and objects (see Fig. 4). To do this, the robot performs an action on an object and observes the resulting effects. By repeating this procedure several times, the robot acquires a set of N trials $D = x^{1:N}$ (see Fig. 5). Let us assume for the moment that we know the dependencies, that is, the structure of the network representing affordances. Given the discrete representation of actions, features, and effects, we use multinomial distributions and their corresponding conjugate, the Dirichlet distribution, to model the CPDs $p(X_i \mid X_{Pa(X_i)}, \theta_i)$ and the corresponding parameter priors $p(\theta_i)$. According to [27], the marginal likelihood for a node X_i and its parents given D is

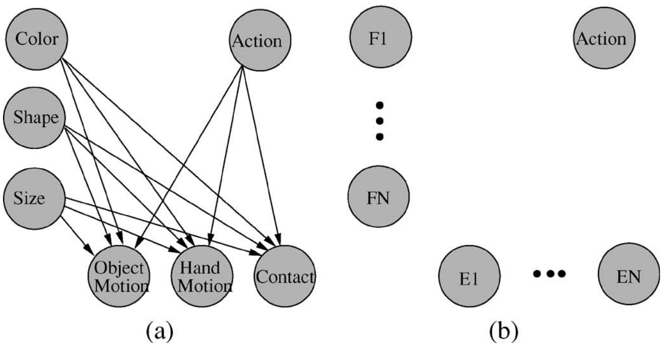


Fig. 4. BN model to represent the affordances. (a) Example of the proposed model using color, shape, and size information for the object features; and motion and contact information as effects. (b) Generic model where the nodes represent the actions A, the object features available to the robot $F(1) \cdots (n)$ and the effects obtained through the actions $E(1) \cdots E(m)$.

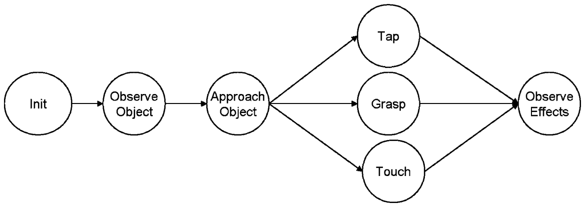


Fig. 5. Experiments protocol. The object to interact with is selected manually and the action is randomly selected. Object properties are recorded in the INIT to APPROACH transition when the hand is not occluding the object. The effects are recorded in the OBSERVE state. INIT moves the hand to a predefined position in open loop.

$$\begin{aligned} p(x_i^{1:N} \mid x_{Pa(x_i)}^{1:N}) &= \int \left[\prod_{n=1}^{N} p(x_i^n \mid x_{Pa(x_i)}^n, \theta_i) \right] p(\theta_i)\, d\theta_i \\ &= \prod_{j=1}^{|X_i|} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{|X_{Pa(X_i)}|} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \end{aligned}$$

where N_{ijk} counts the number of trials with $X_i = j, X_{Pa(X_i)} = k$, $N_{ij} = \sum_k N_{ijk}$, and Γ represents the gamma function. The pseudocounts α_{ijk} are the Dirichlet hyper parameters of the selected prior distribution of θ_i and $\alpha_{ij} = \sum_k \alpha_{ijk}$. The marginal likelihood of the data is simply the product of the marginal likelihood of each node

$$p(D \mid G) = p(x^{1:N} \mid G) = \prod_i p(x_i^{1:N} \mid x_{Pa(x_i)}^{1:N}) \qquad (3)$$

where we have made explicit the dependency on the graph structure G.

A. Learning the Structure of the Network

We are interested in learning the structure G, which is actually an instance of a model selection problem. In a Bayesian framework, this can be formalized as estimating the distribution on the possible network structures $G \in \mathcal{G}$ given the data. Using the Bayes rule, we can express this distribution as the product

[1] We represent a random variable by a capital letter X and its realizations by x.

of the marginal likelihood and the prior over graphs

$$p(G \mid D) = \eta p(D \mid G)p(G) \tag{4}$$

where $\eta^{-1} = p(D)$ is a normalization constant. The prior term $p(G)$ allows to incorporate prior knowledge on possible structures. Unfortunately, the number of BNs is super exponential with the number of nodes [28]. Thus, it is infeasible to explore all the possible graphs and one has to approximate the full solution. Markov chain Monte Carlo (MCMC) methods have been proposed to approximate the distribution $p(G \mid D)$ [29]. In our case, this can be important during the first stages of the learning to keep a set of alternative hypotheses.

When there is enough data, an alternative solution is to perform a local search to obtain the maximum likelihood structure given the data:

$$G^* = \arg\max_{\mathcal{G}} p(G \mid D). \tag{5}$$

This is a local technique, and consequently, may converge to a local minimum.

As the robot itself performs the actions, it usually obtains information of all the variables X_i. There are several algorithms to learn the structure of the network with complete data (see [30] for a review). In our experimental validation, we use the MCMC to approximate the full distribution and the hill-climbing K2 algorithm [28] to explore the neighbors using a gradient technique. Although the model also allows the robot to learn by observation, there may be some missing information. For instance, the action is not available and has to be inferred from visual measurements. In this case, the learning task is much harder and several algorithms have been proposed such as augmented MCMC or structural expectation-maximization [31].

Finally, it is important to consider causality. The previous learning schemes are able to distinguish among equivalent classes.[2] So as to be able to infer the correct causal dependency, it is necessary to use interventional data where we fixed some of the variables to a specific value to disambiguate between graphs in the same equivalent class.

In the case of a robot interacting with its environment, there are several variables that are actively chosen by the robot: the action and the object. These variables are actually interventional since they are set by the robot to their specific values at each experience. Interventional data are currently an important research topic within BN learning algorithms (see [32]). Under the assumption of a perfect intervention of node i, the value of $X_i = x_i^*$ is set to the desired value and its CPD is just an indicator function with all the probability mass assigned to this value $p(X_i \mid X_{Pa(X_i)}, \theta_i) = I(X_i = x_i^*)$. As a result, the variable X_i is effectively cut off from its parents $X_{Pa(X_i)}$.

B. Parameter Learning and Inference

Once the structure of the network has been established, the parameters θ_i of each node are estimated using a Bayesian approach [30]. The estimated parameters can still be sequentially updated online allowing the incorporation of the information provided by new trials.

Since the structure of the BN encodes the relations between actions, object features, and effects, we can now compute the distribution of a (group of) variable(s) given the values of the others. The most common way to do this is to convert the BN into a tree, and then, apply the junction tree algorithm [33] to compute the distribution of interests. It is important to note that it is not necessary to know the values of all the variables to perform inference.

Based on these probabilistic queries, we are now able to use the affordance knowledge to answer the questions of Fig. 1 simply by computing the appropriate distributions. For instance, the prediction of the effects when observing an action a_i on given observed object features f_j is just $p(E \mid A = a_i, F = f_j)$. The query can combine features, actions, and effects both as observed information and as the desired output.

IV. Interaction Games

After interacting with the objects, the robot is ready to start the social phase of its development. In this section, we show how to use affordances in this context. Imitation, as a word used in everyday language, refers to many different behaviors. In biology, imitation aims to achieve the same effect by copying the actions of the demonstrator [34]. This requires to solve the *body correspondence problem* [35], i.e., the correspondence between the demonstrator's actions and the learner's ones. Another common behavior is emulation. In this case, the objective is to match the resulting effect [34]. This means that the learner can choose different actions from those of the demonstrator as long as it achieves the same effect. Indeed, for many authors [36], emulation is strongly related with affordances. This is because affordances provide the means to relate actions to effects. In this paper, we use imitation to refer to this last behavior.

We describe next a set of interaction games between a human and a robot. In each game, the robot observes a human performing an action on an object. Then, the robot is presented with another object or objects and has to perform a compatible action. More formally, let a^d be the action performed by the demonstrator, f^d the features of the object and e^d the resulting effect. We pose the problem as a one-step Bayesian decision problem where a reward (cost) function r defines the objective of the imitation task. The function to optimize is

$$< a^*, o^* >= \underbrace{\arg\max}_{a \in A, o \in \mathcal{O}} \mathrm{E}\left[r(a^d, f^d, e^d, a, f^o, e^o)\right] \tag{6}$$

where f^o and e^o represent the object features and effects of action a. The maximization is over the set of possible actions A and possible objects $\mathcal{O}$. Since the knowledge about the actions, objects, and effects is not deterministic, we need to take the expectation E [] over the reward function. In particular, the probability of the effects of a particular action–object pair $p(E \mid A, O)$ is encoded by the affordance network presented in Section III. For the sake of simplicity, in the remainder of the section, we use the maximum likelihood estimation $\hat{f^d}$ and $\hat{e^d}$ of object features

[2] Two directed acyclic graphs G and G' are equivalent, if for every BN, $B = (G, \Theta)$, there exist another network $B' = (G', \Theta')$ such that they define the same probability distribution.

and effects. We present examples of simple imitation behaviors to illustrate the previous formulation.

1) *Matching of effects*: The objective of this behavior is to achieve the same effect as observed when a single object is present. The reward function is

$$r(e^d) = \begin{cases} 1, & \text{if} \quad E^i = \hat{e}^d \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $\hat{e}^d$ is the most likely effect detected by the robot. Since the reward does not depend on the object or the features, the general expression simplifies to

$$a^* = \arg\max_a rp(E^i = \hat{e}^d | a, f^i) \tag{8}$$

where f^i are the features of the object.

2) *Matching of effects and object selection*: We now describe the more complex situation, where the robot has to select among a set of objects $\mathcal{O}$. If we do not care about the object features f^d, this simply requires the inclusion of the available objects in the optimization

$$< a^*, o^* >= \underbrace{\arg\max}_{a, o_i \in \mathcal{O}} rp(E^i = \hat{e}^d | a, f^{o_i}) \tag{9}$$

where f^{o_i} represent the features of object o_i.

3) *Matching of effects and object features*: The last behavior adds information about the object features in the cost function. This allows the favoring of those objects similar to the one used by the demonstrator. The cost function has the following expression:

$$r(e^d, f^d, f^i) = \begin{cases} 1, & \text{if} \quad E^i = \hat{e}^d \wedge F^i = \hat{f}^d \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Notice that one could weigh the features giving different rewards to different object features. For instance, if the desired object is a big ball, we could weigh the sizes as a function of their distance in the space of the measurements to the class model. Since the current observations of the robot are not deterministic, the expectation of (6) is now also taken over the possible classes of each of the available objects. The resulting expression is

$$< a^*, o^* >= \underbrace{\arg\max}_{a, o_i \in \mathcal{O}} rp(E^i = \hat{f}^d | a, f^{o_i}) p(F^{o_i} = \hat{f}^d) \tag{11}$$

where $p(F^{o_i} = \hat{f}^d)$ represents the likelihood of the features of o_i being equal to the features $\hat{f}^d$. Again this probability is computed based on the clusters of each dimension using a metric on the space of each feature.

V. Experiments

In this section, we present a set of experimental results to illustrate the acquisition and usage of affordance knowledge. We used Baltazar, a 14 DOFs humanoid torso composed by a binocular head and an arm. Using the motor skills of Section II, Baltazar is able to perform three different actions $A = \{a_1 = \text{grasp}(\lambda), a_2 = \text{tap}(\lambda), a_3 = \text{touch}(\lambda)\}$ where λ represents the height of the hand in the 3-D workspace when reaching the object in the image. The robot applies its actions on a set of different objects with two shapes (box and ball) with four colors and three sizes (see Fig. 3).

We recorded a set of 300 experiments following the protocol depicted in Fig. 5. At each trial, the robot was presented with a random object. Baltazar randomly selected an action and approximated its hand to the object using the algorithms of Section II-A. When the reaching phase is completed, it performed the selected action [grasp(λ) or tap(λ) or touch(λ)], and finally, returned the hand to the initial location. During the action, the object features and effects are recorded.

We used the data of these trials to implement steps 3 to 7 of Table I. In this paper, we assume that the motor skills have already been learned as presented in Section II, for details refer to [21]. Next, we present the results for the different steps allowing the robot to evolve from basic sensory–motor coordination to imitation capabilities.

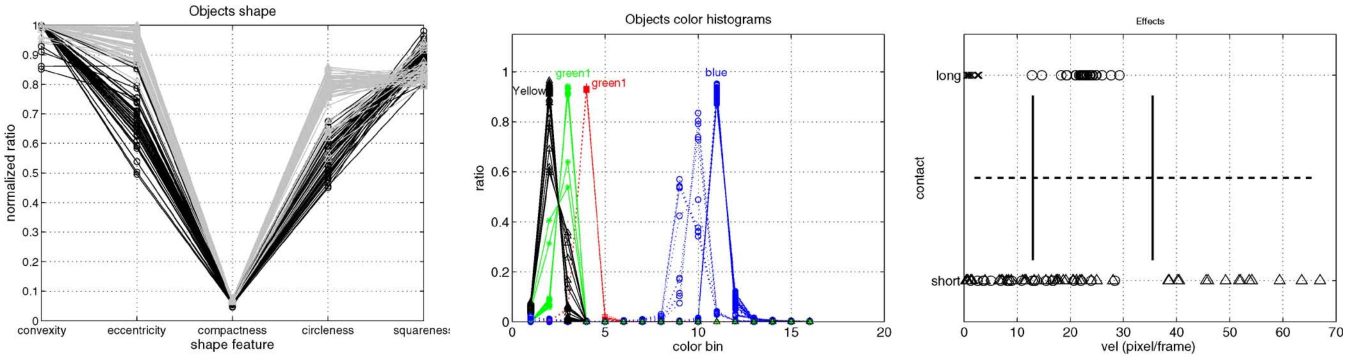
A. Discretization of Perceptual Information

This step plays an important role since it is the basis of the discretization used in the affordance learning algorithms. In our example, we used the three features described in Section II: color, shape, and size. Each one is modeled as a *n-dimensional* vector space. Since our setup is clearly discrete, we applied the *X-means* algorithm [37] to detect clusters in the space of each object feature and in the effects. For the continuous free parameters λ of the actuators such as height of the wrist, we discretized them with a predefined resolution.

It is important to note that the final objective is to learn the affordances given a set of available motor and perceptual skills, not to make a perfect object classification. Indeed, the clustering contains some errors due to different illumination conditions. For instance, the features of some objects were misclassified, and the affordance learning has to cope with this noise.

Fig. 6(a) shows the results of the *X-means* algorithm for the object shape. The two resulting clusters separate easily balls from boxes based mostly on circleness and eccentricity descriptors. Fig. 6(b) gives the equivalent result for colors where the features vector is an histogram of the *hue*. As the objects have uniform color, each histogram has only one salient peak. Finally for the unidimensional size, three clusters were enough to represent five different sizes of the objects presented to the robot.

Fig. 6(c) shows the classes of object velocities and contact patterns detected by the robot following the procedure described in Section II-C. Roughly speaking, a grasp action resulted in medium velocity (except in one case where the ball fell down the table), tap produced different velocity patterns depending on the shape and size of the object, and touch has small velocities. Also, contact information lasted longer for grasp and touch actions than for tap ones. The combination of the different features produces patterns in the feature space that are used to infer statistical dependencies and causation. Table II summarizes the clustering results for the different variables and provides the notation used in the remainder of this section.

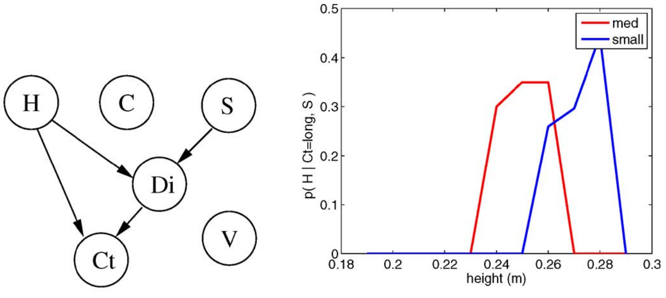


(a) (b) (c)

Fig. 6. Clustering of object features and effects. (a) Shape description of the objects. Five features: convexity, eccentricity, compactness, circleness, and squareness describe the objects. In the experiments box and balls were clustered automatically. Different clusters are represented by circles or plus signs. (b) Color histograms with the corresponding clusters. Each bin relates to a given Hue value.The clusters correspond to: $yellow$, $green_1$, $green_2$, and $blue$. (c) Clustering of object velocity and contact. For each observation grasp is represented by x, tap by $\triangle$, and touch by $\circ$. The vertical lines show the clusters boundaries for velocity and the horizontal line for contact.

TABLE II
Summary of Variables and Values

Symbol	Description	Values
A	Action	*grasp, tap, touch*
H	Height	discretized in 10 values
C	Color	*green*$_1$*, green*$_2$*, yellow, blue*
Sh	Shape	*ball, box*
S	Size	*small, medium, big*
V	Object velocity	*small, medium, big*
HV	Hand velocity	*small, medium, big*
Di	Object-hand velocity	*small, medium, big*
Ct	Contact duration	*none, short, long*



(a) (b)

Fig. 7. Tuning the height for grasping a ball. (a) Dependencies discovered by the learning algorithm. The action and shape for this example are fixed and color does not have an impact on the effects. Node labels are shown in Table II. (b) CPD of height given the robot obtained a long contact (successful grasp).

B. *Affordances*

Based on the previous descriptors of actions and its parameters, features and effects, we present two different experiments to illustrate the ability of the proposed model to capture the affordances. We would like to remark that the robot does not receive any information about the success of the actions. The interest is in understanding the effects obtained by the actions in an unsupervised manner.

The objective of the first experiment is to find the influence of a free parameter of an action. The robot tries the action for different configurations of the free parameters. For a grasp, these parameters are the joint angles of the fingers and height of the hand. The former is used after reaching the object in the closing of the hand, whereas the latter is a free parameter of the SMM used to approximate the hand to the object. We used the K2 algorithm[3] to find the maximum likelihood graph with a random starting point and BDeu priors [27] to give uniform priors to different equivalence classes.

Fig. 7(a) shows how the resulting network captures the dependency of the effects on these parameters. More interestingly, the CPDs provide the probability of producing different effects according to the values of the free parameters. Fig. 7(b) shows the estimated probability of each height conditioned on observing a long contact for medium and small objects (which is the sign of a successful grasp). Since big objects cannot be grasped by the robot's hand, all heights have zero probability for this class. Please note that the distribution of Fig. 7(b) can be directly used to adjust the height of the action for different object sizes.

The objective of the second experiment is to show how the robot is able to distinguish the effects of different actions and simultaneously select those features that are interesting for this purpose. Also, we illustrate the differences between the MCMC estimation of the distribution of possible networks and the maximum likelihood solution provided by the K2 algorithm. In both cases, we use BDeu priors for the graphs and random initialization. Although, one can use conditional independence tests to provide a rough initialization for both algorithms, in our case, we got similar results using randomly generated networks. For the MCMC algorithm, we used 5000 samples with a burn-in period of 500.

Fig. 8(a)–(d) shows the network computed by the K2 algorithm and the three most likely networks computed by the MCMC. For this particular case, there are no major differences between both models. However, in the initial steps, the MCMC probability distributions represent the uncertainty on the model selection problem. When using longer datasets, the probability

[3] The implementation of the algorithms is based on the BNT toolbox for Matlab, http://bnt.sourceforge.net/.

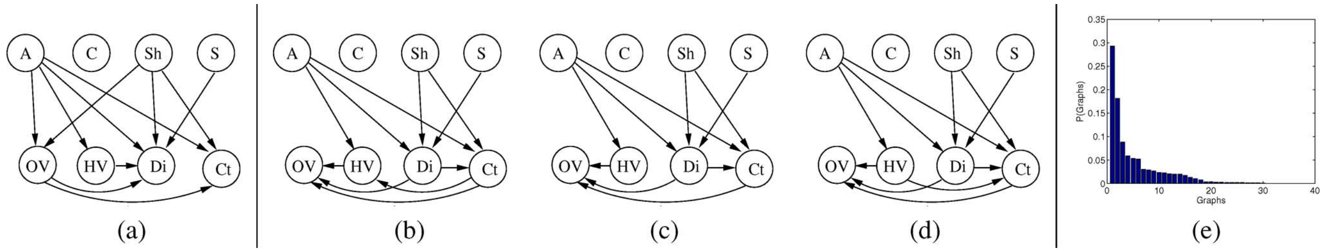


Fig. 8. This figure shows the affordance model estimated by the K2 algorithm and the MCMC. Node labels are shown in Table II. (a) K2 maximum likelihood network. (b)–(d) Three more likely networks obtained by the MCMC for the same data. (e) Posterior probability over graphs computed by the MCMC.

mass concentrates on a single group of very similar networks but it still maintains a set of plausible networks that capture correct relationships between the variables. Fig. 8(e) shows the posterior probability of all the sampled models. Note that in the example, the posterior probability of the K2 model is lower than 0.05 according to the distribution computed by the MCMC. In some situations, due to its greedy approach, we found that the K2 algorithm converged to a model that lacked some relevant relation such as the dependency on the shape of the object. Nonetheless, for most cases, the K2 algorithm converges to a reasonable model even for little data. The price to pay, when approximating the distribution of possible networks, is a higher computational cost for the MCMC algorithm.

Although there is no ground truth to compare the estimated networks, we see that color has been detected as irrelevant when performing any action. Shape and size are important for grasp, tap and touch since they have an impact on observed velocities and contact. In order to show the convergence of the network toward a plausible model, we have estimated a network for different numbers of trials. For each number, we have randomly created 100 datasets from the complete dataset, estimate the posterior over graphs using the MCMC and compute the likelihood of the whole data for the most likely model. Fig. 9 shows how the marginal likelihood of the data converges as the number of trials increases. The figure also indicates that, after 100 trials, the improvement of the likelihood of the data, given more experiments, is very small since the model already was able to capture the correct relations. On the other hand, for the K2 model of Fig. 8(a), the marginal likelihood is -2775 which is lower than the one attained by the MCMC algorithm.

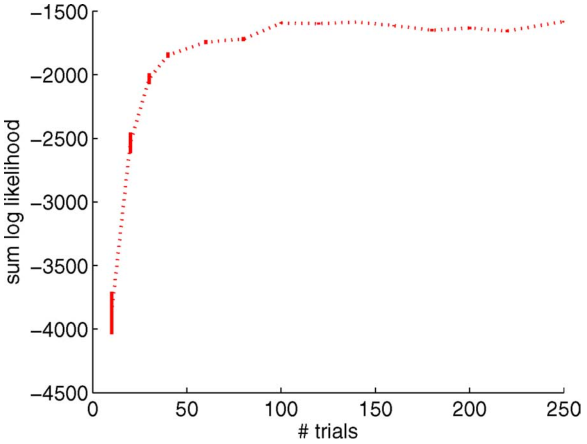


Fig. 9. Marginal likelihood of the data given the learned network as the number of trials increases. The vertical bars show the variance of the likelihood.

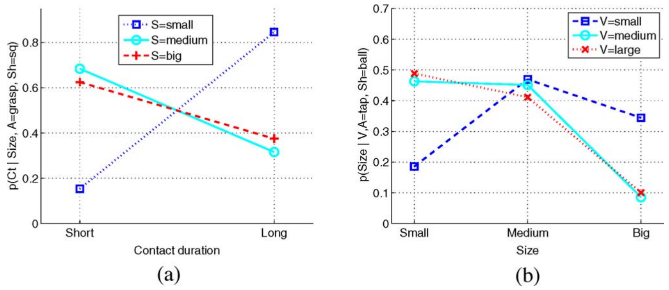


Fig. 10. Examples of CPD for the learned network: (a) $p(Ct \mid S = s_i\,, A = \mathrm{grasp}, Sh = \mathrm{sq})$, the CPD of contact duration given a grasp was performed on a box for every value of size. (b) $p(S \mid V = v_i\,, A = \mathrm{tap}, Sh = \mathrm{ball})$, the CPD of the size of a ball given the action was a grasp for every possible value of velocity.

The actual dependencies are encoded in the multinomial CPDs of each node. Based on the most probable hypothesis generated by the MCMC algorithm, we compute the maximum likelihood parameters using the same dataset. To validate the network actually captures the correct dependencies, we compute some illustrative CPDs. Fig. 10(a) presents the predicted contact duration of a grasp action for different sizes. It basically states that successful grasps (longer contact between the hand and the object) occur more often with small objects than with bigger ones. Fig. 10(b) shows the distribution of size after performing a tap on a ball for different velocities. According to it, small balls move faster than bigger ones and medium ball velocities are highly unpredictable (similar likelihood for all velocities). This actually reflects the behavior of the objects during the trials. For instance, the mean and variance of the ball velocity (μ[pixel/frame], σ^2[pixel^2/frame^2]) were (33.4,172.3), (34.3,524.9), and (17.5, 195.5) for a small, medium, and big balls, respectively.

In order to further validate the model, we have performed a leave one out cross validation to evaluate the action recognition capabilities of the network. For each trial, we computed the network structure and parameters using the other trials and the MCMC algorithm. We then estimated the probability of each action given the object features and the object velocity, hand velocity, and object–hand velocity. Since contact is a proprioceptive measurement, it is not usually available when observing other's actions. The most likely action was correct in more than

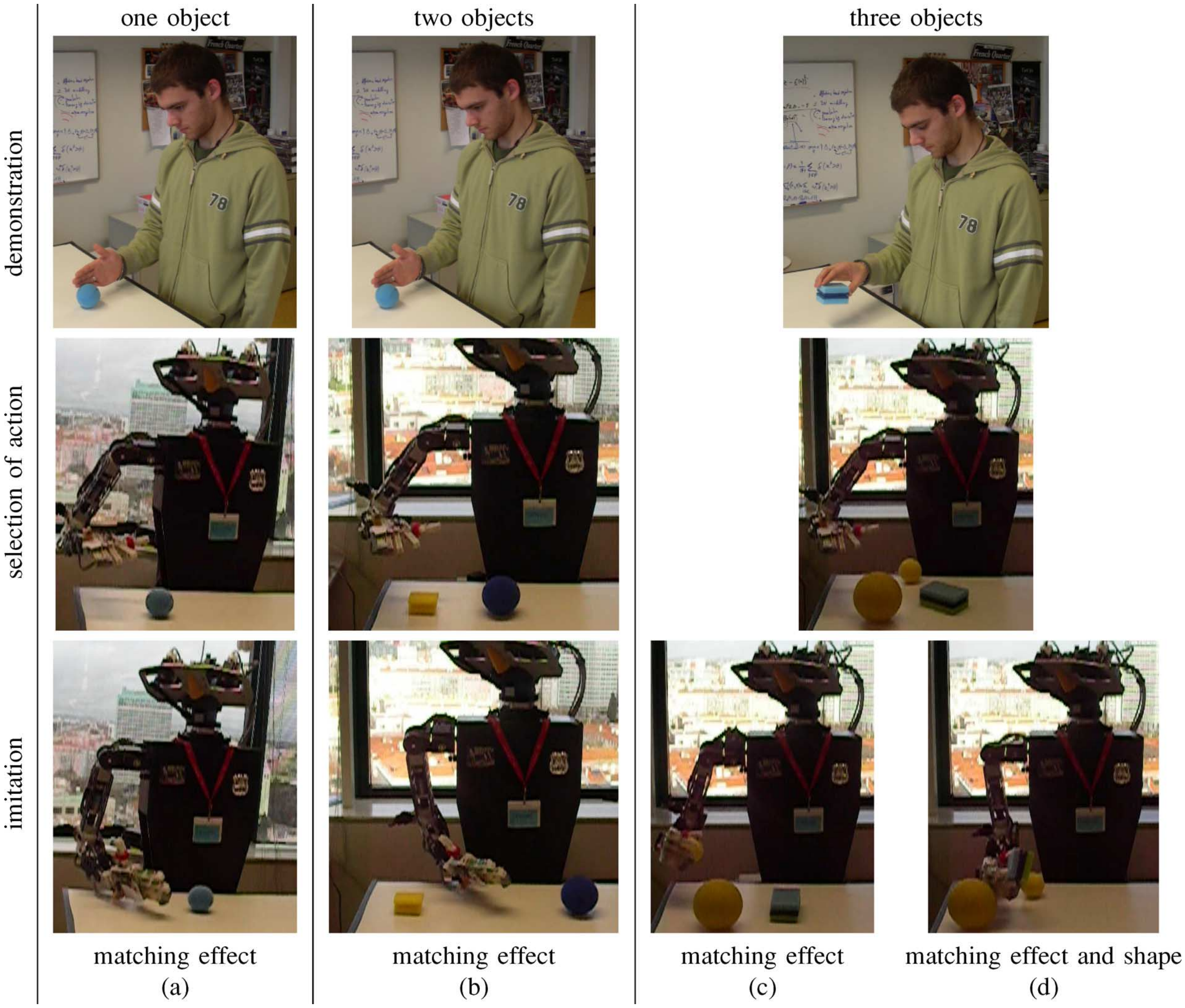


Fig. 11. Different imitation behaviors. Top: demonstration, middle: set of potential objects, bottom: imitation. Situations (a)–(d) represent imitation of actions. (a) Matching effect. (b) Matching effect. (c) Matching effect. (d) Matching effect and shape.

85% of the cases. The errors were mainly due to the absence of contact information. After touching or tapping boxes, the remaining effects are very similar. If contact was included, the ratio of correct recognition was 98%.

Summarizing, we have shown how the robot can tune its motor controllers through experimentation by including the effects of its actions. Once this information is available, it starts establishing relationships between the features of the objects and the resulting effects of its actions. The model can then easily be used to perform simple inference, prediction, and planning. The learning depends on the motor and perceptual skills and is done in a completely unsupervised manner. There is no notion of success or failure, and the network may not be able to distinguish between nonseparable objects given the used descriptors. However, it constructs a plausible model of the behavior of the different objects under different actions.

C. Interaction Games

Finally, we present results on basic interaction games using the affordance network. In this case, the robot observes a person performing an action on a given object. Then, using one of the functions described in Section IV, it selects an action and an object to imitate (emulate) the human. Fig. 11 depicts the demonstration, the objects presented to the robot, and the selected action and object for different reward functions.

We used two different demonstrations, a tap on a small ball resulting in high velocity and medium hand–object distance, and a grasp on a small square resulting in small velocity and small hand–object distance. Notice that contact information is not available when observing others.

The objective of the robot is to obtain the same observed effects. The first situation [see Fig. 11(a)] is trivial as only tap has a nonzero probability of producing a high velocity. Hence, the imitation function selected a tap on the only available object. In Fig. 11(b), the demonstrator performed the same action, but the robot had to decide between two different objects. Table III shows the probabilities for the desired effects given the six possible combinations of actions and objects. The robot selected the highest probability and performed a tap on the ball.

TABLE III
PROBABILITY OF ACHIEVING THE DESIRED EFFECTS FOR EACH ACTION AND THE OBJECTS OF FIG. 11(B)

obj \ action	grasp	tap	touch
Blue, big, ball	0.00	0.20	0.00
Yellow, small box	0.00	0.06	0.00

Fig. 11(c) and Fig. 11(d) illustrate how including the object features in the reward function results in different behaviors. After observing the grasp demonstration, the robot had to select among three objects: yellow big ball, yellow small ball, and blue small box. In the first case, the objective was to obtain the same effects (8). The probability for each of the objects is 0.88, 0.92, and 0.52, respectively, and the robot grasped the yellow small ball even if the same object was also on the table [see Fig. 11(c)]. Notice that this is not a failure since it maximizes the probability of a successful grasp that is the only requirement of the reward function. As described in Section IV, we can include object information within the reward function of the robot using (9). For instance, when the reward was modified to include a similar shape, the robot selected the blue box instead [see Fig. 11(d)].

VI. CONCLUSION

This paper addresses the learning and usage of affordances, i.e., the relations between actions, objects, and effects. We used BNs as a general tool to capture these dependencies and to infer causality relationships by taking advantage of the intervention of the robot and the temporal ordering of the events. Most previous works assumed that the dependencies were known and learned a mapping between pairs of actions and objects or used supervised approaches. Our affordance model does not assume any prior knowledge on the dependencies and tries to infer the graph of the network directly from the exteroceptive and proprioceptive measurements. In addition to affordance learning, the model also allows the robot to tune the free parameters of the controllers. By using Bayesian inference, the robot is able to predict actions, objects features or effects using the available information at a given point in time. Planning and basic imitation behaviors are also posed as a Bayesian decision problem to maximize a reward function.

We have integrated the previous model within a developmental architecture where the robot incrementally develops its skills. We argue that affordances are the bridge between sensory–motor coordination, world understanding and imitation. Affordances not only describe agent–object interactions, but they also provide an interpretation of the observed action in terms of equivalent effects in the robot's body, allowing the robot to emulate others.

Based on the proposed framework, there are plenty of opportunities for future research. Biological systems develop many of their different skills in parallel. We are now investigating how to dynamically incorporate new robot capabilities (actions) or world knowledge in the learning algorithms. Although the proposed model can directly learn through observation of other agents, it is necessary to develop mechanisms to update the knowledge sequentially and to deal with new actions or effects. Finally, more complex plans are required that include temporal dependencies of sequences of actions.

REFERENCES

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin, 1979.

[2] M. Lopes, F. S. Melo, and L. Montesano, "Affordance-based imitation learning in robots," in *Proc. IEEE—Intell. Robot. Syst. (IROS 2007)*, San Diego, CA, pp. 812–814.

[3] S. Schaal, "Is imitation learning the route to humanoid robots," *Trends Cognitive Sci.*, vol. 3, no. 6, pp. 233–242, 1999.

[4] A. Chemero, "An outline of a theory of affordances," *Ecol. Psychol.*, vol. 15, no. 2, pp. 181–195, 2003.

[5] J. Konczak, H. Meeuwsen, and M. Cress, "Changing affordances in stair climbing: The perception of maximum climbability in young and older adults," *J. Exp. Psychol.: Human Percept. Perform.*, vol. 19, pp. 691–697, 1992.

[6] E. Symes, R. Ellis, and M. Tucker, "Visual object affordances: Object orientation," *Acta Psychologica*, vol. 124, no. 2, pp. 238–255, 2007.

[7] M. Turvey, K. Shockley, and C. Carello, "Affordance, proper function, and the physical basis of perceived heaviness," *Cognition*, vol. 73, no. 2, pp. 17–26, Dec. 1999.

[8] E. Oztop, N. Bradley, and M. Arlib, "Infant grasp learning: A computational model," *Exp. Brain Res.*, vol. 158, pp. 480–503, 2004.

[9] M. Lopes and J. Santos-Victor, "Visual learning by imitation with motor representations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 3, pp. 438–449, Jun. 2005.

[10] A. Slocum, D. Downey, and R. Beer, "Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention," presented at the Conf. Simul. Adaptive Behavior, Paris, France, 2000.

[11] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action: Initial steps towards artificial cognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 14–19, 2003, vol. 3, pp. 3140–3145.

[12] H. Kozima, C. Nakagawa, and H. Yano, "Emergence of imitation mediated by objects," presented at the 2nd Int. Workshop Epigenetic Robot., Edinburgh, Scotland, 2002.

[13] I. Cos-Aguilera, L. Cañamero, and G. Hayes, "Using a SOFM to learn object affordances," presented at the Workshop Phys. Agents (WAF), Girona, Spain, 2004.

[14] G. Fritz, L. Paletta, R. Breithaupt, E. Rome, and G. Dorffner, "Learning predictive features in affordance based robotic perception systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 2006, pp. 3642–3647.

[15] A. Stoytchev, "Behavior-grounded representation of tool affordances," presented at the Int. Conf. Robot. Autom., Barcelona, Spain, 2005.

[16] E. Sahin, M. Cakmak, M. Dogar, E. Ugur, and G. Ucoluk, "To afford or not to afford: A new formalization of affordances towards affordance-based robot control," *Adaptive Behav.*, vol. 15, no. 4, pp. 447–472, 2007.

[17] M. Dogar, M. Cakmak, E. Ugur, and E. Sahin, "From primitive behaviors to goal-directed behavior using affordances," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, 2007, pp. 729–734.

[18] J. Weng, "The developmental approach to intelligent robots," presented at the AAAI Spring Symp. Series, Integr. Robot. Res.: Taking Next Leap, Stanford, CA, Mar. 1998.

[19] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey," *Connection Sci.*, vol. 15, no. 40, pp. 151–190, Dec. 2003.

[20] E. Spelke, "Core knowledge," *Amer. Psychol.*, vol. 55, pp. 1233–1243, 2000.

[21] M. Lopes and J. Santos-Victor, "A developmental roadmap for learning by imitation in robots," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 308–321, Apr. 2007.

[22] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[23] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett, *Rethinking Innateness: A Connectionist Perspective on Development*. Cambridge, MA: MIT Press, 1996.

[24] L. B. Cohen, "An information-processing approach to infant perception and cognition," in *The Development of Sensory, Motor and Cognitive Capacities in Early Infancy*, F. Simon and G. Butterword, Eds. East Sussex, U.K.: Psychology, 1999, pp. 277–300.

[25] G. W. D. Mareschal, D. Powell, and A. Volein, “Evidence of rapid correlation-based perceptual category learning by 4-month-olds,” *Infant Child Develop.*, vol. 14, pp. 445–457, 2005.

[26] J. Pearl, *Causality: Models, Reasoning and Inference*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[27] D. Heckerman, D. Geiger, and M. Chickering, “Learning bayesian networks: The combination of knowledge and statistical data,” *Mach. Learning*, vol. 20, pp. 197–243, 1995.

[28] G. Cooper and E. Herskovits, “A bayesian method for the induction of probabilistic networks from data,” *Mach. Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[29] D. Madigan and J. York, “Bayesian graphical models for discrete data,” *Int. Stat. Rev.*, vol. 63, pp. 215–232, 1995.

[30] D. Heckerman, “A tutorial on learning with bayesian networks,” in *Learning in Graphical Models*, M. Jordan, Ed. Cambridge, MA: MIT Press, 1998.

[31] N. Firedman, “The bayesian structural EM algorithm,” in *Uncertainty in Artificial Intelligence, UAI*. San Francisco, CA, 1998.

[32] D. Eaton and K. Murphy, “Exact Bayesian structure learning from uncertain interventions,” in *Proc. 11th Int. Conf. Art. Intell. Statist. AI, Statist.*, Puerto Rico, 2007.

[33] C. Huang and A. Darwiche, “Inference in belief networks: A procedural guide,” *Int. J. Approximate Reason.*, vol. 15, no. 3, pp. 225–263, 1996.

[34] J. Call and M. Carpenter, “Three sources of information in social learning,” in *Imitation in Animals and Artifact*. Cambridge, MA: MIT Press, 2002.

[35] C. L. Nehaniv and K. Dautenhahn, in *Imitation in Animals and Artifacts,* ch. 2*, The Correspondence Problem*. Cambridge, MA: MIT Press, 2002.

[36] M. Tomasello, “Emulation learning and cultural learning,” *Behav. Brain Sci.*, vol. 21, pp. 703–704, 1998.

[37] D. Pelleg and A. W. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” presented at the Int. Conf. Mach. Learning, San Francisco, CA, 2000.

Luis Montesano received the Ph.D. degree in computer science from the University of Zaragoza, Zaragoza, Spain, in 2006.

He is currently a Researcher at the Institute of Systems and Robotics (ISR), Lisbon, Portugal. He has participated in various international research projects in the areas of mobile robotics and cognitive systems. His current research interests include Bayesian statistics, robotics, learning and decision making.

Manuel Lopes received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2006.

He is currently a Researcher at the Institute of Systems and Robotics (ISR), Lisbon. He has participated in various international research projects in the areas of robotics and cognitive systems. His current research interests include robotics, development, computer vision, and human–robot interaction.

Alexandre Bernardino (M’05) received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2004.

He is an Assistant Professor at the IST, and a Researcher at the Computer Vision Laboratory (VisLab), Institute for Systems and Robotics (ISR), Lisbon. He has participated in several national and international research projects in the fields of robotics, cognitive systems, computer vision, and surveillance. He is the author or coauthor of several published articles in international journals and conferences. His current research interests include application of computer vision, cognitive science, and control theory to advanced robotics and automation systems.

José Santos-Victor (S’86–A’88–M’00) received the Ph.D. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 1995.

He is currently an Associate Professor in the Department of Electrical and Computer Engineering, IST, and a Researcher at the Computer and Robot Vision Laboratory (VisLab), Institute of Systems and Robotics (ISR). He is also responsible for the participation of the IST in various European and National research projects in the areas of computer vision and robotics. His current research interests include areas of computer and robot visions, particularly in the relationship between visual perception and the control of action, biologically inspired vision and robotics, cognitive vision and visual controlled (land, air, and underwater) mobile robots.

Dr. Santos-Victor is an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS.