Bootstraping manipulation skills to learn affordances in open-ended environments*

Carlos Maestre[1], Christophe Gonzales[2] and Stephane Doncieux[1]

A cause-effect relation is called *causality* [1]. The causality relations with objects in an environment are known as *affordances*, defined as the acquired cause-effect relation of applying a *behaviour/action* on an *entity/object* to obtain an *effect* [2]. Affordance knowledge is part of the expertise needed by robots to properly perform tasks in an environment. This expertise is threefold: (i) knowledge related to the robot itself, associated to the concept of *self* and some basic motion capabilities, i.e., its kinematic model; (ii) knowledge related to interpreting its environment, e.g., segmentation, identification and tracking of objects; (iii) capabilities to interact with the environment, composed by (a) affordance knowledge, to infer the right actions to perform given an environmental setup; and (b) manipulation skills, motion primitives as grasp and push, to execute those actions.

To the best of the authors' knowledge, in the literature related to affordances the experimental setups are *closed*, i.e. composed by a set of preselected objects. In these environments robots are usually provided with both the information about themselves and the information to interpret the environment, as acquiring the corresponding skills is (an extended review of this literature is available in [3]). Regarding the interaction with the environment, robots are also provided with a set of predefined motion primitives adapted to produce effects on the previously selected objects. For example, a robot could be asked to clutter a table with some objects on it. Therefore, it should be provided with a set of primitives to be able to properly arrange them.

However, in daily situations, environments are usually *open-ended*, i.e., only partially known, with incomplete information. In these environments the definition of general purpose primitives must be complex, due to the unforeseen situations a robot can face. On the previous cluttering scenario, if a soft object, as a cuddly toy, is put on the table the robot would not be able to adapt to the new environment, i.e., it would not be able to produce a desired effect (e.g., push to the left) on this object. Developmental Robotics aims at making robots acquire their own knowledge through their interaction with the open-ended environment thanks to an active exploration, adapting to new situations, similar to what infants do [4].

We propose an iterative methodology for a robot to learn affordances in open-ended environments. A robot recursively interacts with its environment, constantly adapting to new situations and extending its knowledge about its content and how to interact with it. In this methodology two steps are executed iteratively: *information acquisition* and *knowledge exploitation*. In the information acquisition step a robot interacts in an unsupervised fashion with its environment to extend its information about it, i.e., to generate/enhance a dataset of action execution and object-related cause-effect information. In the knowledge exploitation step the robot uses the information obtained during the exploration to learn (i) different manipulation skills to interact with the environment, (ii) the available affordances, and (iii) to evaluate their feasibility computing a *score* based on the effects obtained applying the available affordance knowledge. Henceforth, based on the results of the evaluation the information acquisition step can be newly executed to gather more object-related information to improve the score, continuing the iterative process.

In the knowledge exploitation step the information within the dataset is discretized. This discretization is necessary to produce affordance knowledge suitable for posterior use in high level reasoning processes to execute tasks, e.g., in a knowledge processing framework as ORO; or for the planning of complex tasks composed of simple actions [3]. The discretization to be applied to the dataset can be predefined, or it can be learned during the iterative process, identifying one or more discretization configurations providing a high score. For that purpose, before starting the iterative process several discretization configurations can be defined, randomly or based on some heuristics as landmarks [5], providing different performance values after several iterations of the methodology. Configurations with high scores would be able to generate a high number of desired effects based on the learned affordances.

Based on this iterative methodology we propose Adaptive Affordance Learning (A^{2}L), a method to adapt the learning of affordances in open-ended environments, i.e., environments with incomplete information. This method relies on a Bayesian Network (BN) [1] to represent the discrete affordances, as in Montesano et al. [6]. The reasons for this choice are twofold: (i) because it has a strong inference capability; and (ii) because its internal representation of affordances as a probabilistic representation of dependencies

*This research is sponsored in part by the DREAM project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 640891. This work was performed within the Labex SMART (ANR-11-LABX-65) supported by French state funds managed by the ANR within the Investissements d'Avenir programme under reference ANR-11-IDEX-0004-02.

[1]Carlos Maestre and Stephane Doncieux are with UMR 7222, ISIR, Sorbonne Universites, UPMC Univ Paris 06 and UMR 7222, CNRS, ISIR, Paris, France {maestre, doncieux}@isir.upmc.fr

[2]Christophe Gonzales is with UMR 7606, LIP6, Sorbonne Universites, UPMC Univ Paris 06 and UMR 7606, CNRS, LIP6, Paris, France christophe.gonzales@lip6.fr

allows to analyse and understand the outcomes of learning.

A^2L has two phases: (a) an initial *basic learning phase*, to identify and validate a primary set of affordances of the environment; (b) and an *extended learning phase*, to extend this set and rectify possible incorrect affordances. Each one of these phases includes the aforementioned two iterative steps.

a) Basic learning phase: a naive babbling is initially performed to gather, in a raw dataset, an initial set of interactions between a robot and its environment. This babbling can be generated randomly, or it can be driven by some goal-oriented method, as Novelty-driven Evolutionary Babbling (NovEB) [7], the method relying on Novelty Search to explore possible robot's movements, while focusing on those that generate the highest novelty from the perception point of view. Afterwards, a learning process is performed to learn the relation among the actions and the effects, i.e. the affordances. To that end, the raw dataset is discretized based on a discretization configuration. Once the discretized dataset is available, both the structure and parameters of the BN are learned. Finally, a task-oriented assessment of the learned affordances is realized to measure their robustness. This evaluation is predefined, and made up with a set of affordances available in the setup. During the evaluation the robot must try to reproduce the previous set of affordances. For each reproduced affordance a predefined numerical value is computed related to the effect obtained, called *performance value*. Also a final *score* is computed, result of the addition of all the previous individual performance values. At the end of this phase, a robot must be able to properly reproduce a limited number of affordances. However, it is also possible that some of the learned affordances produce incorrect effects; or that some effects available in the environment cannot be reproduced based on the available affordance knowledge.

b) Extended learning phase: the results obtained from the previous task-oriented assessment identify the affordances that have been reproduced, and those that were not reproduced, or produced incorrect effects. Regarding the latter, it is necessary to generate new object-related information to rectify the incorrect affordances, and to extend the affordance knowledge. Thus, a new exploration is performed, consisting in a short constrained effect-driven babbling around the objects related to the incorrect or not generated affordances. The new interactions with the objects generate new action-effect information. This information is discretized, using the previous discretization configuration, and added to the existing dataset. Then, a new structure and parameters of the BN are learned. A similar goal-oriented assessment than in the previous phase is executed to compute a new score and the corresponding performance values. If incorrect or missing affordances are newly identified this phase is again executed. The iterative process stops when all the affordances of the assessment have been reproduced, or the *learning ratio*, the difference of the score among consecutive iterations, goes under a fixed threshold.

An experiment to learn affordances in a simple setup based on A^2L has been developed. In the proposed scenario a

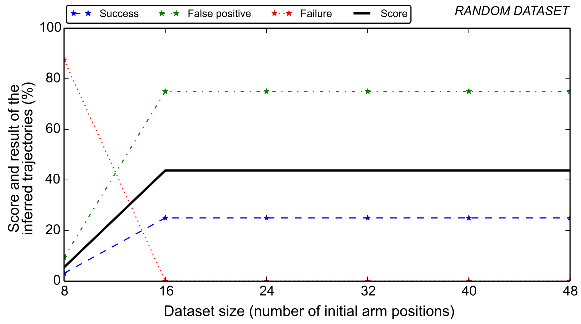

Fig. 1. Score computed using the *Basic learning* applied to different random datasets with different dataset sizes (based on the number of initial positions). Datasets with 16 initial positions or more produce a constant score, i.e., the result of the trajectories inferred by the BN is similar, independently of the dataset size.

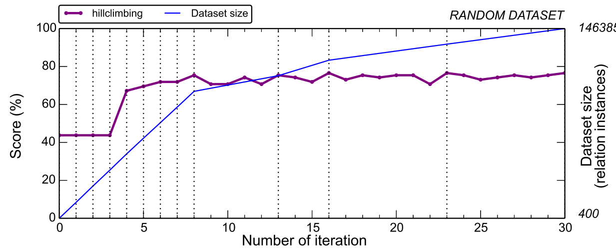

Fig. 2. Score obtained, and dataset size, for each iteration using the *Adaptive learning* for the dataset with 16 initial positions. The dotted vertical lines indicate those iterations extending the dataset, with similar or better score than the previous iteration.

simulated robotic arm learns different trajectories to interact with a box, located in a fixed position, thereby producing different effects on it. Afterwards, the arm is expected to execute a trajectory interacting with the box to obtain a specific effect. In this context a trajectory is a set of moves of the end-effector of the arm; an action relates to a move in a possible direction with a fix distance; and effect represents a new position of the box. Each move is inferred by a BN based on the *hill climbing* learning method, possibly approaching the end-effector to the box or pushing it to a new position. Figures 1 and 2 show how A^2L increases the affordance knowledge of the experiment.

References

[1] J. Pearl, "Causality," *New York: Cambridge*, 2000.

[2] E. Sahin, M. Cakmak, M. R. Dogar, E. Ugur, and G. Ucoluk, "To Afford or Not to Afford: A New Formalization of Affordances Toward Affordance-Based Robot Control," *Adaptive Behavior*, 2007.

[3] E. Ugur, E. Oztop, and E. Sahin, "Goal emulation and planning in perceptual space using learned affordances," *Robotics and Autonomous Systems*, vol. 59, no. 7-8, pp. 580–595, 2011.

[4] A. Cangelosi, M. Schlesinger, and L. B. Smith, *Developmental robotics: From babies to robots.* MIT Press, 2015.

[5] J. Mugan and B. J. Kuipers, "Autonomous Learning of High-Level States and Actions in Continuous Environments," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 1, pp. 70–86, 2012.

[6] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[7] C. Maestre, A. Cully, C. Gonzales, and S. Doncieux, "Bootstrapping interactions with objects from raw sensorimotor data: a Novelty Search based approach," in *ICDL-Epirob*, 2015, p. 6.