Learning Relevant Features to Discover Affordances

Pierre Luce-Vayrac
Institut des Systèmes Intelligents et de Robotique
Sorbonne Université
luce-vayrac@isir.upmc.fr

Raja Chatila
Institut des Systèmes Intelligents et de Robotique
Sorbonne Université
Raja.Chatila@isir.upmc.fr

I. Introduction

The concept of affordances [1] has gained increasing interest in robotics because it enables to ground environment representations by encapsulating proprioception and exteroception in the same framework. The recent work by Zech et al. [7] shows the variety of approaches and contexts affordances have been studied in. However most authors use *predefined* features to describe the environment. We argue that building affordances on predefined features is actually defeating their purpose, by limiting them to a given subspace. Similarly to Nguyen et al. [3] and Mahler et al. [2], we propose here a method for enabling a robot to discover affordances while learning features. but with the notable difference that the exploration and training are done in reality and not in simulation.

II. Approach

We adopt the formalism of Sahin et al. [5], in which an affordance is represented as a triplet *(e, (a,o))*, such that the effect *e* is generated when action *a* is exerted on object *o*.

In most approaches objects are described by a *predefined* set of features. However we affirm the impracticability of predefining a set of features general enough to describe objects in open-ended environments. Instead, we propose an approach to enable the robot to build features relevant to its environments and its capabilities by itself.

In order to decide when to build new features, we propose the concept of *ambiguity*, defined as follows: whenever the agent executes the same action on two apparently similar objects (regarding the current used features), but does not observe the same effect, it has to assume that it does not possess the relevant features to distinguish those objects in regard of this action, hence it needs to learn these features.

In order to learn these new descriptors, we use convolutional neural networks (CNN), for their ability to detect regularities in high dimensional spaces (in our case 2D images) and extract features. Furthermore CNNs enable to transfer learning between networks. Hence we can pretrain the network on another classification task, and then quickly train it on the real situation. In this work, we test two different networks, a *VGG16* [6] pretrained on imagenet, and an eight convolutional layers network (*8conv*) pretrained on a custom smaller dataset (≈ 14000 images from 22 classes).

The affordance is learned by training a multiple layer perceptron (MLP) to predict a discrete effect given discrete objects descriptors values. Figure 1 shows the global architecture of the method, while figure 2 illustrate the reduction of the ambiguity by the construction of a new feature.

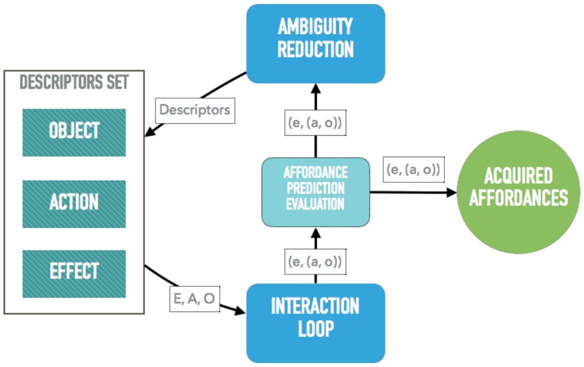


Fig. 1: Global workflow.

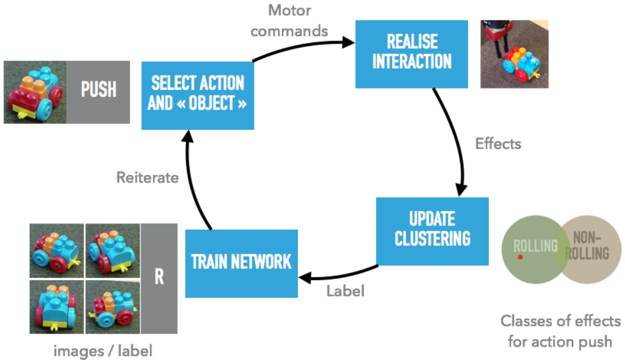


Fig. 2: Ambiguity reduction workflow.

III. Experiments

In order to demonstrate our method we designed two experiments, in which the system would have to build new features to be able to properly learn an affordance.

Both setups consists in an interaction loop where the robot uses a predefined action to explore a set of objects. The actions and object sets are constructed so that different effects will be produced, therefore requiring the robot to classify objects in regard of those effects. We purposefully give a limited initial feature set to describe objects (color and size), to force the agent to build new ones.

Objects are perceived using a kinect v2 RGB-D camera, using method shown in 3.

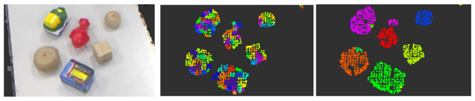
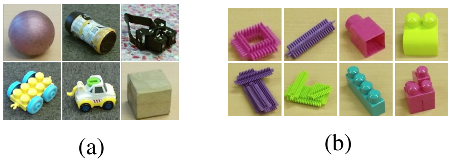


Fig. 3: From left to right, raw point cloud, supervoxels oversegmentation [4], clusters of locally convex supervoxels.

The first setup consists of a set of 37 various objects 4a, some possessing wheels or of spherical shapes (*rolling*), some others not (*non-rolling*). The action is *poke*. The second setup consists of a set of 44 objects of various sizes, textures, colors, shapes and either textured or non-textured 4b. We arbitrarily fix the non-textured ones (*non-movable*) while textured ones are (*movable*). The action is *push*.



(a) (b)

Fig. 4: Examples from objects sets, **(a)** rollable / non-rollable, **(b)** movable / non-movable.

IV. Results

Ambiguity detection is implemented as follows: is considered ambiguous a state in which the training remains underperforming (less than 20% better than the random policy) during 5 training steps. Training starts with only the color and size features, fails to learn the affordance after 5 steps, then proceed to construct a new one by instantiating a CNN. The training of the CNN is done using the method described in 2. The training dataset consists of ten objects at first epoch, then one object is added at each following step. Performance is evaluated on the remaining (unseen) objects. Experiments are done ten times, randomizing selection of objects. Averaged, maximum and minimal performance per epoch are presented in figure 5b and 5a.

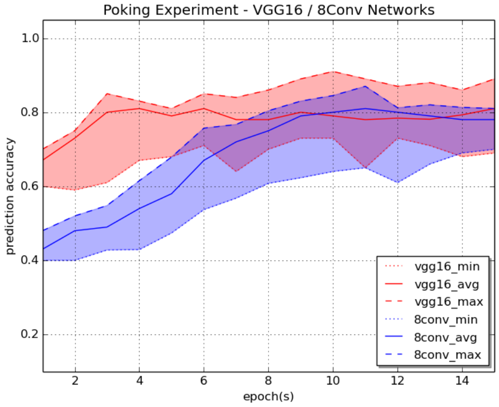
The *VGG16* network, being pretrained on a larger dataset, globally outperforms the *8conv*. However the 8conv is still able to reach a very close performance, especially on the ”push” task where the ”texture” feature may not preexist in the pretrained VGG16. This Implies that it is possible to bootstrap our method without requiring a huge pretraining dataset, and that the features can indeed be learned on the real robotic task.

Acknowledgments

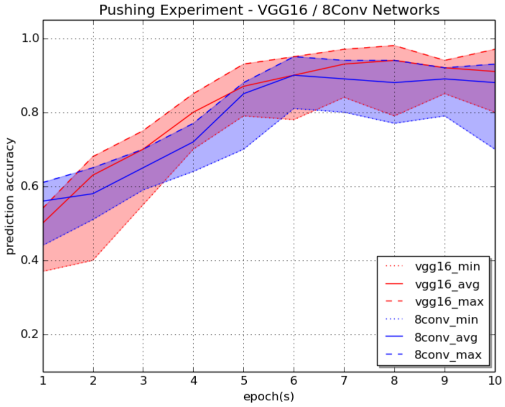
This work has been funded by a DGA scholarship (ER), RoboErgoSum project (ANR-12-CORD-0030), SMART Labex (ANR-11-LABX-65), and EU H2020 Dream Project (no. 640891).

References

[1] James J Gibson. Perceiving, acting, and knowing: Toward an ecological psychology. *The Theory of Affordances*, pages 67–82, 1977.



(a) Poking experiment, prediction accuracy over 15 training steps.



(b) Pushing experiment, prediction accuracy over 10 training steps.

[2] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[3] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Detecting object affordances with convolutional neural networks. In *IROS 2016 IEEE/RSJ*, pages 2765–2770. IEEE, 2016.

[4] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation-supervoxels for point clouds. In *Proceedings of the IEEE Conference CVPR*, pages 2027–2034, 2013.

[5] Erol Şahin, Maya Çakmak, Mehmet R Doğar, Emre Uğur, and Göktürk Üçoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Philipp Zech, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater. Computational models of affordance in robotics: a taxonomy and systematic classification. *Adaptive Behavior*, 2017.