Generalizing Manipulations using Vision Kernels

Oliver Kroemer[1,3], Emre Ugur[2,3], Erhan Oztop[2,3,4], Jan Peters[1,5]

I. Introduction

In order to perform complex manipulation tasks, a robot must know which actions it can perform with the available objects. In unstructured environments, potential manipulations afforded by objects will not be pre-specified, and must instead be learned. Rather than determining each novel object's affordances from scratch, the robot can learn more efficiently by generalizing manipulations from similar known objects.

Actions can be generalized to new objects by learning *direct* mappings from the object's visual features to actions [1]. This approach differentiates itself from indirect methods by not requiring intermediate representations, such as object classes [2]. A robot can autonomously learn the afforded actions of an object by applying the action to the object and observing the resulting effects [3], [4]. If the desired effect is achieved, then the object can be labeled as affording this action. Thus, this affordance learning task can be treated as a binary classification problem for a given action.

Our approach is based on two key insights: 1) The perception of objects and the interactions between objects are based largely on the objects' surface geometries [1], and 2) the affordances of objects are often related to only subparts of objects and not the whole object [5]. Therefore, we propose generalizing actions to new objects by finding subparts of objects that have similar shapes and are, therefore, more likely to have the same affordances.

The subparts of objects are represented in a nonparametric manner, which is based directly on the observed point clouds of the subparts. Thus, the robot does not rely on task-specific visual features, and can discriminate between any subparts that are not visually identical. Using this nonparametric representation, we also define a kernel function for computing the similarity between different subparts. Hence, we can use kernel learning methods [6], such as kernel logistic regression, in order to learn which subparts afford a given action.

The proposed method was successfully tested on a real robot, as shown in Fig. 1. Starting with a single human demonstration of the task, the robot was able to learn to generalize this action to novel objects of different shapes and sizes.

II. Kernel-based Direct Action Perception

As the robot should manipulate a wide range of objects, we require a flexible subpart representation. The subparts of objects are described using three components: the object's point cloud, a weighting function, and the subpart frame. These components are illustrated in Fig. 2 for the handle of a watering can. The subpart frame is similar to a tool center point, and defines the local coordinate system of the subpart. In Fig. 2, the subpart frame is indicated by the black arrows. The point cloud and and the weighting function are both specified in this coordinate system. The weighting function defines which regions of the point cloud are relevant to this subpart. In Fig. 2, the weighting of the points is shown by their color. Points close to the middle of the handle are therefore more important for defining the shape of the handle than points on the spout. In the experiments, we used a Gaussian weighting function centered on the subpart frame.

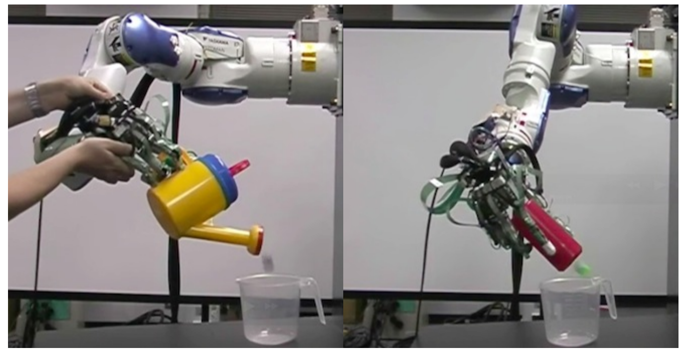

Figure 1. The left image shows the robot being taught a pouring action using kinesthetic teach-in. The robot subsequently learns to generalize the pouring action from the watering can to the cup, as shown in the right image.

Using these three components, we can define a *surface function* for representing the shape of the observed subpart. The function is defined as a mixture of Gaussians centered on the points in the point cloud, and weighted according to the weighting function. The resulting surface function returns a high value when evaluated at a location close to the surface of the subpart, and a value closer to zero when evaluated further away from the surface.

In order to generalize actions to different objects, we must be able to compute the similarity between their shapes. We therefore also define a kernel function for computing the similarity between subparts. The kernel function is defined as the normalized inner product between the surface functions. The kernel function returns a value close to one if both subparts have surfaces in the same locations relative to the subpart frame.

In order to link the actions to the subparts, we use the subpart frame to define the tool center point for the action. Thus, the action defines the trajectory of the subpart within the task space. Manipulation actions often generalize well to new

[1] Technische Universitaet Darmstadt, IAS Lab, Germany
[2] NICT, Advanced ICT Research Institute, Kyoto, Japan
[3] ATR, Cognitive Mechanisms Laboratories, Kyoto, Japan
[4] Ozyegin University, Istanbul, Turkey
[5] Max Planck Institute for Intelligent Systems, Germany

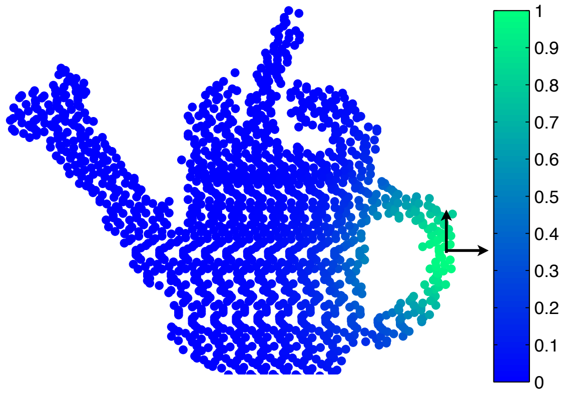

Figure 2. The handle subpart of a watering can. The dots indicate the points of the point cloud. The color indicates the weighting function, which defines how relevant the individual points are to the subpart. The black arrows indicate the coordinate system of the subpart.

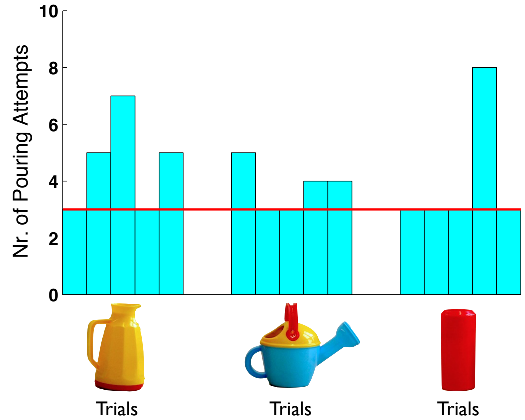

Figure 3. The height of each bar indicates the number of attempts the robot required to achieve three successful pours. Hence, the trials underneath the red line indicate successful pours, and attempts above the line indicate failed attempts. The experiment was repeated five times for each object.

situations when defined in this manner [7], [8]. We represent the actions using dynamical systems motor primitives [9], [10], which can be easily adapted according to the context of the task. In particular, the start and goal states of the movement can be adapted to the selected subpart frame.

Not every subpart will, however, result in a successful manipulation. The robot must therefore also learn which subparts afford the action. Given the vision-based kernel function, we can use kernel logistic regression to classify subparts according to whether they afford the action. Kernel logistic regression first computes the probability that the subpart affords the action, which is then thresholded to obtain the label. Computing this probability is useful for decision making, as it allows the robot to select the subpart that it believes is the most likely to afford the action.

III. Experiment

Our proposed method was tested on both a grasping and a pouring experiment. For safety reasons, the robot had to learn to pour a ball, rather than a liquid, into a container. In each experiment, the robot was given a single demonstration of the task using the large watering can shown in Fig. 1. The robot was then given novel objects, of different shapes and sizes, and had to learn to generalize the action to novel objects. The algorithm was evaluated on how many attempts it required to successfully perform the task with the novel objects.

In the pouring experiment, the robot kept attempting the task until it had successfully poured the ball into the container three times. In each attempt, the robot applied the action to the subpart that was most likely to afford the action, as computed by kernel logistic regression. After each attempt, the robot added the newly acquired information to the training set and retrained the kernel logistic regression for the next attempt. The experiment was repeated fives times on three different objects. The robot's learning was reset between each experiment and each object.

The results of the pouring experiment are shown in Fig. 3. The horizontal red line indicates the three successful trials required to complete the task. Hence, any bar that does not cross this line indicates that the robot immediately generalized the action to the novel object. Overall, the robot only failed to perform the task 17 times, compared to the 45 successful attempts. By learning from its mistakes, the robot was able to successfully generalize the actions to the novel objects in every experiment.

Acknowledgments

The project receives funding from the European Community's Seventh Framework Programme under grant agreement no. ICT- 248273 GeRT, as well as from the JSPS Summer Program 2011.

References

[1] J. J. Gibson, *The Ecological Approach To Visual Perception*. Lawrence Erlbaum Associates, new edition ed., September 1986.

[2] T. Hermans, J. M. Rehg, and A. Bobick, "Affordance prediction via learned object attributes," in *International Conference on Robotics and Automation: Workshop on Semantic Perception, Mapping, and Exploration*, 2011.

[3] S. Griffith, J. Sinapov, M. Miller, and A. Stoytchev, "Toward interactive learning of object categories by a robot: A case study with container and non-container objects," in *Proc. IEEE 8th International Conference on Development and Learning*, (Washington, DC, USA), pp. 1–6, IEEE Computer Society, 2009.

[4] 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.

[5] A. H. Fagg and M. A. Arbib, "Modeling parietal-premotor interactions in primate control of grasping," *Neural Netw.*, vol. 11, no. 7-8, pp. 1277–1303, 1998.

[6] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 1st ed., Dec. 2001.

[7] M. Muehlig, M. Gienger, and J. J. Steil, "Interactive imitation learning of object movement skills," *Autonomous Robots*, vol. 32, pp. 97–114, 2012.

[8] C. C. Kemp, "Robot manipulation of human tools: Autonomous detection and control of task relevant features," in *International Conference on Development and Learning*, 2006.

[9] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *International Symposium on Robotics Research*, Springer, 2004.

[10] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems*, pp. 1547–1554, Cambridge, MA: MIT press, 2003.