A Kernel-based Approach to Direct Action Perception

O. Kroemer[1,3], E. Ugur[2,3], E. Oztop[2,3,4], J. Peters[1,5]

***Abstract*—The direct perception of actions allows a robot to predict the afforded actions of observed objects. In this paper, we present a non-parametric approach to representing the affordance-bearing subparts of objects. This representation forms the basis of a kernel function for computing the similarity between different subparts. Using this kernel function, together with motor primitive actions, the robot can learn the required mappings to perform direct action perception. The proposed approach was successfully implemented on a real robot, which could then quickly learn to generalize grasping and pouring actions to novel objects.**

I. Introduction

In order to plan complex manipulation tasks, a robot must know which actions it can perform with the available objects. In unstructured environments, such as in homes or service industry settings, the potential manipulations of objects will not be pre-specified. Hence, the robot must autonomously determine the possible actions, and adapt these actions according to the specific object being manipulated.

Physically interacting with objects helps an agent to learn object affordances [1], which can then be predicted by learning direct mappings from the object's visual features to specific actions. This approach is known as direct perception of actions, and differentiates itself from indirect methods by not requiring intermediate representations, such as object classes [2], [3]. Direct action perception is a fundamental concept in J. J. Gibson's the theory of affordances [2], which proposes that agents regard objects in their environment according to the actions that these objects allow, or "afford", the agent to perform.

The three main components of an affordance are: 1) the perception of an object entity, 2) the action behavior performed by the agent, and 3) the resulting effect of the action on the object [4], [5]. In this manner, the affordances predicted from visual features can be verified by performing the action and observing the resulting effects. The experience gained from such physical interactions can subsequently be used to predict affordances more accurately in the future. Other physical properties, such as friction and weight distribution, also factor into whether an object affords a particular action. However, this additional data is usually only acquired once the robot has begun manipulating the object. The focus of this paper is on predicting affordances from only vision data.

[1] TU 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

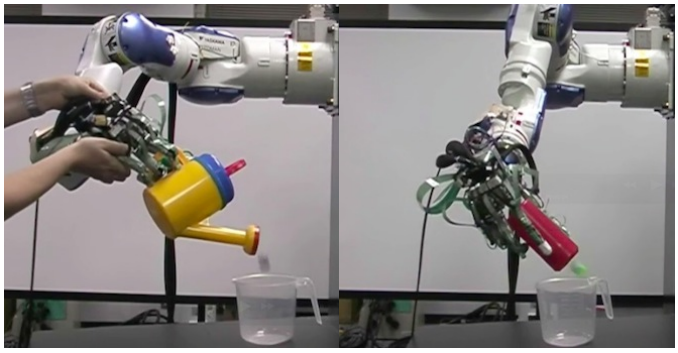

Figure 1. The image on the left shows a human demonstrating a pouring action to the robot using a watering can. The image on the right shows how the robot has learned to generalize the action to a cup using the kernelized direct action perception framework.

In this paper, we propose an example-based approach for robots to learn direct mappings from object point clouds to motor primitive actions. The proposed 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 [2], and 2) the affordances of objects are often related to only subparts of objects and not the whole object [6]. Given these two insights, we propose that the robot should generalize between objects by searching for subparts with similar geometries to those that have previously afforded an action. For example, wedge-shaped subparts can be used for cutting, bowl-shaped subparts can be used for holding fluids, and handle-shaped subparts can be used for grasping.

One of the main challenges of the direct action perception approach is finding a set of suitable visual features for representing objects. If the features do not differentiate between objects that afford an action and those that do not, then it is impossible for the robot to learn the affordance. However, using many features increases the dimensionality of the learning problem and, hence, requires more samples to learn. We propose a non-parametric representation of objects, which is based directly on the point clouds perceived by the robot. In this manner, the robot does not rely on hand-designed features and can learn to discriminate between any objects that are not visually identical. This representation forms the basis of a kernel function, which allows the robot to compute the similarity between subparts. Given this kernel function, kernel-based machine learning methods [7] can be used to learn the shapes of affordance-bearing subparts. The actions are represented using motor primitives, which are flexible and straightforward to adapt to different situations [8], [9].

The kernel function forms the basis for predicting, from previous experiences, the probability that applying the motor primitive to a particular subpart will result in the desired effect. This probabilistic prediction is based on kernel logistic regression, and can be initialized with a single demonstration of a successful action. After initialization, the robot can autonomously learn to predict suitable actions of new objects through interactions with objects in its environment. The proposed approach is called *kernelized direct action perception* (K-DAP). The K-DAP method was implemented on a real robot, as shown in Fig. 1, which was then able to quickly generalize grasping and pouring actions from single demonstrations to various novel objects.

II. Kernelized Direct Action Perception

In order to accurately learn direct action perception, the robot requires flexible representations of both the observed objects and its own actions. Suitable representations for objects and actions are presented in Section II-A and II-B respectively. In Section II-C, we explain how the robot can learn to predict the success probability of applying an action to a specific subpart of an object. We discuss how the proposed approach relates to previous work on affordance learning in Section II-D.

A. Non-parametric Representations of Surface Structures

In this section, we present a non-parametric representation of surface geometries, as well as how the similarity between two subparts of objects can be computed. The proposed representation is based directly on the point clouds acquired from 3D vision systems, such as dense stereo, time-of-flight cameras, and LADAR.

A subpart S is defined by the tuple (O, w, P), where O is the *subpart frame*, $w(\boldsymbol{x})$ is the *weighting function*, and P is a 3D *point cloud* describing the surface of the object. The subpart frame O is a coordinate system defined relative to the real object, which specifies the location and orientation of the subpart in the *task frame*. The point cloud P can then be defined as a set of n points at positions $\boldsymbol{p}_i \in \mathbb{R}^3; i \in \{1, ..., n\}$ relative to the subpart frame O. The weighting function $w(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^3$ is also defined relative to O, gives weights to those regions of the point cloud that pertain to this particular subpart. Given the point cloud P of a whole object, the weighting function defines the points that are relevant for describing the subpart. For the experiments in Section III, the weighting function was defined as an isotropic Gaussian function centered on the subpart frame O.

The surface distribution of subpart S can, thus, be represented by the surface function $f(\boldsymbol{x})$, which has a high value when $\boldsymbol{x}$ is close to the subpart's surface, and a lower value when it is further away. This function can be represented non-parametrically by centering a weighted Gaussian distribution on each of the points $\boldsymbol{p}_i$ from the point cloud. Hence, the surface distribution of the subpart is represented as

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} w(\boldsymbol{p}_i) \exp(-h^{-2} \left\| \boldsymbol{x} - \boldsymbol{p}_i \right\|^2).$$

where $h \in \mathbb{R}$ is a length scale parameter, which can be automatically set using methods such as leave-one-out cross-validation [10]. This non-parametric representation preserves the original vision data and is sufficiently flexible to model any perceivable differences in shape.

In the K-DAP framework, the robot learns to predict the potential actions of an object based on how similar its subparts are to previous subparts. The robot, therefore, requires a similarity measure for comparing the surface distributions of different subparts. The similarity measure, for comparing subpart $S_\alpha = (O_\alpha, w_\alpha, P_\alpha)$ and subpart $S_\beta = (O_\beta, w_\beta, P_\beta)$, is given by the non-parametric surface kernel (NSK):

$$k\left(S_\alpha, S_\beta\right) = \frac{\int_{\mathbb{R}^3} f_\alpha(\boldsymbol{x}) f_\beta(\boldsymbol{x}) d\boldsymbol{x}}{\sqrt{\int_{\mathbb{R}^3} f_\alpha(\boldsymbol{x}) f_\alpha(\boldsymbol{x}) d\boldsymbol{x}} \sqrt{f_\beta(\boldsymbol{x}) f_\beta(\boldsymbol{x}) d\boldsymbol{x}}}.$$

This kernel represents the normalized inner product of the two surface distribution functions, and is closely related to probability product kernels [11]. The value of the kernel has a range from 0 to 1 and the maximal value of 1 is obtained iff the normalized surface distributions of the two objects are identical; i.e. the shapes of the subparts are perceieved to be the same. Higher kernel values are achieved when there is more correlation between the surface distribution functions. When comparing two subparts S_α and S_β, both subparts usually use the same weighting function; i.e., $w_\alpha(\boldsymbol{x}) = w_\beta(\boldsymbol{x}) \forall \boldsymbol{x} \in \mathbb{R}^3$. Therefore, we exclude the subscript for the weighting function in the remainder of this paper. This kernel function allows the robot to use kernel methods [7] from machine learning to predict from the subpart's shape whether it affords an action.

As the cloud points are represented by Gaussians, the kernel value is straightforward to compute analytically. The terms in the denominator need to only be computed once for each subpart. Computing the numerator requires integrating over $n_\alpha n_\beta$ Gaussians. This computation can be performed efficiently by first pruning out points with low weights $w(\boldsymbol{p}_i) \approx 0$, and only considering pairs of points that are near each other according to the length scale parameter h.

B. Linking Visual Features to Action Parameters

Rather than only predicting whether a specific action is applicable to an object, the robot must also adapt its actions according to the subpart it is manipulating. Therefore, rather than using fixed actions, the robot should use *motor primitives*. A motor primitive represents a continuous range of similar actions that an agent can perform [12]. The specific execution of a motor primitive is defined by a small set of meta-parameters, which are selected according to the context of the action.

For the robot, we propose using dynamic systems motor primitives (DMPs) [8], [9], which have been successfully used to allow robots to perform a wide range of motor skills [13], [14], [15]. A DMP can be learned from a single, or multiple [15], demonstrations of an action and defined such that the only open meta-parameters are the movement's initial state $\boldsymbol{y}_s$ and goal state $\boldsymbol{y}_g$ [8]. In the K-DAP framework, the motor primitives are used to define the trajectory of the subpart frames, e.g. O_γ, relative to the task's reference frame. As a

Algorithm 1 K-DAP Learning Procedure

INITIALIZATION FROM A SINGLE DEMONSTRATION:

1. Observe object to obtain point cloud P_0
2. Define example subpart $S_0 = (P_0, w, O_0)$:
   O_0 defines location of subpart in task frame
   $w(\boldsymbol{x})$ defines region of P_0 relevant to subpart
3. **while** human is demonstrating action:
   Record the trajectory τ of O_0 within the task frame
4. Set start state and goal states:
   $y_s = \tau(t = 0)$ and $y_g = \tau(t = end)$
5. Learn DMP action $A(y_s, y_g)$ from τ
6. Set result of demonstration as successful $E_0 = 1$
7. Compute maximum-likelihood estimate of $P(E|S, A)$

FOR EACH NEW SUBPART S_m:

1. Observe new object to obtain point cloud P_m
2. Search P_m for subpart frame O_m:
   $O_m = \arg\max_O P(E_m = 1|(P_m, w, O), A)$
3. Set start state $y_s =$ current pose of O_m in task space
4. Robot executes DMP action $A(y_s, y_g)$
5. **if** action was successful $E_m = 1$, **else** $E_m = -1$
6. Compute maximum-likelihood estimate of $P(E|S, A)$:
   $\boldsymbol{v} = \arg\max_{\tilde{\boldsymbol{v}}} \sum_{i=0}^{m-1} -\ln\left(1 + \exp(-E_i \tilde{\boldsymbol{v}}^T \boldsymbol{k}(S_i))\right)$

OUTPUT:

Affordance-bearing subpart predictor:
$P(E = 1|S, A) = \left(1 + \exp(-\boldsymbol{v}^T \boldsymbol{k}(S))\right)^{-1}$

result, the robot always moves the selected subparts of objects in a similar manner when performing a specific task.

From a developmental viewpoint, 7-10 months old infants can acquire skills more quickly when a caregiver provides a single demonstration of the task [16], as well as draw the infant's attention to task-relevant features and sub-goals [17]. Similarly, the robot's actions may be learned from a human demonstration of the task. This demonstration provides the agent with an example of an affordance-bearing subpart S_0, as well as the trajectory τ of this subpart frame O_0 during the task. This trajectory can then be used to learn the movement parameters of the DMP. The initial and final states are given by the start of the trajectory $\boldsymbol{y}_s = \tau(t = 0)$ and its termination state $\boldsymbol{y}_g = \tau(t = \text{end})$. We denote the learned DMP action, and its hyper-parameters, as $A(y_s, y_g)$.

Given the point cloud P_γ of a novel object to manipulate, the action selection process involves searching the point cloud for a new subpart frame O_γ on which to execute the learned DMP. The robot selects the subpart frame corresponding to the subpart with the highest probability of affording the action. The probability of a subpart being affordance-bearing is based on the robot's previous interactions with similar objects. In Section II-C, we will discuss in detail how the robot computes these probabilities.

Once the new subpart frame O_γ has been chosen, the initial state of the motor primitive is defined by the initial state of the subpart frame $\boldsymbol{y}_s = O_\gamma$. Similarly, the goal state $\boldsymbol{y}_g$, defined relative to the task frame, is assumed to be the same as the one used in the demonstration. For tool usage and similar tasks, the goal state can be defined relative to another object's subpart S_ϵ by defining the task frame as O_ϵ. The selection of the new subpart frame O_γ thus sets the necessary meta-parameters of the DMP and, hence, defines a specific action that the robot can execute $A(y_s, y_g)$.

When selecting a subpart frame O_γ, the robot is effectively defining a new subpart $S_\gamma = (P_\gamma, w, O_\gamma)$. In this manner, each choice of action becomes linked to a specific set of visual features. These visual features can then be used as the basis for predicting whether the action is being applied to an affordance-bearing subpart and will therefore be successful.

C. Learning to Predict Affordance-bearing Subparts

The DMP behaviors in Section II-B are defined for all possible subparts. However, the action will only be successful, and result in the desired manipulation, if the sulected subpart affords the action. The final part of the K-DAP framework is therefore to predict whether applying the motor primitive $A(y_s, y_g)$ to a subpart will result in the desired manipulation. We will assume that the effects corresponding to successful $E = 1$ and unsuccessful $E = -1$ action executions are predefined. The effect classes can also be learned in an unsupervised manner [1], [18], but this is beyond the scope of this paper.

Rather than using a classifier to directly predict the outcome class E, we propose using kernel logistic regression (KLR) to learn the probability of a subpart affording a specific motor primitive. An action A on subpart S is then predicted to be successful $E = 1$ if $p(E = 1|S, A) > p(E = -1|S, A)$. A continuous probabilistic representation is useful for selecting a suitable action, as it allows the robot to differentiate between multiple subparts that are labeled as successful. The KLR approach is based on the maximum entropy principle and, hence, will assign probabilities close to 50% to subparts that are dissimilar to all previous subparts. Using KLR, the predicted probability of successfully applying an action A to a subpart S is given by

$$p(E = 1|S, A) = \left(1 + \exp(-\boldsymbol{v}^T \boldsymbol{k}(S))\right)^{-1},$$

where the i^{th} vector element of $\boldsymbol{k}(S)$ is given by $[\boldsymbol{k}]_i = k(S, S_i)$, the $S_{0\ldots m-1}$ are the m previously encountered subparts, and $\boldsymbol{v} \in \mathbb{R}^m$ is a learned weight vector. One KLR is learned for each affordance. The logistic sigmoid function ensures that the probability is valid and $p(E = 1|S, A) + p(E = -1|S, A) = 1.0$.

The weight vector $\boldsymbol{v}$ is computed by finding the maximum likelihood solution

$$\boldsymbol{v} = \arg\max_{\tilde{\boldsymbol{v}}} \left[\sum_{i=0}^{m-1} \ln\left(\left(1 + \exp(-E_i \tilde{\boldsymbol{v}}^T \boldsymbol{k}(S_i))\right)^{-1}\right)\right],$$

where $E_i \in \{-1, 1\}$ indicates whether subpart S_i had afforded the action A. This optimization problem is convex and, therefore, the global maximum can be found using the Newton-Raphson method. In practice, this optimization is usually computed with a small amount of regularization, which penalizes large values for $\boldsymbol{v}$. Regularization avoids over-fitting

Figure 2. The picture shows the four objects used in the pouring task experiment. The robot was initially shown how to pour with the large watering can on the left. The robot then had to autonomously learn to generalize this demonstrated action to the three objects on the right: a plastic cup, a different watering can, and a small jug.

$\boldsymbol{v}$ to the previous examples and results in better generalization to new subparts. Once the weight vector $\boldsymbol{v}$ has been learned, the robot can predict the success probability of applying the motor primitive to novel objects.

D. Related Works

The direct action perception and affordance learning frameworks have been receiving an increasing amount of interest from the robotics community [4]. However, previous work in this area has largely focused on learning the affordances related to entire objects, such as lifting and rolling [19], [3]. Learning affordances at the subpart level has usually only been studied in the context of learning visual cues for specific actions, such as grasping [20], [21], [22]. These approaches use predefined parametric features to represent objects and their subparts. Instead, the K-DAP approach uses a more flexible non-parametric representation that is based directly on the robot's observations. Our previous work on learning affordances through self-exploration [1] and parental scaffolding [23] has also usually used a fixed set of pre-programmed behaviors. The proposed approach uses DMPs in order to learn actions from a single human demonstration, and adapt these actions to different subparts.

The direct linking of point clouds and actions, as used in the K-DAP framework, was inspired by the work of J.J. Gibson on optical flow for affordances [2]: When the DMP trajectory of a subpart frame O is combined with the relative position of the point cloud P, the robot actually defines a 3D trajectory for each point in the point cloud. Therefore, if two subpart frames O_α and O_β follow the same trajectory, and two points are located at the same positions relative to their subpart frames $\boldsymbol{p}_{\alpha i} = \boldsymbol{p}_{\beta j}$, then these two points will also have the same trajectory. If many points match between the subparts, then these points would induce the same optical flow for the two objects. Therefore, a high NSK value between two subparts $k(S_\gamma, S_\beta) \approx 1$ effectively predicts that the same action on the two subparts will result in a similar optical flow.

III. Experiments

The proposed method was realized on the robot shown in Fig. 1. The robot consists of a 7 degrees-of-freedom Motoman robot, a five-fingered Gifu robot hand, and a Swiss Ranger time-of-flight camera for perceiving the robot's environment.

The robot was given the tasks of generalizing grasping and pouring actions from one object to various other objects that afforded these actions. A key goal of these experiments is to test whether the K-DAP framework can be initialized with a single demonstration. The experiments show that the robot can autonomously learn to generalize the demonstrated actions to new objects.

The general framework of the experiments is explained in Section III-A. The grasping experiment is detailed in Section III-C, and the pouring experiment is explained further in Section III-D.

A. Learning Initial Affordances from Demonstration

Inspired by infant development [23], a parental scaffolding approach is used to teach the robot new motor-skills. First, a human provides an initial demonstration of how the task is performed. Afterwards, the robot is allowed to learn by interacting with similar objects, using the initial demonstration as an initialization point. The grasping and pouring actions were demonstrated to the robot using kinesthetic teach-in, as shown in the left image of Fig. 1. By guiding the robot through the required movements, the demonstrator could transfer their knowledge of the motor skill, to the robot, in an intuitive manner.

The demonstrations for both the grasping and the pouring actions were performed using the large watering can shown in Fig. 2. An important part of the demonstration is defining the relevant subpart. For many tools, the subpart frame can be defined at the main point of contact between the tool and the object that the tool is manipulating. The subpart frames were therefore positioned on the points $\boldsymbol{p}_i$ closest to the other object being manipulated. For grasping, the subpart was positioned on the handle, closest to the hand frame, and aligned with the approach direction of the hand. For pouring, the subpart frame was positioned on the lip of the spout, closest to the container being poured into, and aligned with the tipping axis of the pouring motion. Given the demonstration of the task and the subpart frame, a suitable DMP could be learned.

The point clouds of the objects were acquired from a single perspective of the objects using the robot's time-of-flight camera. The object's point cloud was automatically segmented from the background and the robot's arm. The points were weighted according to an isotropic 3D Gaussian weighting function $w(\boldsymbol{x}) = \exp(-\hat{h}^{-2}\boldsymbol{x}^2)$. The width parameter $\hat{h}$ was set to match the size of the subpart; i.e., the size of the handle for grasping, and the head of the spout for pouring.

B. Searching for Affordance-bearing Subparts

After initializing the system with a human demonstration, the robot had to learn to predict affordance-bearing subparts through interactions with novel objects. For each attempt at the task, the robot evaluated the subpart that it found to be the most likely to succeed $O_m = \arg\max_O P(E_m = 1|(P_m, w_m, O), A)$. The robot started each trial by acquiring a new point cloud P_m for the current object. The objects were shifted between attempts, but always positioned such that an affordance-bearing subpart was perceivable and the action could be executed.

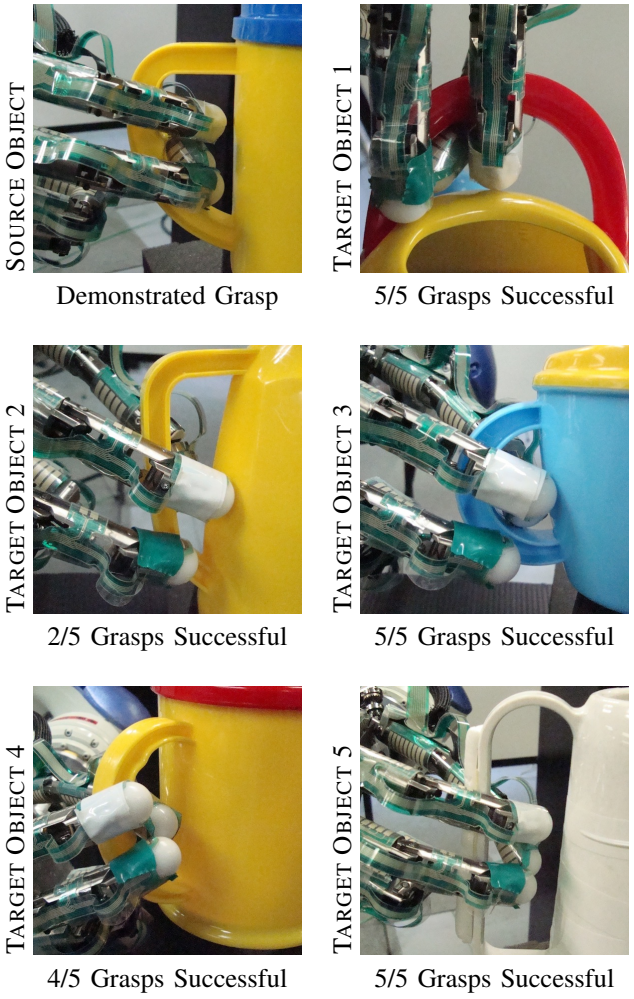

Figure 3. The top left image shows the grasp demonstrated to the robot by a human teacher. Using the K-DAP approach, the robot learned to generalize the demonstrated grasp two five other objects. The other five images show the final grasps learned for these objects. The fraction below each image indicate the number of successful grasps that the robot executed while learning to grasp this object. A 5/5 indicates that the robot could immediately generalize the demonstrated action to these objects

The search for the new subpart frame consisted of two stages. In the first stage, the likelihood of success is evaluated for each point in the object point cloud. The orientations of the subpart frames were set by aligning the principle component directions of the weighted point clouds. The point with the highest success probability was then used to initialize the second stage of the search. This second stage used a stochastic local optimization procedure to find a suitable subpart frame, which could then be evaluated. The entire searching process required on average only two seconds per previous subpart.

After executing the DMP on the subpart, the results of the attempt were evaluated and the learned success probabilities were updated. Although we hand-coded the effect classes for this experiment, the robot could have also discovered them by monitoring and categorizing the created effects autonomously [1], [18]. The updated KLR was then used to determine the subpart for the next attempt. In order to maintain independent experiments, the learning process was reinitialized before the robot began interacting with a new object. In a real world setting, the robot would not reinitialize between objects, and would instead accumulate the knowledge gained from multiple objects. The K-DAP approach for learning affordances from physical interactions is summarized in Alg. 1, including the initialization by a human demonstration.

C. Grasping

In the first experiment, the robot was given the task of generalizing a grasping action to five novel objects. The ability to grasp objects is an important prerequisite to many other manipulation actions. All of the objects had handles, but of varying shapes and sizes. The robot was given only five attempts to grasp each of the test objects, resulting in 25 grasps overall. An attempt was considered a success $E = 1$ if the robot placed its fingers such that it could lift the object afterwards. Otherwise, the attempt was a failure $E = -1$.

The results of the experiment are shown in Fig. 3. A total of 21 of the 25 attempted grasps were successful (84%), and the robot was able to immediately determine a suitable grasp for three of the five objects.

The most difficult object to grasp was the small jug shown in Fig. 2; i.e., Test Object 2 in Fig. 3. The reason for this relatively low score is due to the opening on the top of the container. When viewed from above, the concave sides of the opening result in self-occlusions, and its rim is perceived as a ring floating in space. This ring structure has a similar shape to a handle. After a couple of failed grasps in this region, the robot learned how to correctly grasp the object by the handle.

One possible solution to the problem of self-occlusions is to use full, rather than partial, point clouds of objects. Such point clouds could either be accumulated from multiple view points of the scene, or by predicting the shape of occluded regions [24]. This extension of the K-DAP framework is however beyond the scope of this paper, but will be investigated further in the future.

D. Pouring

The second experiment focused on a pouring task. In order to avoid damage to the robot, the robot learned the pouring action with rubber balls instead of a fluid. The robot had to learn to generalize the pouring action to three different objects: a small jug, a small watering can, and a plastic cup. These three objects are shown in Fig. 2, next to the large watering can used for demonstrating the action. For the pouring experiment, the grasps of the objects were selected such that large parts of the objects' surfaces were visible, including many subparts that did not afford the pouring action. Therefore, the cup was grasped from below rather than the side, as shown in Fig. 1. The robot ran five trials on each object. Each trial consisted of the robot repeatedly attempting the pouring task until it had successfully poured the ball into the plastic container 3 times.

The results of the experiment are shown in Fig. 4. The average number of attempts required to complete the task were 4.6, 3.8, and 4.0 for the small jug, watering can, and plastic cup respectively. Given that each trial contained exactly three successful attempts, the robot failed on average 1.2 attempts per trial while learning.

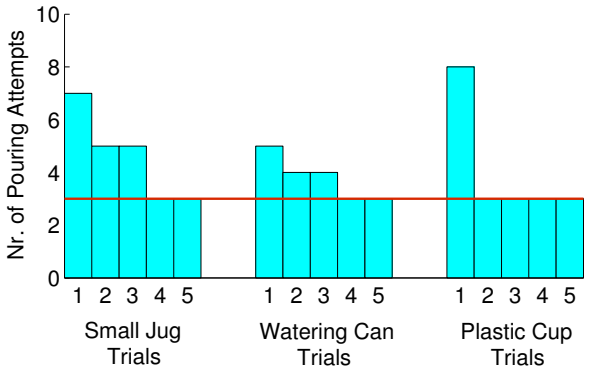

Figure 4. The number of attempts required to successfully pour three times from an object. Each bar indicates one trial, in which the robot is initializes with a single demonstration of grasping a different object. The red line indicates the three successful grasps required to complete the task. The number of grasps above the red line indicates the number of failed grasps that the system used to learn the correct action.

In four of the five experiments with the plastic cup, the robot was able to immediately generalize the pouring action from the large watering can. This is largely due to the fact that the basic shape and rotational-symmetry of the cup tended to result in similar visual features across trials. In the trial that required eight attempts, the first trial found a subpart that resulted in the cup not being tilted enough for the ball to fall out. The robot subsequently tried a few other regions of the cup before learning to use the opening properly. Most of the failures of the watering can corresponded to attempts to pour the ball using the opening on the top. However, this opening did not allow the ball to be poured in a controlled manner and, hence, these trials were regarded as failures.

IV. Conclusion

The direct action perception framework presents an effective approach for a robot to generalize manipulations between different objects. In this paper, we presented a non-parametric approach to representing the surfaces of object subparts. This representation forms the basis of a kernel function, which is used to learn the shapes of affordance-bearing subparts. In order to adapt to different subparts, the robot's actions are defined as motor primitives.

The proposed framework was implemented on a real robot. The robot was initialized with a single demonstration from a human, and subsequently learned through interactions with other objects in its environment. As a result, the robot could then quickly generalize both grasping and pouring actions to new objects.

V. Acknowledgments

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

References

[1] E. Ugur, E. Sahin, and E. Oztop, "Unsupervised learning of object affordances for planning in a mobile manipulation platform," in *International Conference on Robotics and Automation*, pp. 4312–4317, 2011.

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

[3] 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.

[4] 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*, vol. 15, pp. 447–472, December 2007.

[5] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann, "ObjectÐAction Complexes: Grounded abstractions of sensoryÐmotor processes," *Robotics and Autonomous Systems*, vol. 59, pp. 740–757, Oct. 2011.

[6] 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.

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

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

[9] A. Ijspeert, J. Nakanishi, and S. Schaal, "learning attractor landscapes for learning motor primitives," in *advances in neural information processing systems 15*, pp. 1547–1554, cambridge, ma: mit press, 2003.

[10] L. Wasserman, *All of Nonparametric Statistics*. Springer, 2006.

[11] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, December 2004.

[12] T. Flash and B. Hochner, "Motor primitives in vertebrates and invertebrates," *Current Opinion in Neurobiology*, vol. 15, no. 6, pp. 660 – 666, 2005. Motor sytems / Neurobiology of behaviour.

[13] O. Kroemer, R. Detry, J. Piater, and J. Peters, "combining active learning and reactive control for robot grasping," no. 9, pp. 1105–1116, 2010.

[14] J. Kober and J. Peters, "practical algorithms for motor primitive learning in robotics," no. 2, pp. 55–62, 2010.

[15] K. Muelling, J. Kober, and J. Peters, "Learning table tennis with a mixture of motor primitives," in *proceedings of International Conference on Humanoid Robots*, 2010.

[16] D. J. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *Journal of Child Psychiatry and Psychology*, vol. 17, no. 2, pp. 89–100, 1976.

[17] P. Zukow-Goldring and M. A. Arbib, "Affordances, effectivities, and assisted imitation: Caregivers and the directing of attention," *Neurocomputing*, vol. 70, pp. 2181–2193, Aug. 2007.

[18] B. Ridge, D. Skočaj, and A. Leonardis, "Unsupervised learning of basic object affordances from object properties," in *Computer Vision Winter Workshop*, (Eibiswald, Austria), pp. 21–28, February, 4–6 2009.

[19] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, G. Sandini, and G. S, "Learning about objects through action - initial steps towards artificial cognition," in *In Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp. 3140–3145, 2003.

[20] Fritz, Paletta, Breithaupt, and Rome, "Learning predictive features in affordance based robotic perception systems," in *International Conference on Intelligent Robots and Systems*, pp. 3642 –3647, oct. 2006.

[21] L. Montesano and M. Lopes, "Learning grasping affordances from local visual descriptors," in *International Conference on Development and Learning*, pp. 1 –6, june 2009.

[22] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele, "Functional object class detection based on learned affordance cues," in *international conference on Computer vision systems*, ICVS'08, (Berlin, Heidelberg), pp. 435–444, Springer-Verlag, 2008.

[23] E. Ugur, H. Celikkanat, E. Sahin, Y. Nagai, and E. Oztop, "Learning to grasp with parental scaffolding," in *International Conference on Humanoid Robots*, pp. 3140–3145, 2011.

[24] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap - robotic grasping under incomplete observation," in *proceedings of International Conference on Robotics and Automation*, pp. 686–693, 2011.