2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Interactive object classification using sensorimotor contingencies

Virgile Högman, Mårten Björkman, Danica Kragic

Abstract— **Understanding and representing objects and their function is a challenging task. Objects we manipulate in our daily activities can be described and categorized in various ways according to their properties or affordances, depending also on our perception of those. In this work, we are interested in representing the knowledge acquired through interaction with objects, describing these in terms of action-effect relations, i.e. sensorimotor contingencies, rather than static shape or appearance representations. We demonstrate how a robot learns sensorimotor contingencies through pushing using a probabilistic model. We show how functional categories can be discovered and how entropy-based action selection can improve object classification.**

I. Introduction

Psychological studies show that humans commonly use the notion of categories to group objects, actions and other integral parts of their surroundings for an efficient interaction with the environment [1]. In the robotics, computer vision and machine learning communities, object and action categorization has been studied extensively [2], [3], [4], [5], [6]. However, many of the reported approaches consider categorization as a passive observation problem, and formalize it by solely assigning labels to a static representation. The problem boils down to, for example, generating a database of images or image sequences that are representative for a certain class or category of objects and actions.

Our interest is understanding how object categories arise given active interaction with the world and a specific robot embodiment. For example, a robot capable of only performing pushing actions may not have the same ability to build rich representations of objects as the robot capable of grasping and manipulating them. Thus the question we pose is: "*How many different categories of objects can be discovered if a robot is only able to perform pushing actions?*" In addition, we want to measure how effective an action is, e.g. a push in a specific direction, and how we can then define a proper action to classify a new object that a robot is faced with.

The work relates to the studies of O'Regan and Noë [7], who put forward a theory that visual consciousness is the result of an action-based exploration of the world, claiming that *seeing is a way of acting*. This theory further suggests that the outer world can be probed as an external memory, through the sensorimotor contingencies (SMCs), instead of building an internal representation stored in the agent's memory. By doing so, representations redundant to the agent and the limitations of its embodiment are avoided. Thus an agent can develop cognitive behaviors through the mastery of sensorimotor contingencies, representing knowledge by associating sensory outcomes to actions, acquired by experience as illustrated in Fig. 1. According to [7], the agent's sensorimotor contingencies are constitutive for cognitive processes. In this framework, sensorimotor contingencies are defined as law-like relations between movements and associated changes in sensory inputs that are produced by the agent's actions. Seeing is not understood as the processing of an internal visual representation, but the process of being engaged in visual exploratory activity, mediated by knowledge of sensorimotor contingencies. An extended version of the SMC theory, referred to as extended sensorimotor contingencies (eSMCs), has the goal of capturing SMCs at different levels of complexity:

- Modality-related eSMCs capture the specific changes of the sensory signal in a single modality depending on the agent's action.
- Object-related eSMCs concern the effects on the sensory system that are specific for the objects under consideration.
- Intention-related eSMCs consider the long-term correlation structure between complex actions/action sequences and the resulting outcomes or rewards, which the agent learns to predict.

A parallel can be drawn to Object-Action Complexes (OACs) [8]. OACs have been proposed as bricks for cognitive architectures and are formalized to include an object and

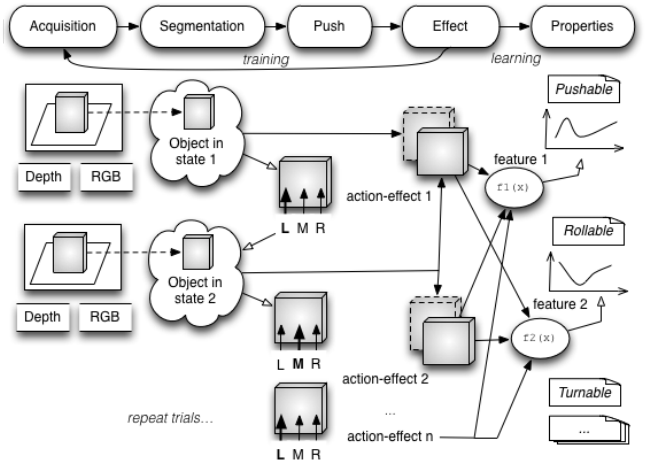

Fig. 1. Learning process: every action applied on the object leads to an effect. By repeating actions several times, the agent gains experience that is used for learning of different object properties.

V. Högman, M. Björkman and D. Kragić are with the Centre for Autonomous Systems, Computer Vision and Active Perception Lab, CSC, KTH Royal Institute of Technology, Stockholm, Sweden. Email: {virgile|celle|dani}@kth.se.

action specific transition function:

$$T : s \rightarrow s'$$

where s is a set of attributes prior to the action and s' is the set afterwards. OACs are organized in different levels of abstraction, where low-level OACs can be loosely mapped to modality-related eSMCs and high-level OACs to intention-related ones.

The Distributed Adaptive Control (DAC) [9] presents a biologically inspired model, with a three-layered architecture. A reactive layer provides a pre-wired set of reflexive behaviors; an adaptive layer allows adaptive classification of sensory events; and a contextual layer uses long-term and short-term memory to support action sequences. It can be formalized as:

$$f : \{a, s\} \rightarrow a'$$

where $\{a, s\}$ is a sequence of action-outcome pairs and a' is a new action for the robot to pursue. Thus unlike the forward model in OACs, where the function gives predicted sensory data after an action, DAC uses an inverse model [10].

In this paper we study object-related eSMCs, sensorimotor contingencies specific to particular objects or object categories, and study their discovery through pushing. We represent these contingencies as object-specific functions:

$$f : (s, a) \rightarrow s'$$

where s is the sensory data prior to an action a and s' is the sensory data afterwards. Object-related contingencies are tightly connected to the notion of object affordances [11], [12], [13], [14], [15] that are defined by the set of actions that can be applied to an object. An object-related eSMC encodes or predicts an outcome of a certain action performed on an object.

The framework used for this study is shown in Fig. 1. From image data (RGB and depth pixels measurements) an object in the center of the view is segmented. A dominating plane is found for this segment and a pushing action is generated with respect to the plane. Translational and rotational changes due to the push are recorded and learned with a model based on Gaussian Processes (GPs) [16] that represents the eSMCs. Categorization is then formalized as the process of grouping objects based on similarities between different GPs. Our contributions can be summarized as:

- Encoding eSMCs with GPs: we show how object features can be learned from a single pushing behavior, using a probabilistic model explicitly relating actions to their effects, predicted with a measure of confidence.
- Demonstrating action selection: we propose an entropy-based policy to evaluate the proper action, in a continuous space, to classify objects, and show the benefits of this policy in experiments with real objects.
- Categorization: we show how the same model can be exploited to learn categories after grouping objects in a unsupervised manner with a similarity distance measure derived from the GPs.

All these aspects are constitutive of a sensorimotor-based approach where an agent acquires knowledge through interaction with objects, giving rise to object-related eSMCs.

II. Related work

There are several reported works that relate to ours. The work reported in [17], [13] use pushing to learn object affordance of "rollability", relatively to their principal axis. The learned effect on an object is defined as the probability of success to roll or not, assuming the agent already knows how to evaluate this. In our work, we do not make this assumption and preserve the effect as sensory information with translation and rotation features.

Sanchez-Fibla et al. [14] introduce affordance gradients to model how object's rotation and translation change as it is pushed. The shape is learned from IR readings of a small E-puck robot that is circulating around the object. Maye and Engel [18] use a similar setup, but let sensory and action spaces be discretized with earlier observations recorded as eSMCs stored in histograms. Our model instead allows continuous sensory and action spaces; and, to learn object-related eSMCs from a more limited set of pushes using a real robotic arm, we represent these as Gaussian Processes (GPs) from which predictions can be made already after a few exploratory trials. The GP model also allows for uncertainty in outcomes, either from noise in sensory data or due to the nature of the objects.

Similar to the work presented here, Ugur et al. [15] learn action-outcome relations and apply these for functional categorization. Object-related sensory data are represented as 43-dimensional vectors, which include object position, visibility and shape. These feature vectors are used for prediction of functional categories of observed objects, with the predicted effect of an action given by a category prototype learned from clustering. Our work differs in that classification is done not from predictions using the sensory data alone, but as a result from multiple observations of effects from actions applied to objects. Instead of having an explicit representation of object shape, shape is represented implicitly by the effects the shape give raise to.

Sinapov et al. [19] combine proprioception, vision and audio from several behaviors for classification of a large range of objects using discriminative models. Sensorimotor contingencies are simply represented by sets of tuples from training observations. In our context, classification should only be seen as one way of probing the acquired knowledge base represented as eSMCs, rather than the goal of the learning process. We will show in Section III that such models give the robot the ability to make predictions and evaluate the confidence of action outcomes.

With regard to developmental robotics, Stoytchev [20] formulates five principles to be considered in a learning system. Among those, we found the *verification* and *grounding* principles particularly relevant for our work. The verification principle states that an intelligent agent can create and maintain knowledge only to the extent that it can verify it by itself. By observing this, we have tried to avoid any representations

other than those derived from observations of action-effect pairs. The principle of grounding sets the boundaries to the concept of verification, which can be achieved by repeating actions in the same context, to gain confidence about their outcomes. Our learning process follows this concept, GPs being trained by sets of observations, with the variance measuring their related uncertainty.

In previous work [21], we addressed pushing in terms of interactive perception to disambiguate between groups of objects. Segmented objects were pushed multiple times much like the work presented here, but classification was done based on the number of objects observed in a scene, rather than on the functional qualities of individual objects. Furthermore, the previous work did not include learning of categories, but assumed classes to be given.

III. Modelling

We describe a model for action-based object classification in terms of object-related eSMCs. We study how objects of different shape behave after a pushing action has been applied to them. Two low-level processes are assumed to exist: a foveation process that directs the gaze towards an object of interest, keeping it in the center of view, and a perceptual grouping process that segments the object from its surroundings. Thus, a reference frame can be established with the object centroid as its origin.

A. Pushing actions and their effects

We adopt an object-centered representation, where a push is generated with respect to the local object frame. A dominating plane is sought by fitting a plane to the 3D point cloud representing the segmented object region, a procedure that is done with RANSAC [22]. The object orientation is defined as the orientation of this plane. Note that for many objects, such as balls, the extracted dominating plane will not correspond to a real physical plane on the object itself, but be a virtual plane through which the highest number of 3D points belong. Once the object orientation is determined, a local coordinate system is defined as shown in Fig. 2. We consider objects lying on a horizontal plane such as a table, so the push is always performed along the Z-axis. Coordinates are normalized along the X- and Y-axes, such that -1 and +1 correspond to the extent of the object on either side.

A pushing action $a = (a_X, a_Y)$ is defined by the position on the extracted dominating plane measured along the X- and Y-axis. In the experiments, we will keep a_Y fixed to positions along the lower part of the object and apply actions in directions parallel to the surface the object is placed on. Even if additional action parameters could be considered, such as the speed and duration of the push, these are not included in the model under consideration. During the experiments the speed is kept approximately constant, but the duration varies slightly from time to time. An eSMC can be expressed as a function $f : (s, a) \rightarrow s'$, where $s = (s_p, s_o)$ is the sensory data prior to the application of the action a, with s_p being the position of the foveated object and s_o the orientation of the dominating plane. However, since the action is applied with respect to the local reference frame of the object, and not a global one, the function can be simplified as:

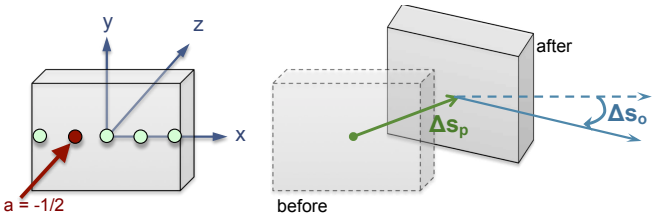

Fig. 2. Effect of a push, described by translation and rotation features, in an object-centered representation. The dots along the X-axis represent the possible actions used in the experiments.

$$f_\Delta : a \rightarrow \Delta s \tag{1}$$

where $\Delta s = (\Delta s_p, \Delta s_o)$ represents the change in position and orientation due to the action a, denoting the translation and rotation features as effects (see Fig. 2). By applying a series of pushing actions on a particular object and observing changes in the sensory data Δs, the function f_Δ is learned.

B. Learning process

Following the principle of grounding [20], an agent acquires knowledge with a certain degree of confidence from repeated observations of action-effect pairs. This can be achieved with a probabilistic representation through statistical learning. We base our model on Gaussian Process regression [16], where the uncertainty takes both the current lack of observations and the noise in outcomes into account. In other terms the variance represents the confidence the robot has over its sensorimotor contingency. This approach also allows to infer on a continuous action space, though the experiments are limited to a discrete case. We decompose f_Δ such that each feature is represented by a GP, defined by a mean and a covariance function. We use a zero mean function and the squared exponential with additive white noise for the covariance. The optimal hyperparameters are found by optimizing the marginal likelihood of the training data as described in [16].

C. Object classification

Once object-related eSMCs have been learned for a given number of classes (object instances or categories), the agent can perform functional classification on new objects. By performing an action and observing its effect, it is possible to determine which class the object is most likely to belong to. Assuming the classes of the training samples to be known, a Bayes classifier is defined as follows:

$$p(c|s,a) = \frac{p(s|c,a)p(c)}{p(s|a)} = \frac{p(s|c,a)\,p(c)}{\sum_{c'} p(s|c',a)\,p(c')} \tag{2}$$

where c is the object class, s is the observed outcome after executing the action a [for clarity, s denotes Δs in (1)], and $p(c)$ the class prior that is independent on the action (uniform distribution in the general case). The likelihood $p(s|c,a)$

is given by the GP previously learned for each class. The estimated object class is then determined by the maximum a posteriori probability (MAP):

$$c^* = argmax_c\, p(c|s,a) \tag{3}$$

Outcomes may be composed of different features $s_1 ... s_F$, such as in our case where we have two; $s_1 = translation$ and $s_2 = rotation$. If we assume these random variables to be independent, as done in the experiments, these can be combined by multiplication (naive Bayes classifier):

$$p(c|s_1, s_2, ... s_F, a) = \frac{\prod_{i=1}^{F} p(s_i|c,a)p(c)}{\sum_{c'} \prod_{i=1}^{F} p(s_i|c',a)p(c')} \tag{4}$$

Observations from several actions can also be combined by multiplying all the probabilities of the sequence. We expect that the more actions we take, the higher the confidence we get by making more observations. To proceed with a sequence of actions, the resulting classification after the execution of an action at time step t can be used as the new prior of each class in the next step $t+1$. Equation (2) can then be rewritten as:

$$p_t(c|s,a) = \frac{p_t(s|c,a)p_t(c)}{p_t(s,a)} \tag{5}$$

where $p_t(c) = p_{t-1}(c|s,a)$, and the initial prior $p_0(c)$ is a uniform distribution. At every time step, the features are combined as described in (4).

D. Action selection

The classification of a new object is done by observing the effects of an action or multiple actions in sequence. To determine which action should be taken, one can look at the outcome distributions of the known classes. As the outcome is not known in advance for a new object, it is preferable to choose the action that will discriminate the classes with the highest confidence, or in other terms, the action that provides the highest expected information gain. This can be seen as if the agent knows which is the most appropriate action for this purpose, given its experience represented by the eSMCs. Based on principles of information theory, we use the entropy of the class distributions as a criterion to measure the uncertainty related to each action. By looking for the action associated with the lowest predicted entropy, we can select the one likely to classify the object with the highest confidence.

In our case, we look for the class given by the highest probability of $p(C|S, A = a)$ for action a, where C is a discrete random variable for the class, and S a random variable corresponding to possible outcomes. Therefore, we measure the conditional entropy of C given S, for the action a:

$$H(C|S, A=a) = -\int \sum_c p(c,s|a)\, log\, p(c|s,a)\, ds \tag{6}$$

Using Bayes' rule in the joint distribution, and rearranging the terms, we reformulate this equation as follows:

$$\begin{aligned} H(C|S, A=a) &= -\int \sum_c p(s|c,a)p(c)\, log\, p(c|s,a)\, ds \\ &= -\sum_c p(c) \int log\, p(c|s,a)\, p(s|c,a)\, ds \end{aligned}$$

As the likelihood $p(s|c,a)$ has been learned with the GPs, the integral can be approximated with a Monte Carlo method, drawing a finite set of samples from each class distribution. The result is normalized with the total number of samples N (N=100 in the experiments):

$$\hat{H}(C|S, A=a) = -\frac{1}{N} \sum_c p(c) \sum_{k=1}^{N} log\, p(c|s_k^c, a) \tag{7}$$

where $p(c|s_k, a)$ can be computed with Bayes' rule as defined in (2), using the likelihood $p(s_k^c|c,a)$ normalized over all the classes, each s_k^c being sampled from the distribution for a given class c and action a. To combine several features, we use the variant defined in (4). Finally, we can select the optimal action as the one resulting in the lowest entropy:

$$a^* = argmin_a\, \hat{H}(C|S, A=a) \tag{8}$$

At every time step the optimal action is evaluated, and the cumulated posterior probability of each class is then updated with (5). Similarly, these posterior probabilities are used as class priors in (7) for the next time step.

IV. Experiments

After presenting the experimental setup, we review results from classification of the object instances, with optimal action selection, followed by unsupervised categorization.

A. Setup

In the experiments objects of different shapes are placed on a table and pushed one by one with a robotic configuration (Kuka Arm & Schunk Hand) shown in Fig. 3. The effects are observed by a fixed Kinect camera, providing color and depth information.

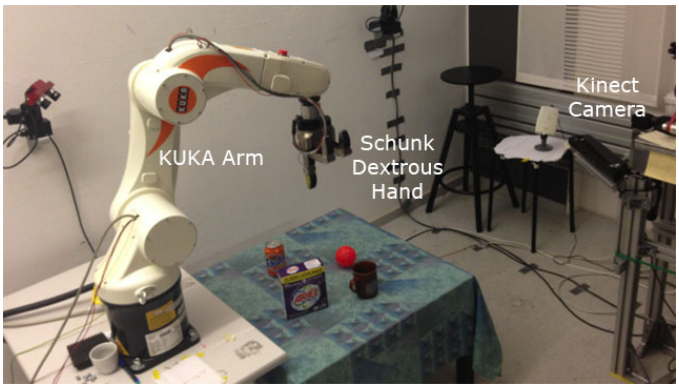

Fig. 3. Setup with Kuka Arm & Schunk Hand. The Kinect camera is fixed and points towards the table where the objects lie.

A total of 12 different objects (see Fig. 4) are considered, organized in four groups (balls, boxes, cylinders and miscellaneous) of three objects each. The miscellaneous group contains objects which present different shapes than the three main groups.

Fig. 4. The 12 objects considered in the experiments, with three instances for each of the four groups: balls, boxes, cylinders and miscellaneous.

The learning phase consists of pushing the objects horizontally at different locations such as described in Section III. Five different actions are considered, as illustrated in Fig. 2. These are represented by the variable a_X taking values in the discrete set $\{-1, -1/2, 0, +1/2, +1\}$. The model allows a continuous action space, but is discretized for the experiments, in order to separate the noise level in outcomes from the variability of the modelled functions in the interpretation of the data.

B. Feature extraction

The object specific sensory data from which the effect of actions are studied are extracted as follows. First an object is segmented from the image using object segmentation [23]. We consider this a low-level function the robot has already learned (modality-related eSMCs). From color and depth data, a 3D point cloud is built, from which the center of mass of the object is estimated. However, only the visible part of the object is taken into account, which means that the center is likely to be incorrect. A dominating plane is found with RANSAC [22] and is used to determine the position of the push. The borders are estimated by a bounding box, after removing 5% of the most distant points along the Z-axis, to filter out noisy data from the segmentation.

For both features, translation and rotation, the effect of a pushing action is measured as differences between values before and after the push, as mentioned in Section III. From the segmented point cloud, we have a vector $P = (x_p, y_p, z_p)$ for the 3D centroid, and four variables a, b, c, d describing the plane defined by $ax + by + cz + d = 0$. The features are described by two single variables $(\Delta s_p, \Delta s_o)$, taking the Euclidian distance of the change in position, and the dihedral angle between the two planes as the rotation. Hence we define $\Delta s_p = ||P_2 - P_1||$ where P_1 and P_2 are the positions before and after push, and $\Delta s_o = \angle(n_1, n_2)$, where n_1 and n_2 are the corresponding normal vectors of the plane $n = (a, b, c)^T$.

C. Gaussian Process models

After pushing each object repeatedly for the given actions, a training dataset with about 720 samples is compiled for analysis. Fig. 5 and 6 illustrate the learned functions for the given classes, respectively for the translation and rotation features. The points correspond to the training data, while the mean and variance of the GPs are described respectively by the lines and the grey areas with a 2σ deviation, representing 95% of confidence.

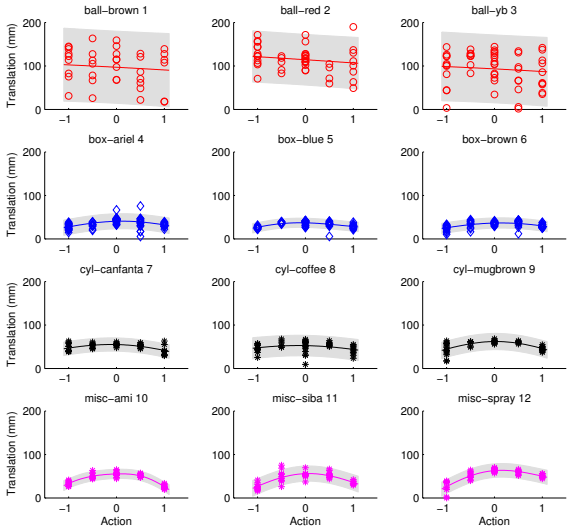

Fig. 5. Learned function of translation for each object, given action. Each row corresponds to a different group: ball, box, cylinder, miscellaneous.

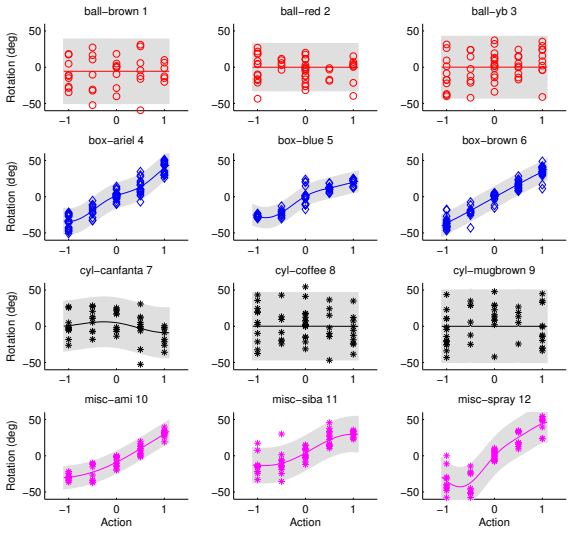

Fig. 6. Learned function of rotation for each object, given action. Each row corresponds to a different group: ball, box, cylinder, miscellaneous.

As expected, the balls roll with a higher amplitude than the other objects that slide a shorter distance. The rotation feature is only relevant for the boxes and the miscellaneous objects, leading to a small variance, while its variance is much higher for the other objects. This can be explained

by the detection of the plane, which is not suitable for spherical or cylindrical objects. However, this characteristic is captured by the variance, which can still be exploited for classification after multiple pushes. For the observed action effects, the miscellaneous objects look mostly like boxes. To what degree they can be categorized to boxes is studied in further experiments.

D. Classification with optimal action selection

By applying a sequence of pushes to a new object, the object is classified using the cumulated posterior probabilities from (5). To verify if the entropy-based action selection policy (MinEnt) improves classification, we compare this policy where the action is given by (8) to randomly selected actions (Random). Since the dataset is limited in number of samples, we proceed by bootstrapping, instead of having two separate training and test sets. For every iteration, the test actions are sampled from the dataset with replacement. Fig. 7 presents the confusion matrices after 10 epochs (number of pushes in sequence), averaged over 100 iterations.

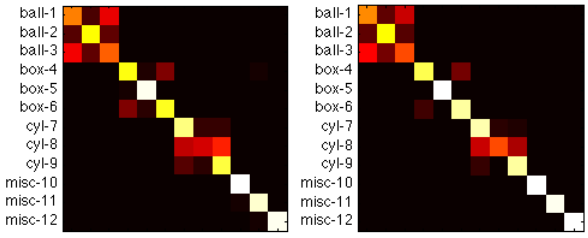

Fig. 7. Confusion matrix for the 12 objects after 10 epochs, taking a Random or MinEnt policy.

At first glance, the result of the classification looks similar in both cases. Some of the objects are classified correctly according to their instance, while others are confused within their respective groups. For the three main groups (balls, boxes and cylinders) we would expect such a confusion, due to similarities in object shapes and the nature of our feature space. We will see later how the objects can be categorized based on these similarities. The miscellaneous objects are recognized individually though. However, the distinctions would not be as obvious with fewer epochs.

To better see the effect of the action policy, we measure the average posterior probabilities of the predicted class given by (3) and the actual class, with classes defined by the original groups of objects. The results shown in Fig. 8 illustrate the benefit of the optimal selection, where both the predicted and the actual classes are found with higher posterior probabilities. Not only the classification can be done faster, but the correct class has also a higher chance of being found earlier. For a prediction target of 80% we save about 2-3 epochs, representing about 20-25% improvement in terms of speed without loss of quality. After 10 epochs, we gain about 5% of confidence. We expect the difference to be even greater if the actions were more distinctive to separate the classes, or if the action space were more complex.

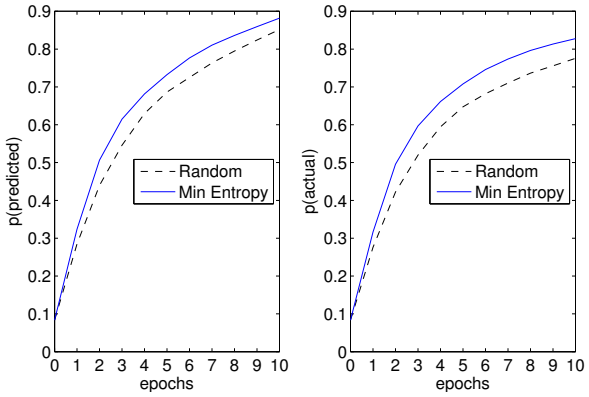

Fig. 8. Average probability of the predicted and actual classes, taking a Random or MinEnt policy. The benefit is given by the difference.

E. Unsupervised categorization

To regroup the objects into categories, we need a criterion measuring how similar two objects are. As the distributions are Gaussians, we compute the Kullback-Leibler divergence of two objects i and j given each feature and action. A symmetric distance measure is then derived as $d_{ij} = KL(i,j) + KL(j,i)$. We average over the features and actions to obtain the similarity matrix shown to the left in Fig. 9. From this, we build a spectral embedding using the Normalized Cuts algorithm [24], with the adjacency matrix $S_{ij} = exp(-d_{ij}^2/\sigma_s^2)$ where d_{ij} is the distance value and σ_s^2 the variance of the similarity matrix.

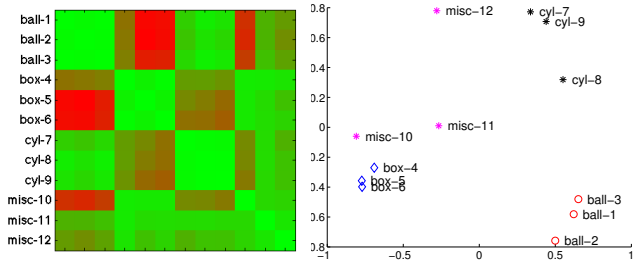

Fig. 9. Similarity matrix (left) based on a symmetric distance measure of the KL-divergence, and the resulting spectral embedding (right).

The result is shown to the right in Fig. 9, where the three main groups are well separated. By creating a minimum spanning tree from pair-wise distances (in the embedding), we cluster the objects into categories, by gradually splitting the tree, removing edges in order of decreasing distances. If this is done with 3 to 5 categories, each main group will remain in the same category. First, the spray bottle (*misc-12*) will be separated from the cylinders and then the oval-shaped container (*misc-11*) from the boxes. The next object to be separated into its own category is *cyl-8*, not the hexagonal box (*misc-10*) that remains grouped to the boxes.

Once the categories have been defined, we retrain the whole model by merging the samples of objects belonging to the same category. We classify the obtained categories as done previously with the individual objects. The benefit of MinEnt is now less significant in terms of speed, despite

a slight improvement of about 5% in confidence. With a smaller number of categories, fewer epochs are required, usually about 3 or 4, illustrating the fact that each category captures well the characteristics of their objects. The results after up to 5 epochs using MinEnt, averaged over 100 iterations, are presented in Fig. 10.

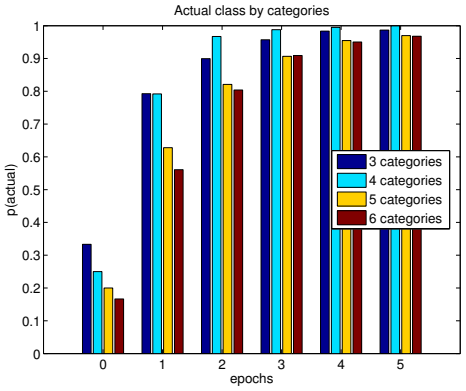

Fig. 10. Probability of the actual class for 3 to 6 categories, after 5 epochs.

V. Conclusion and Future Work

This paper has presented an approach where knowledge is acquired through interaction with objects using a single pushing behavior. By observing the effects of different actions applied to objects, sensorimotor contingencies were learned using a model based on Gaussian Processes with an explicit representation of the effect. The possibility of making predictions with a measure of confidence was tested for classification. To select suitable actions for classification, action selection can be driven by a measure of entropy. The presented experiments show the benefit of following such a policy. By looking at the similarities of eSMCs obtained for different objects, categories can be discovered and learned with the same model.

In the future, the scalability has to be evaluated in terms of categories, dimensions of the action space, and combination of different behaviors. In relation to tool affordances, another aspect worth investigating is the different morphologies that a robotic hand can take, and the integration of different sensory modalities such as tactile measurements. When it comes to the learning process, features were separately learned for each object through supervised learning. Instead of doing the categorization a posteriori, an alternative would be to use fully unsupervised learning, where this knowledge is instead acquired directly from the observations without labeling the classes. The features were also assumed to be independent, while in reality they are not. It remains to be studied whether such a simplification is in fact justified.

Acknowledgments

This work was supported by the EU through the project eSMCs (FP7-IST-270212) and the Swedish Research Council.

References

[1] F. G. Ashby and W. T. Maddox, "Human category learning," *Annu Rev Psychol*, vol. 56, pp. 149–178, 2005.
[2] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *IEEE CVPR*, 2003.
[3] G. Bouchard and B. Triggs, "Hierarchical part-based visual object categorization," in *IEEE CVPR*, pp. 710–715, 2005.
[4] S. Savarese and L. Fei-Fei, "3D generic object categorization, localization and pose estimation," in *IEEE ICCV*, pp. 1–8, 2007.
[5] V. Kruger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: a review on action recognition and mapping," *Advanced Robotics*, vol. 21, no. 13, pp. 1473–1501, 2007.
[6] H. Kjellström, J. Romero, and D. Kragic, "Visual object-action recognition: Inferring object affordances from human demonstration," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 81–90, 2011.
[7] J. K. O'Regan and A. Noë, "A sensorimotor account of vision and visual consciousness," *Behavioral and Brain Sciences 24(5)*, pp. 939–1031, 2001.
[8] 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, "Object–Action Complexes: Grounded abstractions of sensory–motor processes," *Robotics and Autonomous Systems*, vol. 59, pp. 740–757, Oct. 2011.
[9] A. Duff, M. Sanchez-Fibla, and P. F. M. J. Verschure, "A biologically based model for the integration of sensory-motor contingencies in rules and plans: A prefrontal cortex based extension of the Distributed Adaptive Control architecture," *Brain Research Bulletin*, vol. 85, pp. 289–304, 2011.
[10] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cogn Process*, vol. 12, pp. 319–340, Apr. 2011.
[11] J. J. Gibson, *The Ecological Approach to Visual Perception*. Psychology Press, 1979.
[12] A. Stoytchev, "Behavior-Grounded Representation of Tool Affordances," in *IEEE ICRA*, pp. 3071–3076, 2005.
[13] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action - initial steps towards artificial cognition," in *IEEE ICRA*, pp. 3140–3145, 2003.
[14] M. Sanchez-Fibla, A. Duff, and P. F. M. J. Verschure, "The acquisition of intentionally indexed and object centered affordance gradients: A biomimetic controller and mobile robotics benchmark," in *IEEE/RSJ IROS*, pp. 1115–1121, 2011.
[15] E. Ugur, E. Oztop, and E. Sahin, "Goal emulation and planning in perceptual space using learned affordances," *Robotics and Autonomous Systems*, vol. 59, pp. 580–595, July 2011.
[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.
[17] G. Metta and P. Fitzpatrick, "Better Vision through Manipulation," *Adaptive Behavior*, pp. 109–128, 2003.
[18] A. Maye and A. K. Engel, "A discrete computational model of sensorimotor contingencies for object perception and control of behavior," *IEEE ICRA*, 2011.
[19] J. Sinapov, C. Schenck, K. Staley, V. Sukhoy, and A. Stoytchev, "Grounding semantic categories in behavioral interactions: Experiments with 100 objects," *Robotics and Autonomous Systems*, Nov. 2012.
[20] A. Stoytchev, "Some Basic Principles of Developmental Robotics," *IEEE TAMD*, vol. 1, no. 2, pp. 122–130, 2009.
[21] N. Bergström, C. H. Ek, M. Björkman, and D. Kragic, "Scene understanding through autonomous interactive perception," in *ICVS*, pp. 153–162, 2011.
[22] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24(6), pp. 381–395, 1981.
[23] M. Björkman and D. Kragic, "Active 3D scene segmentation and detection of unknown objects," *IEEE ICRA*, 2010.
[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE TPAMI*, vol. 22, no. 8, pp. 888–905, 2000.