Using Relational Histogram Features and Action Labelled Data to Learn Preconditions for Means-End Actions

Severin Fichtl[1,2], Dirk Kraft[2], Norbert Krüger[2] and Frank Guerin[1]

Abstract— **The outcome of many complex manipulation actions is contingent on the spatial relationships among pairs of objects, e.g. if an object is "inside" or "on top" of another. Recognising these spatial relationships requires a vision system which can extract appropriate features from the vision input that capture and represent the spatial relationships in an easily accessible way. We are interested in learning to predict the success of "means end" actions that manipulate two objects at once, from exploratory actions, and the observed sensorimotor contingencies. In this paper, we use relational histogram features and illustrate their effect on learning to predict a variety of "means end" actions' outcomes. The results show that our vision features can make the learning problem significantly easier, leading to increased learning rates and higher maximum performance. This work is in particular important for robots that need to reliably predict the success probability of their multi object manipulating action repertoire in novel scenes.**

I. INTRODUCTION

We want to learn to predict the success of means-end actions (i.e.where one action is used in order to facilitate another) grounded in sensorimotor contingencies (SMC). The outcome of means-end actions in complex environments is contingent on the spatial relationships among the manipulated objects. Robots performing means-end actions in complex environments need a sound understanding of how these spatial relations affect the success of their actions. Learning how spatial relationships affect action outcomes can be slow and difficult, if the state space representation does not capture and represent the important information appropriately and easily accessible. In previous work we have developed a histogram feature which is good at capturing information about spatial relationships for a variety of object pairs [1]. This paper uses that feature, but instead of learning to recognise manually defined spatial relationships as done in [1], it focuses on the learning of preconditions that determine the outcome of actions that are based on spatial relationships such as pushing an object which is 'under another', or lifting an object which 'contains another'.

The action preconditions we learn are similar to affordances of object pairs. It is the spatial relationship between the pair of objects that determines what actions are possible, i.e. what actions are afforded by the pair of objects in the current spatial configuration. These spatial relationship based object pair affordances are important for service robots that interact in complex environments like everyday home environments. For example a robot carrying a kitchen tray with cups and plates on the tray needs good knowledge about the spatial relations between the objects.

Our approach to learning action preconditions is to ground the learning in the robot's own sensorimotor contingencies. This grounding helps to avoid problems with classical AI which relied on a humans' judgement of what knowledge and representation might be appropriate for the robot [2]. Hand-coded knowledge tended to result in brittle systems (i.e. these systems broke down when the task went outside the scenarios that the human had foreseen). Knowledge learnt from action can be expected to be more useful to the robot, and more robust, in line with the "Verification Principle" [3]:

> "An AI system can create and maintain knowledge only to the extent that it can verify that knowledge itself."

To enable the robot to efficiently learn important spatial relationships that afford certain actions, we use as state space representation the RGB-D sensor based visual histogram features introduced in [1]. In [1] we used these features to learn to recognise spatial relationships among objects from data with labels that are based on the hand defined spatial relationships. With this, our current work continues our previous work from [4] in which we learnt classifiers predicting the outcome of actions but used a less adequate state space representation, making the learning of preconditions of multi-object manipulation actions difficult. The main new contribution here is that we demonstrate the usefulness of our histogram features for learning action preconditions, which, as we show, implicitly capture spatial relationships among objects. In particular, we aim to demonstrate how the learning of preconditions for manipulation actions can be improved in both, learning speed and maximum reachable performance by using relational histogram features as vision based state space representation. The goal of the classifiers trained here is to accurately predict whether the action associated to the classifier can be executed successfully in a given scene, depending on the objects and their relative positions.

While our histogram features make it easier to learn some preconditions that are contingent on the spatial relationships among objects, we show that they can also limit a classifier's performance if not adding relevant information to the learning task, due to their large amount of inputs.

The remainder of this paper is structured as follows: Section II reviews the literature. Sections III and IV describe the methods and experiments used. Section V presents the results. Section VI concludes the work.

*This work was supported by the EU Cognitive Systems project XPERIENCE (FP7-ICT-270273)

[1]University of Aberdeen, Aberdeen, United Kingdom (f.guerin@abdn.ac.uk)

[2]University of Southern Denmark, Odense, Denmark (fichtl@mmmi.sdu.dk, kraft@mmmi.sdu.dk, norbert@mmmi.sdu.dk)

II. RELATED WORK

Our concept of a precondition is loosely related to the notion of an affordance [5] used as a planning operator, which has been well studied within the field of developmental robotics (see e.g. [6], [7]). We focus on learning classifiers that predict the success of actions that are contingent on the spatial relationship among a pair of objects. This is quite close to work on learning relational "affordances" [8], [9] (i.e. not just the affordance of a single object, but a pair).

The relational features considered in the work of [8] are the distance between two objects, their relative positioning to one another and whether or not they are touching. Our relational histogram approach provides a much more detailed state space representation, that encodes not only object positions, orientations and sizes, but also relative spatial relations between surface patches of the two objects. With this, our relational histograms encode whether, for example, one object encloses (parts of) another object, or whether one object is in front or above another.

Ugur et al. [9] follow a different approach to learn "paired object affordances". They learn to predict the outcome of a "stacking" action where one object is placed on top of another. The effects of the action observed were "tumbled over", "piled up" (i.e. successfully stacked), "covered" (when the top object is a cup that covers and completely contains the lower object), and "inserted in" (when the lower object is the container and the top object drops into it). They attempted to learn the stacking effects with 18 pairs of objects. As input to their classifiers they used the combined set of shape features of each object. The shape features consist of histograms of normal vectors for object surface points. Given these visual object shape features for both objects, their classifiers learnt to predict the effect of the stacking action. This is a significant difference in our approaches as we are looking at the *spatial relationship of two objects* in order to determine the effect of an action, whereas Ugur et al. are looking at the features of the two objects before they are put in a relationship, in order to determine what relationship they might end up in after an action.

Our work is also related to infant development. In the period from six months of age through to two years human infants undergo significant development in their skills and understanding relating to physical world objects and their manipulation. Observations of infants show that, from as early as three months of age, they possess a repertoire of behaviours which they apply to various objects or surfaces they encounter [10], [11], [12]. Each such behaviour could be seen as roughly analogous to a planning operator in Artificial Intelligence (like an "OAC" in [13]), because there are situations which make them likely to be executed (like the precondition of a planning operator), and expected effects (postcondition), as well as some motor control program describing the behaviour executed. As infants develop they solve the problems of (i) identifying when a new behaviour should be created, (ii) learning the new precondition, (iii) postcondition, and (iv) motor program for the new behaviour.

In this paper, we focus on learning the precondition for a new behaviour. This is a particularly interesting problem in the case of means-ends behaviours (i.e. where one action is used in order to facilitate another [14]), because it is through learning means-ends behaviours that infants begin to learn about relationships between objects [15]. The precondition must capture the relationship between objects which determines where the behaviour works or does not work.

III. METHODS

In this work, we collected data using a physically realistic simulation environment [16] designed for robot simulations and a vision system using a simulated Kinect camera [17], inclusive the noise of real Kinect devices [18]. This gives us data about the depth to the objects in our 3D scene similar to what we would have obtained from a real Kinect looking at a real scene with 3D objects. As robot we used a simulated six degrees of freedom (DOF) arm mounted on a table with a two finger gripper as its hand (see Fig. 1).

A. The Perception System

The Kinect sensor is mounted opposite to the robot, looking down towards the robot, as illustrated in Fig. 2. Using the Kinect data we calculate a high resolution 3D point cloud of the scene (as illustrated in the right image of Fig. 2).

In this work, we used a trivial method for object segmentation. For this simple method to work, it is assumed that the objects are coloured in one of a known set of colours. This is a strong assumption also made by others, e.g. Rosman and Ramamoorthy [19], but it could be relaxed by using more sophisticated segmentation methods (e.g. [20]), which take into consideration factors like discontinuities of surface curvatures and colour differences. After segmentation, each object is assigned its unique set of points.

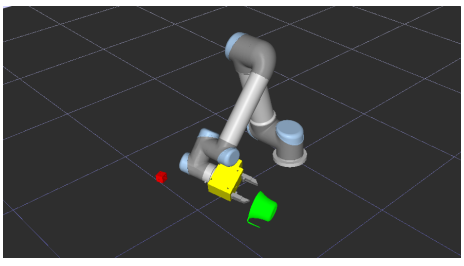

Fig. 1. Illustration of the simulated robot grasping a cup.

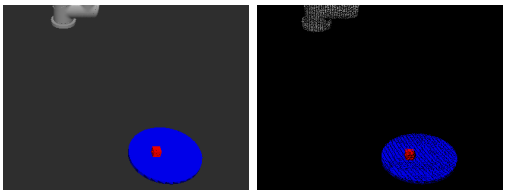

Fig. 2. Kinect camera looking at workspace. The left image shows a die on a plate-like object, with the robot "shoulder" visible at the top. The right image shows the point cloud representation of the same scene as on the left.

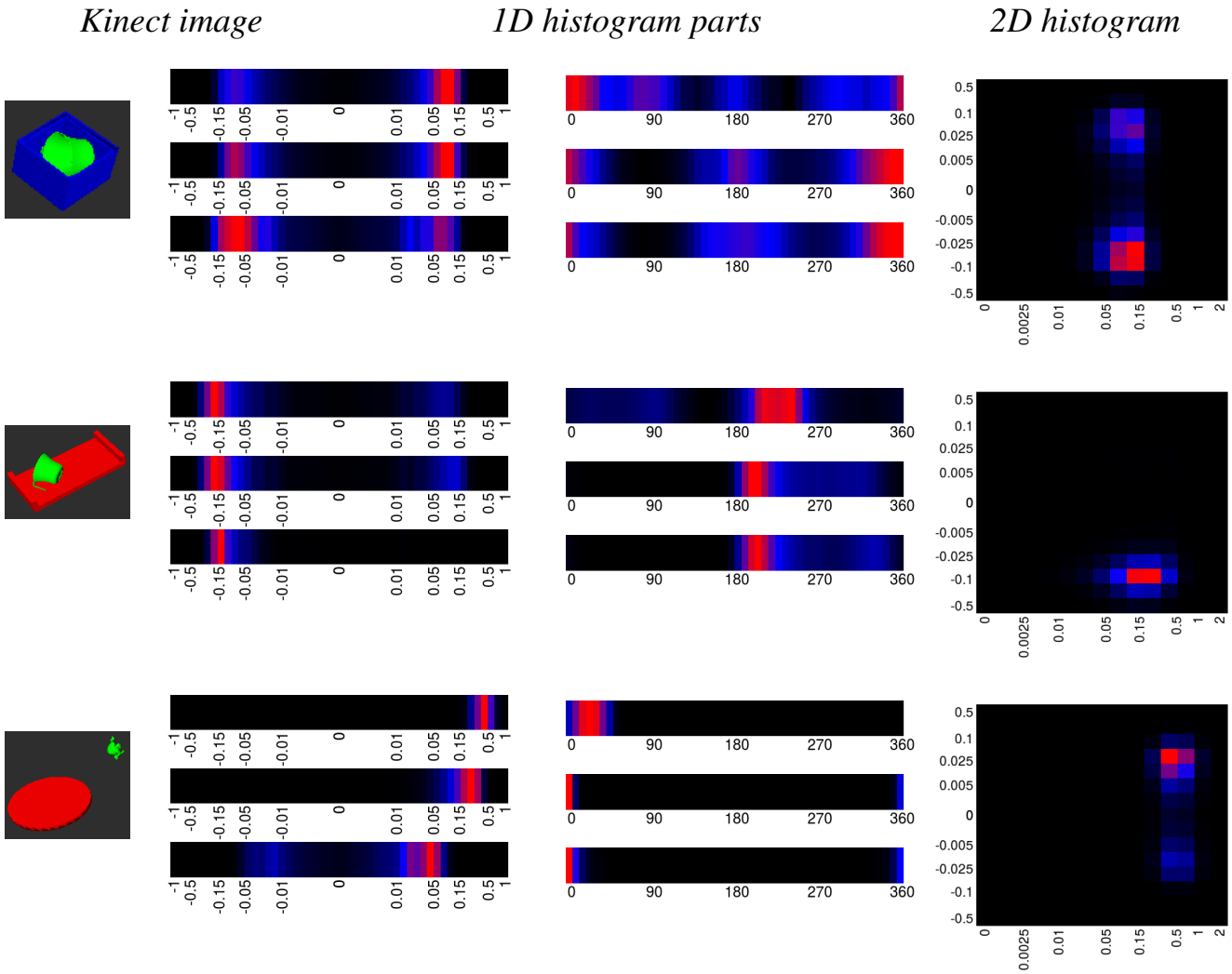

Fig. 3. 1D and 2D histogram illustrations of "inside", "ontop" and "beside" cases. On the left is the image recorded by the Kinect camera. In the middle/left are the X/Y/Z distance parts of the 1D histogram. In the middle/right are the XY/XZ/YZ angle relation parts of the 1D histogram. On the right side is the 2D histogram.

We apply our learning approach on different state space representations to compare their efficiencies with regard to representing the state space in a accessible way. From each segmented point cloud, our vision system extracts, using PCA, approximations of the position of the object's centre of gravity, the object's orientation and the object's dimensions. Thus, each segmented object is described by nine variables. These are X, Y and Z for the position, Roll, Pitch and Yaw for the orientation and three size values for the elongation along the objects three PCA axes. These variables are the baseline vision state space representation. We will refer to this as the *PCA* state space in the remainder of this paper.

We then use the segmented point clouds, to create relational histograms to capture the spatial relations between objects. These relational histograms form a relational space into which the absolute geometric information (3D position and orientation) of the 3D points is transferred. To achieve this transfer, we define a set of relational features which encode the spatial relationship structure of the objects in the scene.

More specifically, for each scene we have two point clouds Π^1 and Π^2 representing the segmented objects 1 and 2 in the scene. For each cross object pair of points of the form $\Pi^1_i \oplus \Pi^2_j$ we calculate four *Euclidean* distances $R_d(\Pi^1_i, \Pi^2_j)$ (the Euclidean distances along the X, Y and Z axes respectively and in the XY plane, where the X axis goes towards the front of the robot, the Y axis goes towards the left and the Z axis represents the height) and three *Angle Relations* $R_a(\Pi^1_i, \Pi^2_j)$ (the line through the two points is projected onto one of the planes XY, XZ, or YZ, and we look at the angle between the projected line and the axes X, Y and Z respectively). Fig. 5 illustrates this process for two example point clouds projected into the XY plane. The amount of feature vectors, describing the relation between the two objects in the scene, is variable and determined by the amount of points extracted by the vision system. To obtain a generic input vector of fixed length to apply *Supervised Learning Algorithms* on, we compute 1-, 2-, and 3 dimensional relational histograms from the calculated vectors $R_d(\Pi^1_i, \Pi^2_j)$ and $R_a(\Pi^1_i, \Pi^2_j)$ and use these as learning input.

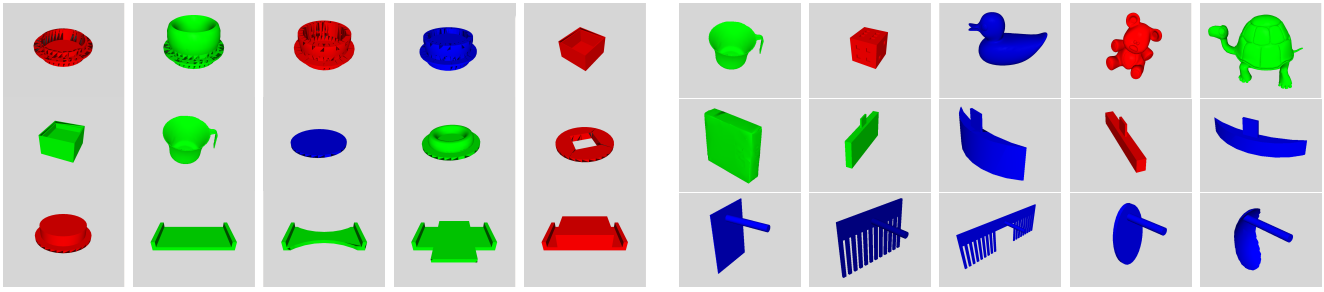

Fig. 4. Illustration of the objects used for the experiments of this work. On the left side are the base objects, on the right side are per row: toys, obstacles and rakes.

TABLE I
LIST OF ACTIONS

Action	Motor Program	Goal
Lift	Grasp base object and lift it.	Toy object is lifted.
Move	Move hand to toy object and push it aside.	Toy & base objects have moved aside.
Pull	Grasp base object and pull it.	Toy object is pulled closer.
Push	Grasp base object and push it.	Toy object is pushed further away
Rake	Put rake head behind toy object and pull.	Toy object has been brought closer.
Take	Grasp toy object and lift it.	Toy object is lifted.
Pour	Grasp base object and lift & tilt it.	Toy object is lifted.
Slide	Grasp base object and lift & tilt it.	Toy object has moved but not been lifted.
Unobstruct	Grasp base/obstacle object and push aside.	The toy object that wasn't reachable before, is now reachable.

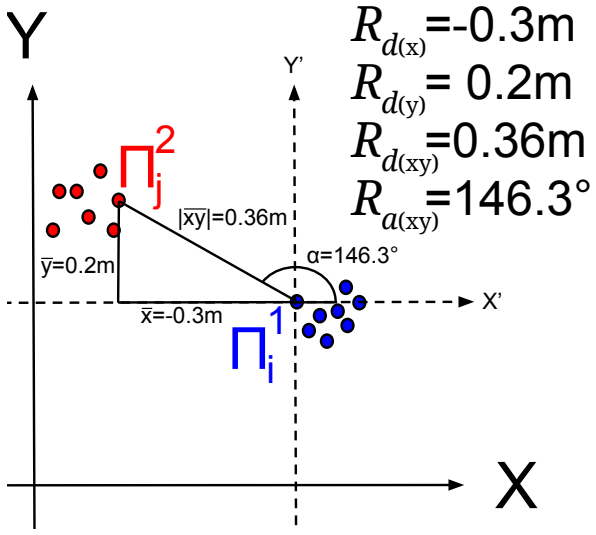

Fig. 5. Illustration of the variable extraction process that leads from point clouds to Histograms. Visualised as projections into the XY plane are two point cloud examples Π^1 and Π^2, from which the points i and j respectively have been selected to calculate distances and angles.

The 1D relational histogram are a combination of six individual 1D histograms, capturing the distances between points along each of the three main axes and the angle relations between points in each of the three planes spanned by the three main axes.

The 2D relational histograms capture the absolute distance of inter-object pairs of points in the XY plane and puts it into relation with the height difference (i.e. difference along the Z axis).

The 3D relational histograms capture the distances between points among three dimensions, in a similar fashion as the 2D histogram does for two dimensions. For the 3D histogram, however, we used the actual position differences among all three main axes (X, Y and Z).

When creating the histograms from the distance values we apply pre- and post-processing methods on the input data and on the histograms to increase robustness and performance. These methods are logarithmic scaling of input data and histogram normalisation and smoothing. Fig. 3 illustrates the final histograms for three different scenes. See [1] for more details on pre- and post-processing methods and their effect compared to histograms without pre- and post-processing.

The state space is then made up from the 18 values of the PCA state space representation on their own, or combined with one of the relational histograms, e.g. 18 + 300 for the 1D case. We will refer to these as the *PCA* or *1D*, *2D* or *3D histogram* state spaces respectively.

IV. EXPERIMENTAL SETUP

In the following subsections we will describe in more detail the objects (see Section IV-A) used during the experiments and the actions (see Section IV-B) executed on them.

A. Objects

We use 29 different objects in our experiments (see Fig. 4). These 29 objects can belong to four different object categories[1].

1. Toys (5 Objects)
2. Bases (15 Objects)
3. Obstacles (5 Objects)
4. Rakes (5 Objects)

[1]One Object (Cup) is member of two Groups (Toys and Supports/Containers)

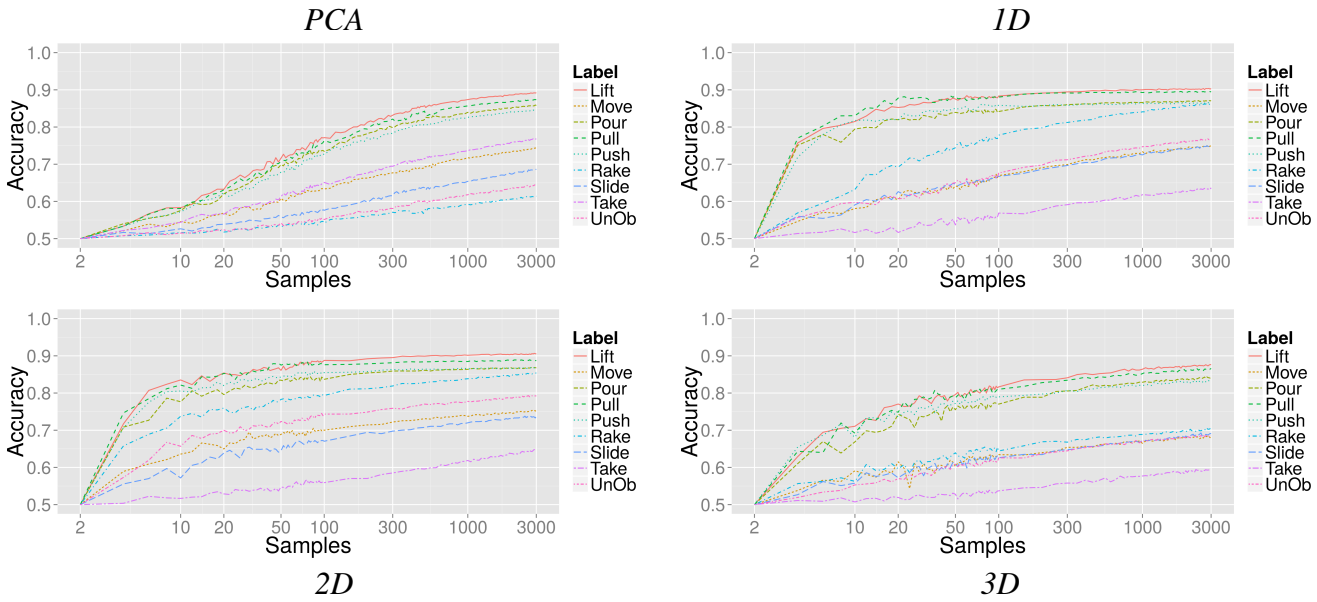

Fig. 6. Illustration of the precondition learning speed for nine actions in different state spaces. Note that the X-axes are logarithmic scaled

We used exactly two objects for every experiment, where one object was a toy and the other object was a member of a different group. The workspace of the robot forms a semi-circle with a radius of 1.8 metres with the robot arm placed in the centre of the semi-circle. The robot has a maximum reach of approximately 1.2 meters. The two objects are randomly distributed in the workspace area. An exception to this were experiments with the rake objects. Rakes were attached to the robot arm, replacing the gripper.

B. Actions

We equipped the robot with nine actions it could execute. The actions, their motor program and their goals are briefly described in Table I.

The action's motor control follows a naive forward kinematics approach. A target position for the gripper to manipulate an object is selected and forward kinematics used to find appropriate robot arm joint angles. The joint angles are then driven directly to the calculated target values. As no path planning is applied, some action execution trials lead to error states. For some toy objects the grasp targets are unreliable, leading to low success rates of actions that primarily manipulate toy objects.

Out of all data samples that were collected in the simulations, we picked a smaller data set with approximately 10.000 samples per action, with 50% positive and 50% negative samples. The positive samples were selected uniformly. The negative samples were selected with a bias such that the distribution of distances between the centres of gravity (COG) of the two objects is similar within the groups of positive and negative samples. We focus on negative samples with centre of gravity distance distributions similar to that of positive samples to make the differentiation between positive and negative cases less trivial.

V. RESULTS

Fig. 6 illustrates the learning rate for the different actions using different state space representations for up to 3000 random samples on a logarithmic scaled X-axis.

As can be seen by comparing the different subfigures, an appropriate state space representation is of significant importance for learning. The more expressive state representation using histograms massively outperforms learning without histograms.

These results are based on histogram features, where the 1D histograms have 300 variables, the 2D histograms have 225 variables and the 3D histograms have 1000 variables. These histogram sizes are a compromise between size and resolution and where found to give best results. In this case, the 1D and 2D histograms allow for the fastest learning, with both being about equally good for the actions that were learnt fastest (see e.g. the Lift action in Fig. 7), and the 2D histogram a bit better for the actions that were learnt at a moderate speed (see e.g. the Unobstruct action in Fig. 7). We believe that the 1D histogram is a bit more generic and should lead to increased performance for a larger spectrum of potential actions. Whereas the 2D histogram is especially designed to differentiate between two objects being ontop or inside of each other vs. beside each other. This potentially gives it the leading edge compared to the 1D histogram for our set of actions where the success relies on the difference between ontop, inside and beside. The 3D histogram is the most simplistic approach to histograms with the highest dimensionality and at the same time with the lowest resolution. All three properties making learning from the 3D histogram more difficult than learning from 1D or 2D histograms, but learning from 3D histograms still outperforms learning from the PCA representation.

The "Take" action serves as a good example of the potential shortcomings of hand designed state spaces. The extended state space representation does not benefit the "Take" precondition classifier. Instead, the increased amount of input variables causes a decrease of its learning speed (curse of dimensionality). The reason for this is likely to be the increased "noise" as the toy grasping success rate of the take action is not very high. At the same time, the likelihood of the inside relation is fairly low. This means that most negative samples are not negative due to the inside relation, which could be better recognised in the extended state spaces, but due to the noise in the motor program success, e.g. the low success rate of grasping toy objects.

Fig. 7 illustrates the degree to which the success prediction of the different actions correlates to the actual spatial relation between the objects, which we labelled based on the manually defined spatial relationships for this comparison as "ontop", "inside" or "beside". One can see that many of the precondition classifier capture knowledge that strongly correlates to one or more of the manually defined spatial relationships

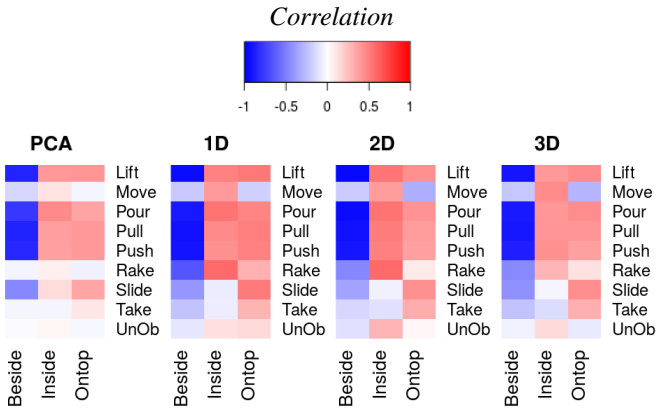

Fig. 7. Correlations between Actions and the heuristically learnt categories.

VI. CONCLUSIONS

In this paper we have demonstrated that our relational histogram features can significantly increase the learning speed of precondition classifiers for complex actions manipulating pairs of objects, compared to learning from the simple PCA features only. This achievement is possible when the outcome of actions does rely on the spatial relations that our relational histogram features are capturing. This highlights the importance of appropriate features that describe the state space in an *informative and accessible* way.

The precondition classifiers were found to implicitly capture categories such as 'on top', 'not on top' or 'inside'. In our ongoing research we attempt to extract this implicit knowledge into explicit symbolical category knowledge. This might serve as a first step towards higher level symbolic reasoning and planning, similar to the pathway in infants from simple action development through sensorimotor contingencies to higher level reasoning with abstract ideas e.g. about containers and containment and object permanence.

In future work we intend to evaluate the spatial relationship discriminating performance of our relational histogram features in a real world set up in a similar way as we did in [1]. But instead of a simulation we would work with real Kinect sensors and more sophisticated segmentation algorithms such as [20].

REFERENCES

[1] S. Fichtl, A. McManus, W. Mustafa, D. Kraft, N. Kruger, and F. Guerin, "Learning spatial relationships from 3D vision using histograms," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 501–508.

[2] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, pp. 139–159, 1991.

[3] R. S. Sutton, "Verification, The Key to AI," p. 1, 2001. [Online]. Available: http://www.cs.ualberta.ca/ sutton/IncIdeas/KeytoAI.html

[4] S. Fichtl, J. Alexander, D. Kraft, J. Jorgensen, N. Krüger, and F. Guerin, "Learning object relationships which determine the outcome of actions," *Paladyn*, vol. 3, no. 4, pp. 188–199, 2012.

[5] J. J. Gibson, *The Ecological Approach To Visual Perception*. Lawrence Erlbaum Associates, 1986.

[6] 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.

[7] L. Paletta and G. Fritz, "Reinforcement learning of predictive features in affordance perception," in *Towards Affordance-Based Robot Control*, ser. Lecture Notes in Computer Science, E. Rome, J. Hertzberg, and G. Dorffner, Eds. Springer Berlin Heidelberg, 2008, vol. 4760, pp. 77–90.

[8] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *IEEE Intl. Conf. on Robotics and Automation*, 2012, pp. 4373–4378.

[9] E. Ugur, S. Szedmak, and J. Piater, "Bootstrapping paired-object affordance learning with learned single-affordance features," in *The Fourth Joint IEEE Intl. Conf. on Development and Learning and on Epigenetic Robotics (ICDL-Epirob), Genoa, Italy*, 2014, pp. 468–473.

[10] J. Piaget, *The Origins of Intelligence in Children*. London: Routledge & Kegan Paul, 1936, (French version 1936, translation 1952).

[11] J. J. Lockman, "A perception-action perspective on tool use development," *Child Development*, vol. 71, no. 1, pp. 137–144, 2000.

[12] F. Guerin, D. Kraft, and N. Krüger, "A survey of the ontogeny of tool use: from sensorimotor experience to planning." *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 1, pp. 18–45, 2013.

[13] 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, no. 10, pp. 740–757, 2011.

[14] P. Willatts, "Development of problem-solving strategies in infancy," in *Children's Strategies: Contemporary Views of Cognitive Development*, D. Bjorklund, Ed. Lawrence Erlbaum, 1990, pp. 23–66.

[15] J. Piaget, *The Construction of Reality in the Child*. London: Routledge & Kegan Paul, 1937, (French version 1937, translation 1955).

[16] J. A. Jørgensen, L.-P. Ellekilde, and H. G. Petersen, "RobWorkSim - an Open Simulator for Sensor based Grasping," in *ISR/ROBOTIK 2010 (41st International Symposium)*. VDE-Verlag, Jun. 2010.

[17] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[18] S. Olesen, S. Lyder, D. Kraft, N. Krüger, and J. Jessen, "Real-time extraction of surface patches with associated uncertainties by means of Kinect cameras," *Journal of Real-Time Image Processing*, vol. 10, no. 1, pp. 105–118, 2015.

[19] B. Rosman and S. Ramamoorthy, "Learning spatial relationships between objects," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1328–1342, Sep. 2011.

[20] K. M. Varadarajan and M. Vincze, "Object part segmentation and classification in range images for grasping," in *Advanced Robotics (ICAR), 2011 15th International Conference on*, Jun. 2011, pp. 21–27.