2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Learning Spatial Relationships From 3D Vision Using Histograms

Severin Fichtl*, Andrew McManus*, Wail Mustafa†, Dirk Kraft†, Norbert Krüger† and Frank Guerin*
*University of Aberdeen, Aberdeen AB24 3UE, Scotland, Email: s.fichtl@abdn.ac.uk, f.guerin@abdn.ac.uk
†University of Southern Denmark, 5000 Odense C, Denmark, Email: {wail, kraft, norbert}@mmmi.sdu.dk

***Abstract*— Effective robot manipulation requires a vision system which can extract features of the environment which determine what manipulation actions are possible. There is existing work in this direction under the broad banner of recognising "affordances". We are particularly interested in possibilities for actions afforded by relationships among pairs of objects. For example if an object is "inside" another or "on top" of another. For this there is a need for a vision system which can recognise such relationships in a scene. We use an approach in which a vision system first segments an image, and then considers a pair of objects to determine their physical relationship. The system extracts surface patches for each object in the segmented image, and then compiles various histograms from looking at relationships between the surface patches of one object and those of the other object. From these histograms a classifier is trained to recognise the relationship between a pair of objects. Our results identify the most promising ways to construct histograms in order to permit classification of physical relationships with high accuracy. This work is important for manipulator robots who may be presented with novel scenes and must identify the salient physical relationships in order to plan manipulation activities.**

I. Introduction and Motivation

Effective robot manipulation requires a vision system which can extract features of the environment which determine what manipulation actions are possible. The preconditions of a robot's planning operators use these features of the environment to predict the successful outcome of manipulation actions. If a robot's planning operators have reasonable accuracy then they allow the robot to sequence learnt actions to achieve a goal. The use of planning precondition here is loosely related to the notion of an affordance [1], which has been well studied within the field of developmental robotics (see e.g. [2], [3]). We are particularly interested in possibilities for actions afforded by relationships among pairs of objects. The physical relationship between a pair of objects is very important for manipulation, because many relationships determine the outcome of an action. For example if one object rests "on" another then pulling the lower one also causes the upper one to move. If one object is contained "inside" another, then shaking the container will not make the contained object fall out, whereas if the first object is merely "on" the second, then it will easily fall off. We need to be able to recognise these relationships in a generic way, in scenes with objects we have not been trained on. Within existing robotics work on affordances most seems to focus on single objects, although some of this work does implicitly capture to a relationship, e.g. the effect of pushing on a spherical object [2] depends on a relationship between that object and the object it is resting on (sphere on smooth surface will roll, but not on rough surface). Thus looking at relationships rather than single objects could lead to a more generic framework for affordances.

Of course it would be relatively easy for a programmer to simply code in the required relationship so that a robot would not have to learn it; however we are motivated by ideas of developmental robotics (see [4], [5] for background), and are looking for techniques which would permit a robot to learn relationships which are salient for its own actions, as it needs to. For this reason we do not wish to tailor our system for a predefined set of relationships decided by a human designer; especially we do not want to hardwire the system to use a different set of features for each relationship, where those features are hand designed to be suitable for that particular relationship. Instead we want to make available one set of features which seem to be useful for multiple relationships, and then use a classifier to learn particular relationships with the generic features as input. With generic features we strengthen the possibility that the system could then learn relationships which might not have been envisaged by the human designer (although that is not explored in this paper). To this end we have looked at a number of ways of constructing object-relation data from our vision system, in order to find feature vectors which are good for multiple relations.

We consider three physical relationships: The objects may be *on-top* of, or *inside* each other, or in a position such that pulling one will cause it to contact the other and also bring it closer, as in the use of a rake; we call this *rakeable*. In addition we have negative examples of all of these. We train the system from a large set of scenes of pairs of objects in randomly generated positions, and then test its classification accuracy on novel positions and some novel objects. We have tackled part of this problem in previous work [4]. Here we report on recent improved results using a novel application of histograms to visually recognise a spatial relation between objects in the environment. Using this histogram based approach we are able to report a very high rate of success when the system is asked to recognise a spatial relation.

II. Related Work

Work on learning "affordances" is quite close to ours; Ugur et al. [2] learns affordance predictors for behaviours by learning the mapping from the object features to discovered object effect categories. These predictors can then be used by

an agent to make plans to achieve desired goals. Apart from the fact that we use pairs of objects rather than single objects, this work is quite similar to ours in that essentially it boils down to classification; i.e. once effect categories have been clustered Ugur et al. use a classifier to learn the mapping from the initial object features to these effects. They use SVMs where we use random forests.

In more recent work Ugur et al. [6] use an approach somewhat close to ours in that they look at parts of objects, e.g. to recognise a handle. Identifying a part of an object based on its relationship with the main object is somewhat akin to considering it as two objects in a relationship. Ugur et al. [6] also compile histograms from low level visual features. Our histogram approach is quite different however, as it is the extension of an idea in a different work; our work is heavily inspired by the approach of Mustafa et al. [7]. That work compiles histograms over relationships between surface patches (distances and angles) in a single object. These histograms characterise the object, and are quite robust to variations in viewpoint. Mustafa et al. use this for object recognition. In our work we borrow the idea of compiling histograms over relationships among surface patches; however we look at pairs of objects, and compile histograms which relate every patch on the first object with every patch on the second. Our idea is that these histograms should characterise the relationship between the objects.

We do not feel that our work is particularly close to computer vision work in scene understanding (e.g. [8]) because those works typically recognise all objects, and then can use higher level knowledge to assist in understanding. Our work in contrast is at a lower level, and is more concerned with the physical relationships among surfaces without regard for object knowledge. We think of it more like how an infant might recognise simple physical relationships between household objects without any idea of what their names are or what their typical purposes are.

The most closely related work on learning spatial relations between objects in a 3D space is [9] who use a support vector machine based approach. In this approach the support vectors are picked from for their ability to differentiate the point cloud into two objects. This has the effect that the subset of points considered by the classifier are on the edges of the object. Relations are then learnt based upon the relative positions of clusters of the support vectors.

For any classification based approach to be successful, it requires that similar relations have a similar representation; at the level of point clouds/texlets the representation of a relation can be very different. In the case of [9], the scene is reduced to clusters with xyz coordinates. We feel that our histogram based approach allows for a more generic representation of the scene — we maintain a higher proportion of the important information about the relations between objects.

We can also relate our work 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, at any particular age, they possess a repertoire of behaviours or manual skills which they apply to various objects or surfaces they encounter [10], [11]. Each such behaviour could be seen as roughly analogous to a planning operator in Artificial Intelligence, 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 [12]), because it is through learning means-ends behaviours that infants begin to learn about relationships between objects [13]. The precondition must capture the relationship between objects which determines where the behaviour works or does not work. In preconditions the infant is learning new important abstractions over its sensor space. This can change how an infant understands a scene because the infant can begin to see things at a higher level of abstraction, seeing precisely those relationships which are important in determining what object manipulations are possible (by itself or other agents).

III. Method

A. Overview

Learning a spatial relationship from raw sensor data is challenging as it may take many thousands of examples to train an effective classifier. For this reason we first perform an abstraction using histograms to convert data into a form which simplifies the learning procedure.

After suitable abstractions have been found, to learn classification models for the different *Spatial Relations* we use the Random Forest [14] algorithm. Random Forests (RF) are particularly well suited for our use case, as they inherently do feature selection and hence identify the relevant features from the large amount of input variables.

B. Data Collection

In this work, for learning spatial relations between objects, we collected data using a physically realistic simulation environment [15] designed for Robot Simulations and a sophisticated vision system using a simulated Kinect camera (See III-C).

The objects used range from simple to complex household items and incorporate *supporting* objects like trays, *container* objects like soup bowls and other objects like plastic toys. Figure 1 depicts the objects used in the experiments.

Using the Simulator, we (randomly) placed pairs of objects on a surface and labeled their spatial Relation, before extracting a relational histogram based representation of the scene using the Early Cognitive Vision (ECV) system (See III-C).

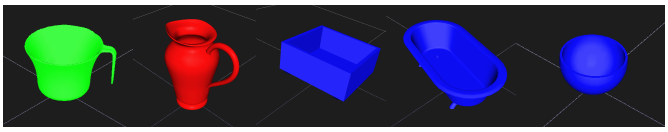

Fig. 1a. Container Objects

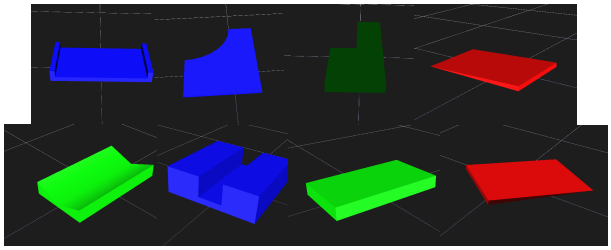

Fig. 1b. Support Objects

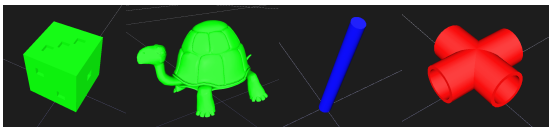

Fig. 1c. Other Objects

Fig. 1: Overview of all Objects used in this work

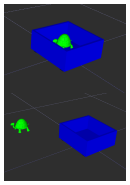

Fig. 2a. Inside / not inside

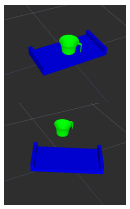

Fig. 2b. On top / not on top

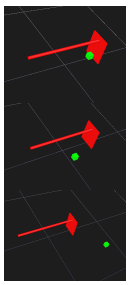

Fig. 2c. Rakeable / almost rakeable / not rakeable

Fig. 2: Overview of the three relationships used in this work. 2a illustrates the "Inside" case with a turtle toy either inside a box or not, 2b similarly illustrates the "On-top" case and 2c depicts from top to bottom Rake catching a die, Rake close to catching a die and Rake not near the die.

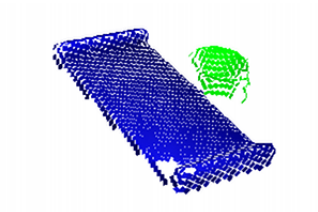

Fig. 3: Texlet representation of a scene. Note that some texlets (surface patches) are missing where surfaces appeared too bright.

Figure 2 illustrates different *Spatial Relations* we focussed on in this work.

For collecting the data, using the RobWork simulator, we randomly distributed one object on a workspace using a normal distribution around the centre of the camera view. For the second object we followed different strategies. This was necessary to promote the chance of certain Relations amongst the objects as with complete uncorrelated random positioning the likelihood of "Inside" and "On-top" scenarios occurring by uninformed random placement is too low to make a reasonable sized training set. In order to promote "desired" cases to collect the necessary data, we placed the second object around the first object with a normal distribution and a smaller variance. In a few cases we limited the possible positions of the second object to "Inside" or "On-top" relations. The orientation of objects in the 3D space was uniformly distributed over roll, pitch and yaw, with the only limitation to ensure *Support, Container* objects stand upright. The *Rakes* had a further restriction to be always in the same orientation perpendicular to the camera, to ensure the second object is visible to the camera and not obscured by the rake itself. Other objects had no limitations to their orientations.

C. Vision System

In our work we use a kinect based vision system called *Early Cognitive Vision* (ECV) system [16], [17].

1) Visual Representation: Using the picture of the scene and the depth map, both provided by the simulated *Kinect* [18] camera, our vision system calculates a 3D point cloud as it is common amongst state of the art vision systems [19].

Based on this 3D point cloud and the colour information of the scene, our vision system creates surface patches as shown in Figure 3. There are different layers of surface patches. We only use the basic layer with surface patches which we call texlets. These texlets describe the surface of the scene with additional information, e.g. not only position in the space, but also the orientation and colour of the surface [20].

In our simulator, we also simulate the noise of real Kinect devices. This gives us data about the depth to the objects in our 3D scene just as we would have obtained from a real Kinect looking at a real scene with 3D objects. The data from the simulated vision system is hence more noisy and less accurate than the perfectly accurate data which could be provided the simulator. The noise affects the position and orientation of the texlets. In addition some texlets can be missing, dependent on the placement of light sources in the simulator, as happens in a real scene; see Figure 3.

2) Object Segmentation: We acknowledge that highly sophisticated Object Segmentation algorithms exist and we assume they could be employed to work in a more complex environment. In this work, however, we used a trivial method for Object Segmentation. The method we present here is based on colour information of the Texlet "Cloud".

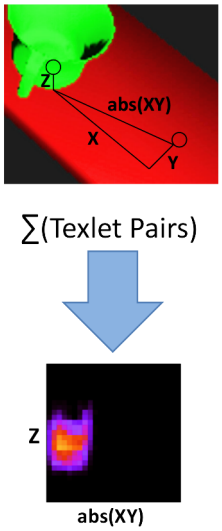

Fig. 4: Illustration of the Histogram creation process from texlets to histograms.

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 very strong assumption, but it could be relaxed by using more sophisticated segmentation methods, which could take into consideration factors like discontinuities of surface curvatures and colour differences. We simplified our segmentation problem because in this work we wanted to focus on finding which type data from the pairs of objects would be most useful for characterising relationships, without our results being affected by errors in segmentation. Future work could look at how the segmentation problem interacts with the problem of recognising spatial relationships.

We coloured our objects either Red, Blue or Green and the background was Grey. The Texlets where then grouped based on their colour or where neglected if they were Grey (or any colour other than Red, Green or Blue).

After segmentation, each object is assigned its unique set of texlets. A texlet representation of a scene with two segmented objects can be seen in Figure 3.

3) Relational-Histogram Creation: Using the segmented texlet based scene representation, we 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 texlets 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 2 texlet groups Π^1 and Π^2 representing the segmented objects 1 and 2 in the scene. For each cross object texlet pair of the form $\Pi_i^1 \oplus \Pi_j^2$ we calculate four *Euclidean Distances* $R_d(\Pi_i^1, \Pi_j^2)$ (The Euclidean Distances along the X, Y and Z axes respectively and in the XY plane) and three *Angle Relations* $R_a(\Pi_i^1, \Pi_j^2)$ (The line through the two texlets 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, Z and Z respectively).

The size of these feature vectors, describing the relation between the two objects in the scene, is variable and determined by the amount of texlets extracted by the vision system. As we want to apply *Supervised Learning Algorithms*, we need the input vector to be generic and of fixed length, for all possible scenarios.

For this, instead of using the data vectors $R_d(\Pi_i^1, \Pi_j^2)$ and $R_a(\Pi_i^1, \Pi_j^2)$ directly, we compute 1-, 2- or 3 Dimensional *"Relational" Histograms* from the data vectors and use these as learning data input, similar to Mustafa et al. [7].

4) Histogram Types: In this work we experimented with 4 different kinds of Histograms for learning *Spatial Relations*. Two 1D composites of Relational Histograms, one 2D and one 3D Histogram. The two 1D composites of Histograms are combinations of 1D Histograms.

- 1D Histograms capture simple relational features between inter-object texlet pairs. For the first 1D composite Relational Histogram, we calculate 3 1D histograms capturing the distances between texlets along each of the 3 main axes X, Y and Z respectively and put them together as 1D learning input. For the second 1D composite Relational Histogram we compute the angle relations in the 3 planes in the space opened by the 3 main axes (XY, XZ and YZ planes) and calculate the angles between texlets in these planes as described above. These angle relations put together alongside the 3 distance histograms, make up the second 1D Relational Histogram.
- The 2D Histogram used in this work, captures the absolute distance of inter-object texlet pairs in the XY plane and puts it into relation with the height difference of the two texlets (i.e. Z difference). In Figure 4 a plain 2D histogram is illustrated.
- The 3D Relational Histogram captures distances between texlets amongst 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 amongst all three main axes (X, Y and Z). 3D Histograms have not been graphically illustrated in this paper mainly because they did not give particularly good results, so it was less interesting to inspect them visually.

In Figure 5 we show examples of all 1D and 2D histograms. 3D histograms have not been illustrated here, but the results are presented in Sec. IV. Note that the first (leftmost) figure in each row is all that the camera sees, so that sometimes some part of an object might be missing if it is very close, or sometimes objects might be far away. Note also that texlets also come only from seen parts; this explains why the first 1D angle histogram shows a peak at both 0 and 360 degrees; it is because the parts of the pitcher close to the camera dominate (the vector from object to camera is 0 or 360 degrees; from the object to the right is 90 degrees, and 180 points to the back). Thus even though the pitcher totally surrounds the rod, the system does not see most of the texlets surrounding it. Note that our system is different to

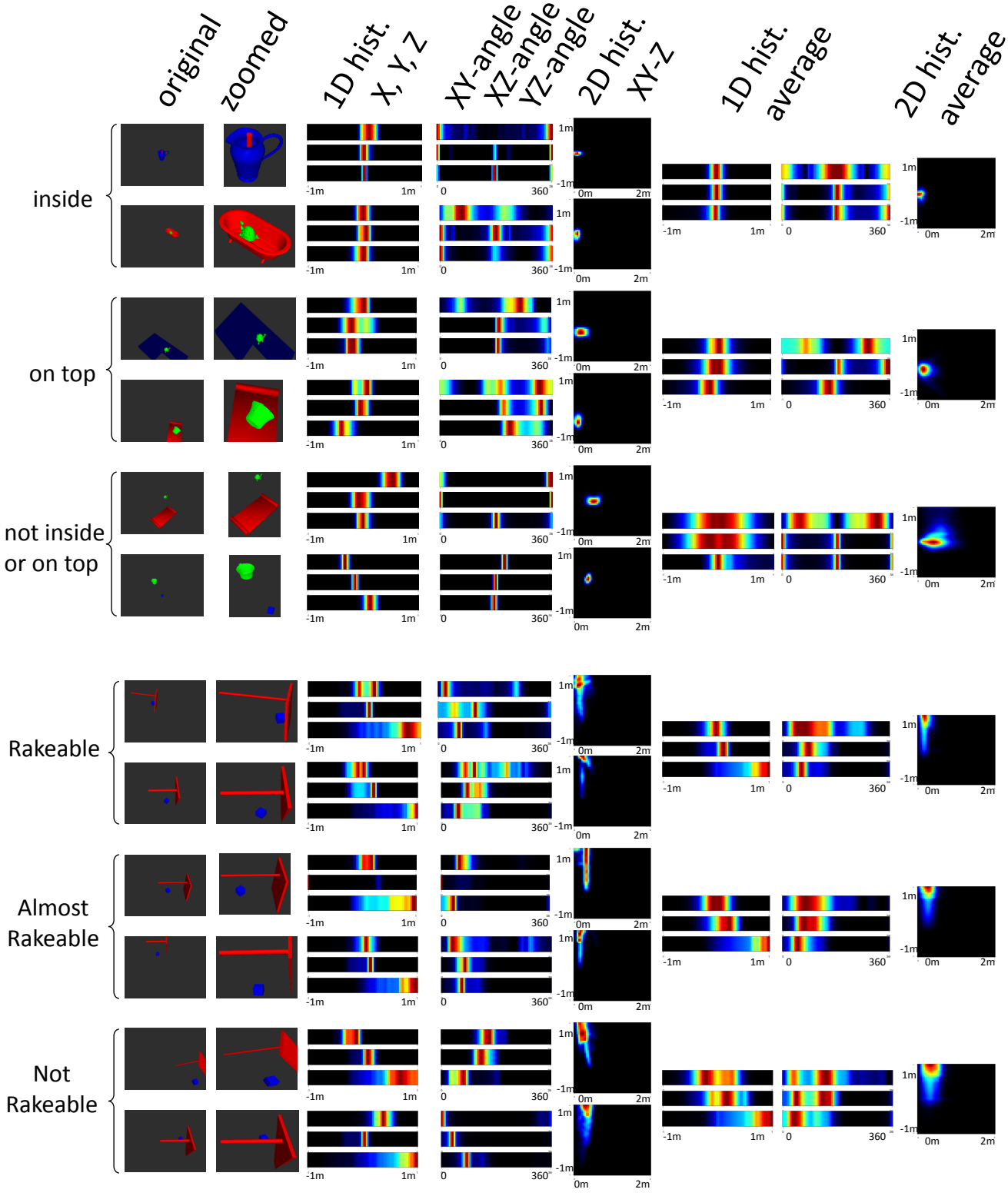

Fig. 5: This Figure is provided to illustrate comparatively how well 1D and 2D histograms can discriminate between the various relationships we have experimented with. For each row of this figure, from left to right we have: an illustration of a scene with a pair of objects, the 1D histograms of that scene, the 2D histograms of that scene, the 1D average histogram across all training samples, the 2D average histogram across all training samples. The Histogram colour code is as follows: Black -> the according distances/angles were not in the data. Red -> most frequent distances/angles.

Mustafa et al. [7] in this respect because they use 3 cameras, surrounding the object.

D. Data & Histogram Post Processing

Mustafa et al. [7] have demonstrated the potential of histograms for object recognition. However, we found the lack of generalisation capabilities of Random Forests to be a limitation in their applicability when it comes to learning Spatial Relations, as it is of major importance to be able, to not only recognise relations between known objects, but also for never before seen objects. In this we differ from Mustafa et al. who only want to recognise well known objects. Therefore in order to not only increase the learning performance, but especially increase the robustness of recognising spatial relations among novel objects, we implemented some feature vector and histogram post processing methods.

The efficiency of these post processing methods on the spatial recognition rate and robustness was investigated in prior work [21]. Given the clear improvements shown there, we use these methods for all learning in this paper.

1) Histogram Normalisation:
Histogram Normalisation proved to vastly increase the robustness of the recognition rate when it comes to novel object pairs and their relations [21]. This is not surprising as the numbers in the un-normalised histograms rely heavily on the sizes of the objects, and the amount of texlets extracted for them by the vision system. Hence, two large objects would generate bigger numbers than two small objects in the same relation. The according histograms would hence look very similar but with different scales. Normalising these histograms removes these scaling effects caused by the sizes of the objects.

E.g. the two imaginary histograms $[1|2|4|1]$ and $[2|4|8|2]$ could describe the same relations for objects pairs with different sized objects. Normalisation would bring both histograms down to $[0.25|0.5|1|0.25]$ and hence remove the differences caused by the object sizes, allowing them to be recognised as the same *Spatial Relation*.

2) Histogram Smoothing:
Histogram Smoothing using normal Gaussian smoothing considering only direct neighbours (i.e. Window size 3) was also found to increase performance, but with a smaller effect on the robustness in case of novel objects [21]. For Smoothing we applied a standard Gaussian Smoothing algorithm with a variance $\sigma^2 = 1$ and a window size of 3 bins, i.e. only direct neighbours to values are taken into account for the smoothing.

Smoothing was found especially useful when used on 2D and 3D Histograms as these are naturally quite sparse, also compared to the according 1D histograms. The smoothing accounts for noise in the histograms caused by kinect camera and the limits in its resolution.

3) Data Scaling:
We apply *Logarithmic Scaling* to the feature vectors preceding the creation of Histograms. This logarithmic scaling had the biggest impact on general classification performance [21] but was only applied on distance features; the angle relation features were not scaled as this would not be sensible.

To scale the data, we replaced the original values of the feature vectors, i.e. distances, with $f(x) = ln(x + 1)$. To compensate for the fact that the logarithm is not defined for negative values, we applied the logarithm on the absolute value and used the negative of the result for originally negative values. Adding 1 to the each absolute values before taking the logarithm ensures that the return value is always positive and the values do not overlap for positive and negative values. This logarithmic scaling has the effect that in the histograms created from the scaled feature vectors, for small distances there is a higher resolution than for larger distances. This has a positive effect because in the smaller distances lies the most useful information about *Spatial Relationships*. It is evident, that if the distance between inter-object texlet pairs is large, the two objects are unlikely to be in a "On-top" or "Inside" relation, but instead are unrelated distributed in the scene.

The Logarithmic Scaling leads to distortion effects of the histograms which makes them less intuitive for human readers. Because of this, in figure 5, for illustration purposes we used post processed histograms without logarithmic scaling. Figure 6 in comparison shows the “Inside” part of figure 5 with complete post processing to illustrate the effects of logarithmic scaling on the Relational Histograms.

IV. Results

To test the performance of the different Histograms and the robustness when it comes to Novel objects we use two different test sets. For every scenario we have a Validation set. This Validation set contains the same object pairs as the Training set, but different instances; i.e. of the overall set of available cases of each object pair setup, some were put into the training set, some others into the Validation set. Furthermore, for the “Inside” and “On-top” scenarios, we also have kept some object pairs out of the Training and Validation sets, to test the performance on not before seen object pairs, to verify the robustness of the Histograms when it comes to novel objects.

For the “Inside” and “On-top” Relations, we considered “Inside” as a subgroup of “On-top”. Any “Inside” case is hence also considered to be “On-top” but not necessarily the other way round.

For the “Inside” Relation we have 9738, 11799 and 5381 instances in the Training, Validation and Novel sets respectively. For the “On-top” Relation we have 11348, 14551 and 5616 instances in the Training, Validation and Novel sets respectively, on-top of the “Inside” instances. We also have 11008, 12943 and 5621 instances in the respective sets of the relationship free instances (neither “inside” nor “On-top” samples).

For the Rake Relation sets, we have 1750 and 4087 samples in the Training and Validation sets respectively for each of the 3 classes “Rakeable”, “Almost Rakeable” and “Not Rakeable”.

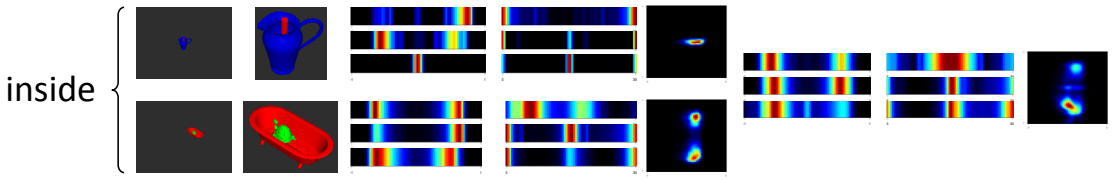

Fig. 6: Logarithmic scaled relational histograms. Compare to Figure 5

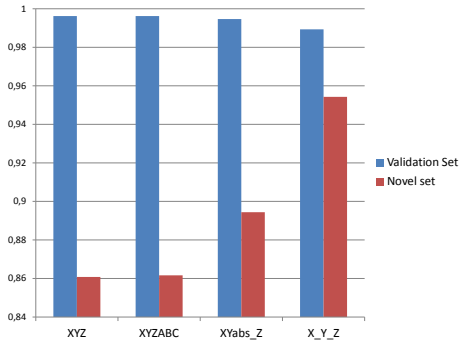

Fig. 7: Performance of "Inside" classifier on the Validation and the Novel test sets.

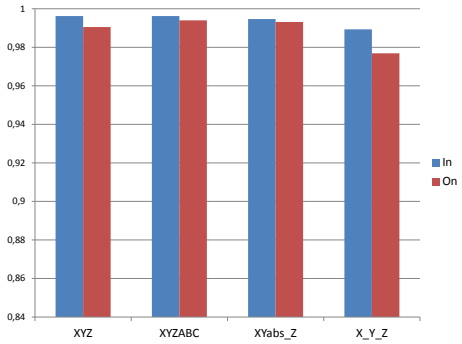

Fig. 9: Comparison of the the "Inside" and "On-top" classifiers performance on the Validation set.

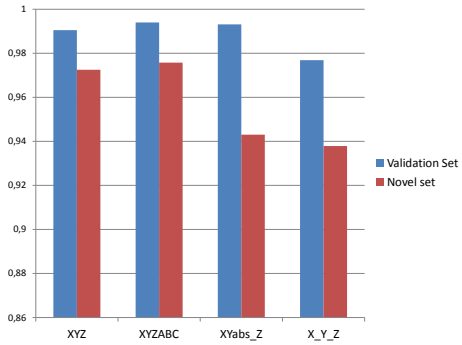

Fig. 8: Performance of "On-top" classifier on the Validation and the Novel test sets.

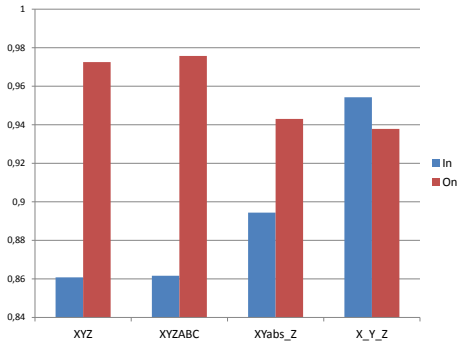

Fig. 10: Comparison of the the "Inside" and "On-top" classifiers performance on the Novel set.

In the following we call the trained classifiers *XYZ* for the classifier based on the 1D Histograms without angle information. *XYZABC* identifies the 1D Histogram based classifier with the angle features. The 2D Histogram based classifier we call *XYabs_Z* and the 3D Histogram *X_Y_Z*.

For each Relationship we trained four individual Binary classifiers, one per Histogram type. Each Binary Classifier was trained and tested 10 times on the Validation and Novel sets where applicable. The results presented here are always the averages of these 10 runs.

A. Inside and On-top Relations

In Figure 7 we show the performance of the four different classifiers for the "Inside" case. The Blue bars show the performance on the Validation set, the red bars show the performance on the Novel set.

Figure 8 shows the same graph as Figure 7, but for the "On-top" case.

Figures 9 and 10 directly compare the four classifiers of "Inside" and "On-top" on the "Validation" and "Novel" test sets respectively.

B. Rake Relations

Figure 11 compares the classifier performances of the different Relations and of the different classifiers on each Relation at the same time.

V. Discussion

Overall the best histograms for classification purposes seem to be the 1D histograms including angles. Their advantage is marginal for "inside" or "on top", but more pronounced for the rake. The 3D histogram performs worst, and this is likely due to the sparsity of data in a 3D histogram.

Above we have presented an empirical evaluation of how well our various histograms can characterise a relationship between two objects. We can also do a thought-experiment

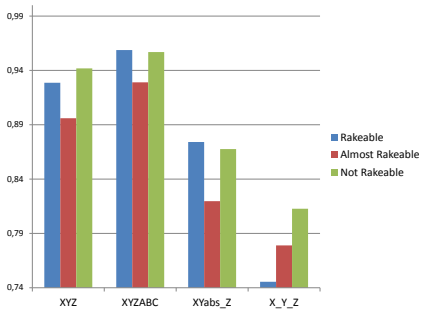

Fig. 11: Comparison of the four classifiers on the Rake relations.

type of analysis, somewhat akin to a mathematical proof where we try to construct a counterexample for our classifiers. E.g. for the relationship "inside" we can contrive objects which would be misclassified by our classifier. For a false positive we can think of how to "fool" the 1D histogram of angles in the XZ plane for example. This histogram has large values for angles which are at the same Z value as the contained object, but offset from it in the $Y-$axis; these are surface patches of the container bounding the contained object. There are two weaknesses: (i) the orientation of the surface patch on the container is not considered — so a sort of louvered surface full of gaps would be admissible; (ii) the Y of the surface patch on the container is not considered, so a large hole at small Y values could be mitigated by surface patches at larger Y values, leading to a container with a missing side being a false positive. Similar examples can be contrived for other histograms. It is harder to contrive a false negative, meaning that we have a "weak" notion of container, because it admits a large set of objects, even with gaps.

Clearly we could upgrade the histograms with a histogram which looks at Y values in conjunction with the XZ angles, however we are wary of constructing features which are specifically tailored to the recognition of one particular relationship ("inside" in this case), because of our desire to allow the system to have generic features so it could learn new relationships which the designer might not have foreseen the need for.

One major weakness of the system is that it makes no effort to guess at unseen parts of objects. This is probably why the results for "inside" are worse than "on top" for novel objects. We suspect that a human recognising relationships such as the stick in the pitcher at the top of Figure 5 would complete absent texlets based on object knowledge and gestalt principles, and so "see" a rod completely surrounded by texlets.

VI. Future Work

In future work we plan to repeat the experiments here with a set of real objects, and real Kinect cameras. We may also experiment with objects which are not simply coloured so that we need to tackle a more realistic segmentation problem. In addition we plan to test our classifiers on the training set of objects used by [9], in order to have a direct comparison which would permit us to consider the relative strengths and weaknesses of the two approaches.

Acknowledgment

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

This work was performed using the Maxwell High Performance Computing Cluster of the University of Aberdeen IT Service (www.abdn.ac.uk/staffnet/research/hpc.php), provided by Dell Inc. and supported by Alces Software.

We thank Nicolas Pugeault for help with his random forests library.

References

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

[2] E. Ugur, E. Oztop, and E. Sahin. Goal emulation and planning in perceptual space using learned affordances. *Robotics and Autonomous Systems*, 59(7–8):580–595, 2011.

[3] Lucas Paletta and Gerald Fritz. Reinforcement learning of predictive features in affordance perception. In Erich Rome, Joachim Hertzberg, and Georg Dorffner, editors, *Towards Affordance-Based Robot Control*, volume 4760 of *Lecture Notes in Computer Science*, pages 77–90. Springer Berlin Heidelberg, 2008.

[4] Severin Fichtl, John Alexander, Dirk Kraft, Jimmy Alison Jorgensen, Norbert Krüger, and Frank Guerin. Learning object relationships which determine the outcome of actions. *Paladyn, Special Issue on Advances in Developmental Robotics*, 3(4):188–199, 2013.

[5] Frank Guerin, Dirk Kraft, and Norbert Krüger. A survey of the ontogeny of tool use: from sensorimotor experience to planning. *IEEE Transactions on Autonomous Mental Development*, 5(1):18–45, 2013.

[6] E. Ugur, H. Celikkanat, E. Sahin, Y. Nagai, and E. Oztop. Learning to grasp with parental scaffolding. In *IEEE Intl. Conf. on Humanoid Robotics, Bled, Slovenia, October*, pages 480–486, 2011.

[7] Wail Mustafa, Nicolas Pugeault, and Norbert Krüger. Multi-View Object Recognition using View-Point Invariant Shape Relations and Appearance Information. In *IEEE International Conference on Robotics and Automation*, 2013.

[8] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[9] Benjamin Rosman and Subramanian Ramamoorthy. Learning spatial relationships between objects. *The International Journal of Robotics Research*, 30(11):1328–1342, 2011.

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

[11] Jeffrey J. Lockman. A perception-action perspective on tool use development. *Child Development*, 71(1):137–144, 2000.

[12] P. Willatts. Development of problem-solving strategies in infancy. In D.F. Bjorklund, editor, *Children's Strategies: Contemporary Views of Cognitive Development*, pages 23–66. Lawrence Erlbaum, 1990.

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

[14] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[15] Jimmy A Joergensen, Lars-Peter Ellekilde, and Henrik G Petersen. RobWorkSim - an Open Simulator for Sensor based Grasping. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–8, June 2010.

[16] Norbert Krüger, Nicolas Pugeault, and Florentin Wörgötter. Visual primitives: local, condensed, semantically rich visual descriptors and their applications in robotics. *International Journal of Humanoid Robotics*, 07(03):379–405, 2010.

[17] Søren Maagaard Olesen, Simon Lyder, Dirk Kraft, Norbert Krüger, and Jeppe Barsøe Jessen. Real-time extraction of surface patches with associated uncertainties by means of Kinect cameras. *Journal of Real-Time Image Processing*, pages 1–14, 2012.

[18] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2):1437–1454, 2012.

[19] R B Rusu and S Cousins. 3D is here: Point Cloud Library (PCL). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4, May 2011.

[20] N. Pugeault, F. Wörgötter, and N. Krüger. Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. *International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision)*, 7(3):379–405, 2010.

[21] Andrew McManus. Learning spatial relationships. Master's thesis, University of Aberdeen, 2013.