Multi-model approach based on 3D functional features for tool affordance learning in robotics

Tanis Mar[1], Vadim Tikhanoff[1], Giorgio Metta[1], Lorenzo Natale[1]

***Abstract*— Tools can afford similar functionality if they share some common geometrical features. Moreover, the effect that can be achieved with a tool depends as much on the action performed as on the way in which it is grasped. In the current paper we present a two step model for learning and predicting tool affordances which specifically tackles these issues. In the first place, we introduce Oriented Multi-Scale Extended Gaussian Image (OMS-EGI), a set of 3D features devised to describe tools in interaction scenarios, able to encapsulate in a general and compact way the geometrical properties of a tool relative to the way in which it is grasped. Then, based on these features, we propose an approach to learn and predict tool affordances in which the robot first discovers the available tool-pose categories of a set of hand-held tools, and then learns a distinct affordance model for each of the discovered tool-pose categories. Results show that the combination of OMS-EGI 3D features and multi-model affordance learning approach is able to produce quite accurate predictions of the effect that an action performed with a tool grasped on a particular way will have, even for unseen tools or grasp configurations.**

I. Introduction and Related Work

Affordances were defined by J.J. Gibson in the late 70's within the context of psychology as the action possibilities of an entity in the environment available to an agent [1]. However, this definition and its further formalizations, most notably the one describing affordances as the relationship between an action (executed by an agent), an object (in the environment) and the observed effect [2], have had a remarkable influence in the field of robotics. The reason for this is that such formalization provides an effective framework for robots to learn the effects of their actions through interaction with the environment, a critical aspect in developmental robotics.

Since the early study in that line carried out by Fitzpatrick et al. [3], many groups have proposed different approaches to learn object affordances. Montesano et al. proposed a model in which Bayesian Networks are applied to learn the relationships among the terms of an affordances as conditional probabilities [4]. Other groups have used general purpose classifiers such as SVMs or K-NN in order to map between the different elements of affordances [5], [6], although more neurally plausible approaches such as Hebbian learning have also been tried [7]. Common among many of these studies has been the the application of clustering methods to the available objects [8], [9], effects [5], [10], or both [7], [6], as a way to ease learning and enable better generalization.

A smaller number of authors have also tackled the problem of robotic tool affordances, where an intermediate object mediates the interaction between the robot and the target object. An approach applied in the first studies in the topic was to learn the affordances of a set of labeled tools [11], [12]. While this approach does not allow to predict the affordances of previously unseen tools, it has proven quite successful in simple scenarios where the robot could experiment with all available tools [13].

More recently, a few studies have been published where descriptors, referred to as *functional features*, were applied to represent the tool, with the aim to match the tool's characteristics expressed by such features with its possible affordances. In [14], they focused on the features of the tooltip, as it is this part of the tool that commonly determines the interaction with target objects. On [15], they studied how certain tool contour features matched particular targets (the way the cross in a screwdriver matches the cross in a screw) and hence afford their interaction, albeit in a very fixed behavior scenario.

In [16] and [17], Jain & Inamura included for the first time functional features in an affordance learning framework, by merging them with the robot's available actions into a new term named Manipulator Pairs among which functionally equivalent sets were inferred. Following their work, Goncalves et al. applied simple geometrical features to describe both the tool and the target object [18], while Mar et al. proposed a more general set of shape features in an approach where the grasp configuration of the handled tool was also considered [19]. The recent work by Myers et al. [20] applied for the first time functional features from 3D images, but the tool affordances are not learned by interaction but rather determined by a human.

The approach taken in these studies is also in line with a growing body of evidence from neuroscientific studies which suggests that primates and humans also perceive objects and tools in terms of their affordances, rather than only their category [21], [22], [23]. The recent experiments carried out by Natraj et al. [24] even argue that the context and hand posture modulate tool-object perception in the brain.

Yet, a question still unanswered both in neuroscience and robotics is whether the set of functional features should represent a particular tool in a particular grasp configuration or an abstracted/canonical version of the tool, on which a particular action and grasp could be applied later. In the current paper we present a comparative evaluation of the

[1] T. Mar, V. Tikhanoff, G. Metta and L. Natale are with the iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, 16163, Genova, Italy (email: tanis.mar@iit.it,vadim.tikhanoff@iit.it, giorgio.metta@iit.it and lorenzo.natale@iit.it).

This work was supported by the European FP7 ICT project No. 270273 (Xperience), and project No. 288382 (POETICON++)

two alternatives, based on the proposed Oriented Multi-Scale Extended Gaussian Image (OMS-EGI) descriptors. OMS-EGI provide a representation of the 3D geometry of a tool based on a concatenation of voxel-wise normal histograms, which is dependent on how the tool is grasped, and hence specifically suited to describe tools in robotic interaction scenarios.

Furthermore, we argue that in order to ease learning and enable more precise affordance predictions for tool use scenarios, instead of learning a single model that tries to relate all the possible variables in an affordance, the robot should learn a separate affordance model for each set of tool-poses sharing common functionality. This is akin to the functionally equivalent sets of Manipulator Pairs proposed by Jain & Inamura [16], but in this case the equivalence is not found among pairs of action-tool features, but rather among sets of 3D geometrical features, so that the consequences of different actions can be learned separately for each of these equivalent sets. In particular, we apply these models to compare the affordances of a large set of handled tools in different grasp configurations, based on of a set of drag actions on a fixed target object.

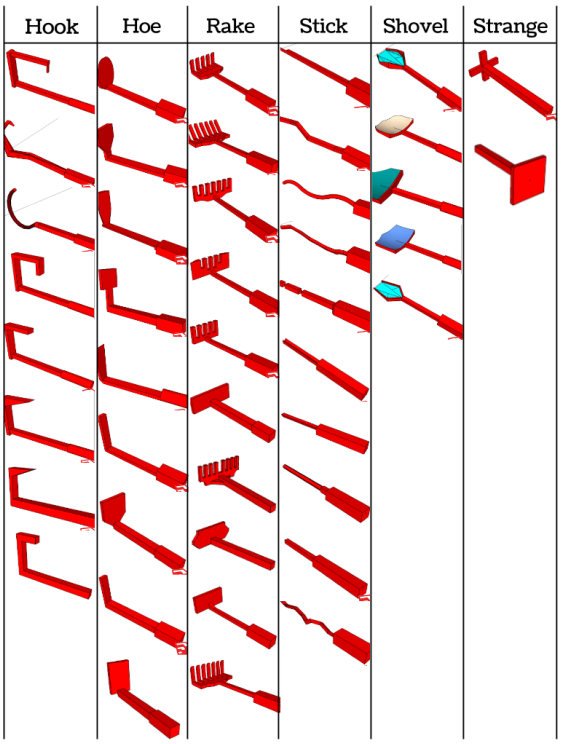


Fig. 1: Tool 3D models used in the current study.

II. Materials and Methods

A. Experimental setup

The experiments carried out in the present study were performed using the iCub simulator. The software controlling the iCub is based on YARP middleware, which enables functionally independent executables (modules) to exchange information in order to achieve the desired behaviors on the iCub [25]. Low level motor control and stability are achieved using available general purpose YARP and iCub modules and libraries [26], while modules for action execution, sensory processing and experimental flow control were implemented specifically for this study. Processing of 3D models and feature extraction was implemented using the Point Cloud Library [27], while experimental data analysis including learning and visualization was implemented in MATLAB, relying on the third party SOM_Toolbox [28] for Self-Organized Map analysis and on the built-in Neural Network Toolbox for regression analysis.

As for the experiment itself, we have used 44 different virtual tools which roughly correspond to 6 different categories, as can be observed in Figure 1. With these tools, each grasped on a few different configurations, the iCub robot performs a series of dragging actions in 8 possible directions on a target object and observes the effects in terms of displacement of the object. The target object is a cube whose initial position before each drag action is fixed at 40 cm in front of the iCub and 10 cm to the right side of the robot's sagittal plane on a virtual table of known height, so that the iCub will always use the right arm holding the tool to perform the action.

B. Oriented Multi-Scale Extended Gaussian Image for tool representation in interaction scenarios

A few studies on tool use learning in robotics have applied functional features in order to represent tool in a way that would help predict their affordances [17], [18], [19]. However, to the best of our knowledge, all studies tackling tool affordance learning from interaction with the environment apply only 2D information. And yet, real world objects functionality depends on, and in many cases emerges directly from, their 3D geometry. Therefore we argue that moving from 2D to 3D features to describe tools in affordance studies is a desirable step. On the one hand, doing so will spare us many of the most common drawbacks of 2D image analysis, such as the objects' representation dependence on perspective or occlusion induced errors. On the other, it also provides much richer information about the actual functionality of tools and objects.

Within the fields of robotics and computer vision, 3D features are mainly applied for object retrieval or recognition/classification [29], [30], [31], [32], [33]. Accordingly, they are usually designed to be similar for similar objects, and also, as opposite of desired in interaction scenarios, independent of the object's pose. Therefore, we introduce **Oriented Multi-Scale Extended Gaussian Image** (OMS-EGI), a set of features devised to describe grasped objects (tools) in interaction scenarios. OMS-EGI are able to encapsulate in a general and compact way the geometrical properties of a tool on a particular grasp configuration (which we refer to as a *tool-pose*, borrowing the term from [34]).

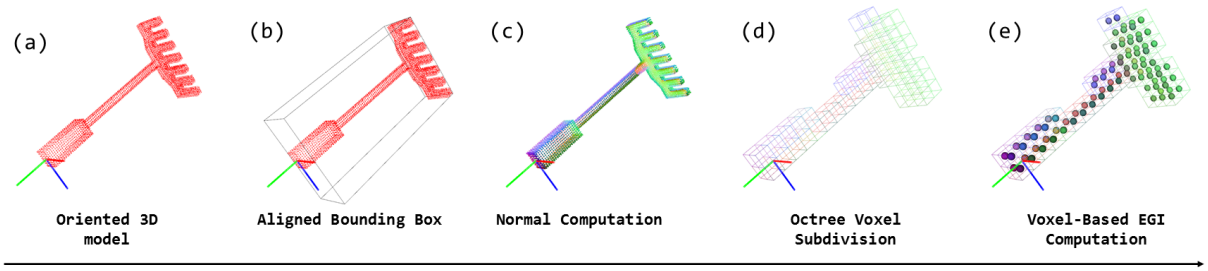


Fig. 2: OMS-EGI Computation Steps: Starting with a 3D model oriented w.r.t the hand reference frame according to the way in which it is grasped (a), its axis-aligned bounding box (AABB) is computed (b), as well as its normals (c). On the next step, the volume enclosed by the AABB is iteratively divided into voxels of different resolutions (d), and for each of them the histogram of normal orientations is computed (e). Finally, all histograms are concatenated in order to build the OMS-EGI feature vector (not shown). For visualization purposes, only one resolution scale is displayed. Also, normal values and normal histograms are represented by colors by mapping XYZ values to RGB values, and XYZ histograms to RGB histograms and averaging over color space.

Figure 2 shows the steps for the computation of the OMS-EGI features, where XYZ histograms have been mapped to RGB to ease visualization.

The OMS-EGI descriptor is a modification and extension of the Extended Gaussian Image (EGI), proposed by Horn in 1984 [35]. The original EGI represents a 3D model as a histogram on the spherical space of the normal orientations of the model, weighted by the area of the faces with such normals. The most important variation in the OMS-EGI with respect to the original EGI formulation is that instead of taking a single normal sphere histogram representing the whole model, OMS-EGI consists of a concatenation of voxel-based-EGIs computed at different resolution scales. We refer by *voxel-based-EGI* as the EGI computed from the portion of the model enclosed in a particular voxel, whereas *scale* corresponds to the size and number of voxels from which the voxel-based-EGIs are computed.

The other determining aspect of OMS-EGI is that voxels are computed from octree subdivisions of the model's axis-aligned bounding box (AABB) with respect to the robot's hand reference frame. Considering that the tools 3D models are oriented with respect to this according the way in which they are grasped, this characteristic ensures that the information on the OMS-EGI is relative to the current tool-pose. Additionally, it also helps bypass the problem of initial pose estimation, present in all other non-rotationally invariant 3D features. Nevertheless, if an actual canonical or preferred pose exists for the model, its OMS-EGI can naturally be computed in such pose, which would enable further analysis in absence of any particular grasp.

When dealing with pointclouds instead of surface meshes, as is the case in the present study, there are slight approximations to be made with respect to the original EGI formulation. As faces do not exists on pointcloud representations, normals can not be computed from them, but rather estimated from the surrounding point neighborhood support of the point (also called k-neighborhood) [36]. Therefore, the voxel-based-EGIs can not be weighted by the area of the faces. Instead, under the safe assumption that points in the pointcloud are reasonably uniformly distributed along the model's surface, we normalized each voxel based histogram by the number of points (each corresponding to one normal) enclosed by the voxel, in such a way that the sum of all the values of each voxel-based-EGI is 1.

Accordingly, there are two main parameters to control how detailed the OMS-EGI representation of a 3D model is:

- The number of bins per dimension of the normal histogram (nB), which reflects the accuracy with which each voxel-based-EGI will represent the normals contained in its corresponding voxel. Each voxel-based-EGI consists of nB^3 values.
- The number of octree levels explored (D from Depth), starting from level 0, i.e. the whole bounding box. D represents thus the resolution at which the voxel-based EGIs will be computed, by controlling the number and size of these voxels.

The total size of the OMS-EGI feature vector is computed as: $length(OMSEGI) = \sum_{l=0}^{D}(8^l \cdot nB^3)$, where l: octree level.

C. Functional tool-pose clustering

As stated above, the main aim of this paper is to propose a method that tackles the question of how could a robot take advantage of the fact that different tools can afford similar functionality if they share some common geometrical features and are grasped in a similar way, or in other words, that similar tool-poses have similar affordances. We have already introduced OMS-EGI features, which we devised as a means to encapsulate these properties (geometry and grasp) of a handled tool in order to ease the rest of the analysis. In order to make sense of it, though, the robot needs to analyse a large set of tool-poses and find out the eventual commonalities among them, thus discovering the available tool-pose categories. To that end, unsupervised clustering is applied on the OMS-EGI data. The method chosen to do so was Self-Organized Maps based K-Means (SOM K-means),

due to the relative high-dimensionality of the feature vector when compared with the available number of samples, which would cause simple K-means to yield very irregular and unbalanced results.

SOMs provide a lower dimensional representation of the input data based on an iterative method of vector quantization [37], on which K-means can be performed without the issues appearing when applying it directly on the higher dimensional data. Still, K-means is very sensitive to the initialization conditions, and does not provide an automatic way of selecting K. In our study, we select K in function of an *ad hoc* implemented cluster quality index, defined as the combination of the Davies-Bouldin index [38], commonly used to assess cluster separation, and a value to promote clustering trials that led to more balanced clusters (in terms of number of samples per cluster). This last term was computed as the standard deviation of the histogram of the resulting cluster indices (normalized by dividing by maximum), multiplied by a constant that determines its influence over the Davies-Bouldin index, which we set to 2.

D. Tool-pose category dependent affordance models

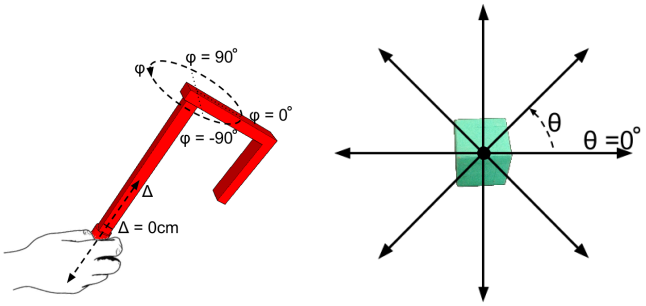
Once the set of available tool-pose categories has been discovered by clustering the OMS-EGI features with the methods described above, the robot should learn what the common affordances of each tool-pose category are. In this study, as we did in [19], we consider affordances as defined by the relationship between the terms of the tuple $\{$*tool, grasp, action, effect*$\}$. The target object is not included, because its affordances are assumed to be learned in a previous stage, after which they could be combined with knowledge about tool affordances in order to model the whole interaction scenario, but this is out of the scope of this study. From the terms we do take in account, the *tool* is represented by the OMS-EGI features described in Section II-B. The other three terms $\{$*grasp, action, effect*$\}$, referred together as *affordance data*, are formalized as described below:

1) Grasp: The tool poses available for the robot in the present study are described by 2 parameters, graphically described in Figure 3a. These control the orientation (φ) and displacement (Δ) of the handled tool with respect to its canonical pose ($\varphi = 0$ and $\Delta = 0$). Here, we consider canonical pose as the one in which the tool's handle axis is oriented along the extended thumb axis, and the tool effector pointing towards the the extended index finger axis (-Y and X axis on the iCub hand reference frame, respectively).

2) Action: The set of actions that the robot can perform in the current experiment was limited to a drag action parametrized by the angle $\theta \in \{0, 360\}$ degrees along which the robot tries to drag the object, as displayed in Figure 3b.

3) Effect: The effect of the robot's tool use was measured as the euclidean distance between the object's position before and after the action execution.

To the best of our knowledge, all previous affordance studies apply a single-model approach, where only one model aims at learning the relationships between the terms in



(a) Grasp parameters. φ controls the rotation around the tool's handle axis, while Δ controls the displacement respective to the hand on the same axis.

(b) Diagram of the drag action. The tooltip is initially placed slightly behind the object, and then the tool is displaced 15 cm along the radial direction given by θ.

Fig. 3: Parameters controlling tool-pose and interaction: (a) Grasp and (b) Action.

the affordance tuple for all their possible values. We propose instead a multi-model approach, where a distinct affordance model is learned for each tool-pose category, following the assumption that tool-poses clustered together share common geometrical properties and hence also similar functionality. In order to train each of those models, the affordance data is divided in as many subsets as tool-pose categories have been discovered, in such a way that each subset contains only affordance data generated by tool-poses belonging to a particular cluster. Then, each affordance model is trained separately with the affordance data subset corresponding to the tool-pose category to which it is associated. An explanatory diagram of the proposed approach can be observed on Figure 4.

Moreover, we wanted to assess whether grasp configuration information should be provided explicitly as an input to the affordance model, or rather implicit as part of the tool representation. Therefore, we used two different training schemas, which differed on the information present on the OMS-EGI feature set used to discover the tool-pose categories, and hence in the number of affordance models initialized and the affordance data that correspond to each one. In the first schema, which we refer to as *Oriented features*, the pose of the 3D models from which the OMS-EGI descriptors were extracted matched the way in which the actual tools were handled in the simulator to interact with the environment. Here, the OMS-EGI features implicitly encode grasp information, and hence the tool-pose category to which each model is associated depends, albeit indirectly, on the grasp parameters. Therefore, the models learned when using Oriented features map directly *action* $\rightarrow$ *effect*, for all given tool-pose categories.

In the second schema, referred to as *Canonical features*, OMS-EGI features were extracted from the 3D models being in their canonical pose, which did not match the pose of the actual tool in the simulator. The grasp information is thus not encoded by the OMS-EGI vector, which is therefore

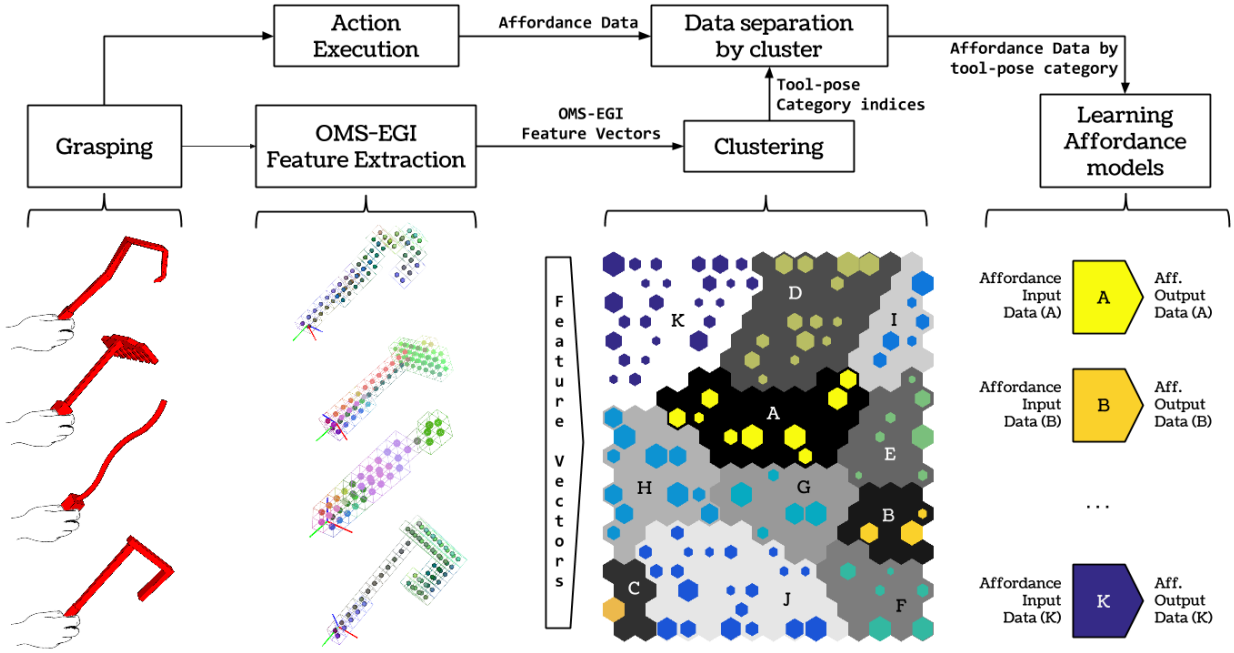


Fig. 4: Diagram of the proposed multi-model approach for tool affordance learning. From left to right: OMS-EGI features are extracted from the tools' 3D models, and subsequently clustered using SOM K-means. Then, recorded affordance data is divided according to the cluster to which the corresponding tool-poses belong, and used to train separate affordance models, so that each of them models the affordances of a particular tool-pose category.

constant for each considered tool, independently of the way it is grasped by the robot. Hence, the grasp parameters of each trial are explicitly fed to each affordance model, which thus performs the mapping $\{grasp,\ action\} \rightarrow effect$, for all given canonical tool-pose categories.

Independently of the feature schema, all affordance models' inputs and outputs are real values, so the learning problem becomes one of regression / function approximation: $\hat{e} = f_{tp}(i)$ where $\hat{e} \in \mathbb{R}$ is the predicted effect, f_{tp} the affordance function to learn for each tool-pose category, and i is the affordance input. For Oriented features, where the only input is the action parameter θ, $i \in \mathbb{R}$. For Canonical features, where the grasp parameters are also fed to the affordance models, $i \in \mathbb{R}^3$. Given that the elements present in the tuple are relatively low dimensional, the regression models do not need to be utterly complex. In the present work, we use generalized regression neural networks (GRNN), a modification of radial basis networks which is able to approximate arbitrary functions and avoid local minima with 1-pass training [39]. This kind of networks have a single hyper-parameter σ which serves a regularization parameter by controlling the spread of its radial basis function. In order to find the optimal σ for each affordance model, we performed recursive line search based on cross-validation results on the training subset. The parameter σ for which the average cross-validation accuracy is highest is used to train the final model for each tool-pose category.

III. Results

A. Experimental data collection and separation

In the current experiment, 44 virtual tools represented by between 1500 and 4500 points (see Figure 1) were used by the iCub in simulation to gather interaction data. Each tool was grasped in 9 different poses, corresponding to the combinations of 3 different grasp orientations ($\varphi = \{-90, 0, 90\}$) and 3 different grasp displacements ($\Delta = \{-2, 0, 2\}$), adding up to a total of 396 tool-poses. For each tool-pose, two OMS-EGI feature vectors were computed: Oriented and Canonical, as described in Section II-D. For each of these tool-poses, the iCub performed the drag action described in Section II-D.2 in 8 directions, corresponding to angles θ from 0 to 315 degrees in intervals of 45 degrees ($\theta = \{0, 45, 90, 135, 225, 270, 315\}$), thus executing a total of 3168 actions.

For each of these actions, all the affordance data values ($\{grasp, action, effect\}$) were recorded, in association with the tool-pose that generated them. Before any further processing, these data were separated in training and tests sets, which remained constant throughout the experiment and the off-line data analysis. 75% of all the tool-poses were selected randomly, and all the data associated with them used for training. For each tool-pose, these data consisted of the Canonical and Oriented OMS-EGI vectors and the affordance data tuples recoded for each of the 8 performed actions. The data corresponding to the remaining 25% of the tool-poses were used for testing. Thus, the training set consisted

of the Oriented and Canonical OMS-EGI vectors of 297 tool-poses, and the 2376 affordance data vectors associated with those tool poses, while the test set was formed by the OMS-EGI vectors of the remaining 99 tool-poses, and their corresponding 792 affordance data vectors.

Furthermore, given that there is no ground truth for the clustering process, and that interaction data is itself very noisy, errors might appear even if both processing steps worked perfectly. Hence, we needed to set a performance baseline against which we could compare the performance of our approach. To that end, we carried out two additional data processing runs, corresponding respectively to the Oriented feature schema and the Canonical features schema, where after performing clustering, all the indices of the category corresponding to each tool-pose were shuffled. Therefore, the affordance data used to train and test the affordance models in these runs were not corresponding anymore to a particular tool-pose category for each model, but rather distributed among them at random. We refer to these data as *Oriented-shuffled* and *Canonical-shuffled*, respectively.

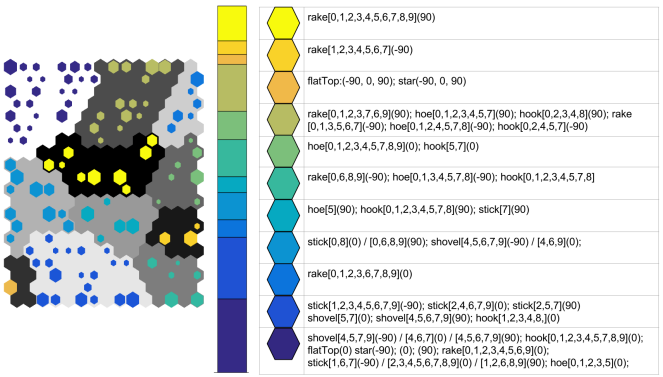
B. Discovery of tool-pose categories

Once the data had been sorted out, the first step in our approach to model tool affordances was to discover the available tool-pose categories by clustering the OMS-EGI features of those belonging to the training data. In this study, we set the parameters of the OMS-EGI algorithm to be $D = 2$ and $nB = 2$, so the total length of the OMS-EGI feature vector is of 576 features. On the analysis of Oriented features, each tool-pose produced a distinct OMS-EGI vector, which thus add up to a total 297 samples clustered. In the case of Canonical features, as the canonical pose was always the same for each tool, independently of the way in which it was actually grasped by the iCub (in simulation), only 33 distinct OMS-EGI vectors were extracted and clustered. The clustering results for both schemas can be observed on Figure 5

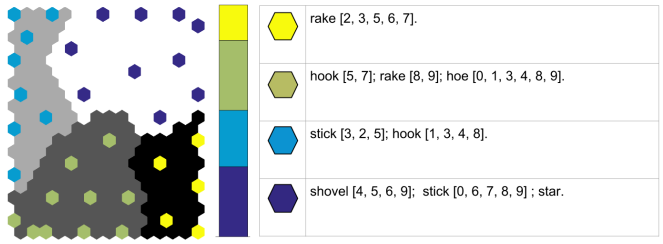
C. Prediction of tool-pose affordances

Through the clustering procedure, 11 and 4 tool-pose categories were discovered for the Oriented and Canonical features schemas, respectively. An equal number of affordance models were trained in each case, each of them with the affordance data generated by the tool-poses in the cluster associated with the affordance model. As described in Section II-D, each affordance model was implemented using a GRNN whose σ parameter was determined by recursive line search based on cross-validation.

Then, in order to evaluate the validity of our approach, we assessed the predicted effect values obtained with the data from the tool-poses belonging to the test set, which had not been used either on the clustering procedure or for training the affordance models. To that end, the first step was to classify the OMS-EGI features of each of the test set tool-poses into the previously discovered categories. This was done by finding the best matching unit (BMU) of each test OMS-EGI feature vector on the trained SOM, and



(a) Oriented features.



(b) Canonical features.

Fig. 5: Clustering results on (a) Oriented features and (b) Canonical features. Left side: cluster distribution on the SOM with superimposed best-matching units of the clustered samples. Center: proportion of samples per cluster. Right: training tool-poses belonging to each cluster, i.e. tool-pose category, represented as `name of tool + [tool indices] + (grasp orientation)`.

determining the cluster to which that BMU belonged. Then, all the affordance input data associated with each of the test tool-poses was fed to the corresponding GRNN affordance model, which produced a prediction of the effect for each of the input data vectors. Finally, we computed the following measures of prediction error between the average predicted effect ($\hat{e}$) and the average recorded effect e for each action of each tool-pose category:

- Mean Absolute Error (MAE): Average of the absolute difference between the recorded and the predicted values:
$$MAE = mean(abs(e - \hat{e}))$$
- Mean Absolute Percentage Error (MAPE): Measure of the absolute error in relation to the value of the recorded effect:
$$MAPE = mean(abs(e - \hat{e})/e)$$

Figure 6 shows the predicted effect for all actions performed with each test tool-pose, compared with the recorded affordance data. For Canonical features, the data was divided according to the kind of grasp that generated it for visualization and evaluation, but the number of trained models remained equal to the number of categories discovered on the clustering process, i.e., 3. Table I, in turn, displays the numeric values of the achieved performances.

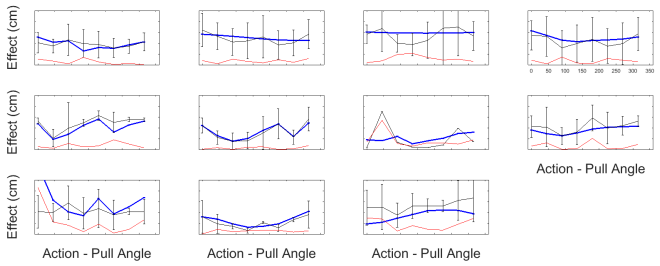
Oriented features	Prediction Error	Avg
MAE (cm)	1.29 1.73 2.82 1.73 1.89 0.67 4.27 1.67 5.78 1.37 4.47	2.06
MAPE (%)	15.12 14.21 19.20 14.34 21.03 8.74 92.49 18.79 30.85 27.40 63.31	21.75
Oriented-shuffled	**Prediction Error**	**Avg**
MAE (cm)	2.32 3.81 4.60 2.95 2.52 3.09 6.40 1.67 1.48 4.80 2.39	2.90
MAPE (%)	19.62 46.83 50.77 36.06 21.88 33.96 69.31 16.18 16.13 50.04 21.26	30.88
Canonical features	**Prediction Error**	**Avg**
MAE (cm)	4.40 2.57 3.54 7.81 0.80 4.92 2.15 1.49 1.41 2.72 2.75 2.73	2.72
MAPE (%)	36.42 20.14 28.99 73.36 7.11 46.24 27.90 18.73 17.59 31.43 34.64 31.71	28.52
Canonical-shuffled	**Prediction Error**	**Avg**
MAE (cm)	2.61 2.93 3.21 10.63 2.03 6.23 2.85 2.65 2.27 3.51 6.78 2.29	3.62
MAPE (%)	25.64 28.77 31.84 122.03 23.27 70.43 31.07 28.05 23.23 29.41 55.78 18.73	37.08

TABLE I: Prediction Error for Oriented and Canonical Features and baseline for comparison. Prediction Error on Oriented features shows the error for each of the trained models (one per tool-pose category). On Canonical features, the results for each of the considered grasp orientations ($\varphi = \{-90, 0, 90\}$) on each model are displayed, corresponding to the results plotted on Figure 6b.

Observing subfigure 6a, we can see that for Oriented features, the affordance models' predictions match the average recorded effect of their corresponding tool-pose categories quite well in most cases, even in some of the categories with larger variance among tool-poses' individual effects. These results mean that the clustering step was successful in discovering and partitioning the oriented tool-poses into functionally similar categories, which in turn means that the OMS-EGI extracted from oriented tool models provided enough information to do so. On subfigure 6b we can notice that the distance between the predictions for Canonical features and the recorded effects are larger than those for Oriented features. This observation is also supported by the numerical results on Table I, which shows that the Oriented features schema enables smaller error on the affordance prediction. It also shows, nevertheless, that in either case the proposed clustering procedure leads to a considerable improvement of the prediction accuracy when compared to the baseline.

IV. Conclusions

In the present paper we have tackled the question of how can robots take advantage of the fact that similar tool-poses have similar affordances. In doing so, we have presented two novel contributions to the field of tool affordance learning in robotics. On the one hand, we introduced OMS-EGI, a set of 3D features devised specifically for tool use scenarios. This set of features encapsulate the geometry of a tool and the way in which it is grasped, and thus, as it has been shown, relate nicely with the functionality of the tool and hence its prediction. We have also determined that these features provide more information about the tools' functionality when extracted from the oriented 3D models than when extracted from a canonical pose, even if combined with explicit grasp



(a) Oriented features.

(b) Canonical features.

Fig. 6: Prediction results on the test data by (a) tool-pose category for Oriented features and (b) considered grasp for each tool-pose category for Canonical features. The blue line represents the average affordance model's prediction for each possible action. The black line represents the average recorded effect for all the tool-poses determined to belong to the corresponding category, where the vertical errorbars represent its standard deviation. The red line represents the absolute error between recorded data and prediction. Effect axis in all graphs spans from 0 to 20 cm, while Action-Pull ranges from 0 360 degrees.

information.

On the other hand, we proposed a multi-model approach where instead of a single model aiming at generalizing all possible cases, a different affordance model is learned for each tool-pose category, where categories are found by clustering the tool poses based on their geometrical properties.

Results show that the combination of OMS-EGI 3D features and multi-model learning approach is able to produce quite accurate predictions of the effect that an action performed with a tool grasped on a particular way will have. Nevertheless, albeit promising, these results also leave plenty of space for refinement and improvement; although the tool representation has certain level of complexity, the possible actions and the way of measuring the tool use effect are admittedly limited. In order to move towards more realistic tool use scenarios, these elements will need to be further developed and studied in conjunction with the rest of the system, and of course, in real robot experiments.

REFERENCES

[1] J. J. Gibson, "The Ecological Approach to the Visual Perception of Pictures," *Leonardo*, vol. 11, no. 3, pp. 227–235, 1978.

[2] A. Chemero, "An Outline of a Theory of Affordances," *Ecological Psychology*, vol. 15, no. 2, pp. 181–195, 2003.

[3] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, G. Sandini, L. Natalet, S. Raot, and G. Sandinit, "Learning About Objects Through Action - Initial Steps Towards Artificial Cognition," in *Procceedings of the IEEE Intrernational Conference on Robotics and Automation*, 2003, pp. 3140–3145.

[4] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Modeling affordances using Bayesian networks," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2007, pp. 4102–4107.

[5] E. Ugur, E. Sahin, and E. Oztop, "Predicting future object states using learned affordances," *2009 24th International Symposium on Computer and Information Sciences*, pp. 415–419, Sep. 2009.

[6] S. Griffith, J. Sinapov, V. Sukhoy, and A. Stoytchev, "A Behavior-Grounded Approach to Forming Object Categories: Separating Containers from Non-Containers," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 1, pp. 54–69, 2012.

[7] B. Ridge, D. Skocaj, and A. Leonardis, "Self-Supervised Cross-Modal Online Learning of Basic Object Affordances for Developmental Robotic Systems," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 5047–5054.

[8] P. Osório, A. Bernardino, and R. Martinez-cantin, "Gaussian Mixture Models for Affordance Learning using Bayesian Networks," in *International Conference on Intelligent Robots and Systems*, 2010, pp. 1–6.

[9] T. Hermans, J. M. Rehg, and A. Bobick, "Affordance Prediction via Learned Object Attributes," in *IEEE International Conference on Robotics and Automation (ICRA 2011). Workshop on Semantic Perception, Mapping, and Exploration*, 2011.

[10] B. Akgun, N. Dag, T. Bilal, I. Atil, and E. Sahin, "Unsupervised learning of affordance relations on a humanoid robot," *2009 24th International Symposium on Computer and Information Sciences*, pp. 254–259, Sep. 2009.

[11] A. Stoytchev, "Robot tool behavior: A developmental approach to autonomous tool use," Ph.D. dissertation, Georgia Institute of Technology, 2007.

[12] J. Sinapov and A. Stoytchev, "Detecting the functional similarities between tools using a hierarchical representation of outcomes," in *2008 7th IEEE International Conference on Development and Learning*. Ieee, Aug. 2008, pp. 91–96.

[13] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta, "Exploring affordances and tool use on the iCub," in *Humanoids*, 2013.

[14] C. C. Kemp and A. Edsinger, "Robot Manipulation of Human Tools: Autonomous Detection and Control of Task Relevant Features," in *IEEE International Conference on Development and Learning*, 2006.

[15] T. E. Horton, "A Partial Contour Similarity-Based Approach to Visual Affordances in Habile Agents." Ph.D. dissertation, 2011.

[16] R. Jain and T. Inamura, "Learning of Tool Affordances for autonomous tool manipulation," *2011 IEEE-SICE International Symposium on System Integration SII*, pp. 814–819, 2011.

[17] ——, "Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools," *Artificial Life and Robotics*, vol. 18, no. 1-2, pp. 95–103, Sep. 2013.

[18] A. Gonçalves, J. a. Abrantes, G. Saponaro, L. Jamone, and A. Bernardino, "Learning Intermediate Object Affordances : Towards the Development of a Tool Concept," in *IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob 2014)*, no. October, 2014, pp. 13–16.

[19] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised learning of grasp dependent tool affordances on the iCub Humanoid robot," in *International Conference on Robotics and Automation*, 2015, pp. 3200 – 3206.

[20] A. Myers, A. Kanazawa, C. Fermuller, and Y. Aloimonos, "Affordance of Object Parts from Geometric Features," in *International Conference on Robotics and Automation2*, 2015, pp. 5–6.

[21] S. T. Grafton, L. Fadiga, M. a. Arbib, and G. Rizzolatti, "Premotor cortex activation during observation and naming of familiar tools." *NeuroImage*, vol. 6, no. 4, pp. 231–236, 1997.

[22] A. M. Proverbio, R. Azzari, and R. Adorni, "Is there a left hemispheric asymmetry for tool affordance processing?" *Neuropsychologia*, vol. 51, no. 13, pp. 2690–2701, 2013.

[23] P. O. Jacquet, V. Chambon, A. M. Borghi, and A. Tessari, "Object affordances tune observers' prior expectations about tool-use behaviors." *PloS one*, vol. 7, no. 6, p. e39629, Jan. 2012.

[24] N. Natraj, V. Poole, J. C. Mizelle, A. Flumini, A. M. Borghi, and L. a. Wheaton, "Context and hand posture modulate the neural dynamics of tool-object perception," *Neuropsychologia*, vol. 51, no. 3, pp. 506–519, 2013.

[25] G. Metta, "Software implementation of the phylogenetic abilities specifically for the iCub & integration in the iCub Cognitive Architecture," Tech. Rep. 004370, 2006.

[26] U. Pattacini, "Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub," Ph.D. dissertation, 2011.

[27] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011.

[28] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-Organizing Map in Matlab: the SOM Toolbox," in *Matlab DSP Conference*, 2000, pp. 35–40.

[29] L. Zhang, M. João, and A. Ferreira, "Survey on 3D shape descriptors," Tech. Rep., 2004.

[30] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441–471, Dec. 2007.

[31] A. Alexandre, "3D Descriptors for Object and Category Recognition : a Comparative Evaluation," in *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[32] F.-w. Qin, L.-y. Li, S.-m. Gao, X.-l. Yang, and X. Chen, "A deep learning approach to the classification of 3D CAD models," *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 2, pp. 91–106, 2014.

[33] Z. Wu and S. Song, "3D ShapeNets : A Deep Representation for Volumetric Shapes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, 2015, pp. 1–9.

[34] S. Brown and C. Sammut, "Tool Use Learning in Robots," in *Advances in Cognitive Systems*, 2011, pp. 58–65.

[35] B. K. P. Horn, "Extended Gaussian Images." *Proceedings of the IEEE*, vol. 72, no. 12, pp. 1671–1686, 1984.

[36] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," *Kuenstliche Intelligenz*, vol. 24, no. 4, pp. 1–4, Aug. 2010.

[37] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[38] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE transactions on pattern analysis and machine intelligenceP*, vol. 1, no. 2, pp. 224–227, 1979.

[39] D. F. Specht, "A general regression neural network," *Neural Networks, IEEE Transactions on*, vol. 2, no. 6, pp. 568–576, 1991.