Action-Grounded Surface Geometry and Volumetric Shape Feature Representations for Object Affordance Prediction

Barry Ridge and Aleš Ude†

***Abstract*— Many 3D feature descriptors have been developed over the years to solve problems that require the representation of object shape, e. g. object recognition or pose estimation, but comparatively few have been developed specifically to tackle the problem of object affordance learning, a domain where the interaction between action parameters and sensory features play a crucial role. In previous work, we introduced a feature descriptor that divided an object point cloud into coarse-grained cells, derived simple features from each of the cells, and grounded those features with respect to a reference frame defined by a pushing action. We also compared this action-grounded descriptor to an equivalent non-action-grounded descriptor coupled with action features in a push affordance classification task and established that the action-grounded encoding can provide improved performance. In this paper, we investigate modifying more well-established 3D shape descriptors based on surface geometry, in particular the Viewpoint Feature Histogram (VFH), such that they are action-grounded in a similar manner, compare them to volumetric octree-based representations, and conclude that having multi-scaled representations in which parts at each scale can be referenced with respect to each other may be a crucial component in action-grounded affordance learning.**

I. INTRODUCTION

The task of learning to predict object affordances with a robotic system is a significant one and, despite the comparative ease of affordance learning in humans, belies subtle challenges at the intersection between action and perception that have yet to be fully deciphered. Gibson observes that *"One may consider the layout of surrounding surfaces with reference to a stationary point of observation, . . . Or one may consider the layout of surrounding surfaces with reference to a moving point of observation along a path that any individual can travel. This is much the more useful way of considering the surroundings, and it recognizes the fact that animals do in fact move about. The animal that does not move is asleep – or dead."* [1]. In this paper, we consider the idea that a dynamic frame of reference based on possible actions, when describing object surface geometries with feature descriptors, may be an important discriminative tool when predicting their respective affordances. We refer to features defined in this way as *action-grounded features*.

As Stoytchev notes, *"Grounding is a familiar problem in AI. . . . Grounding, however, is also a very loaded term. Unfortunately, it is difficult to come up with another term to replace it with."* [2]. What we loosely mean here by action-grounded features are features that are defined, in some important way, within the context of an action, or in other words are variant with respect to the action. More concretely, in our particular scenario, this means features that are derived within a reference frame formed by the contact point and push direction vector derived from a push action trajectory acting on an object. As different pushes from different directions and with different contact points are applied to the object, the reference frame will change, and so too will the features that are derived thereof.

While our previous work provided some tentative results demonstrating the potential of action-grounded features in an object push affordance bootstrapping context [3], and in comparison with non-action-grounded features [4], the 3D shape features that were used were relatively simplistic compared to the raft of state-of-the-art 3D descriptors currently in common usage. This study aims at integrating the crucial aspects of our past efforts with a cross-comparison between some of those state-of-the-art 3D descriptors re-purposed to work in an action-grounded setting where possible.

One of the reasons that this can be achieved is because many of these feature descriptors, having originally been designed for pose estimation, contain a viewpoint component, whereby the content of the descriptor varies depending on the position of the viewpoint relative to the object. The viewpoint, and the view direction vector, can function as effective analogues to the contact point on the surface of an object given a pushing action, and the push direction vector respectively. That is precisely how we exploit this component in order to motivate the design of the first of our proposed feature descriptor contributions which is based on describing object surface geometry using angular histograms. The second of our proposed contributions is based on multi-scale octree subdivision of object point clouds and providing geometrical feature descriptors within each of the subdivisions. The subdivisions, and their respective features, are defined relative to a reference frame defined by the push contact point and direction.

In the following we first review the literature on action-grounded features in robotic affordance learning, as well as both classical and more modern 3D object feature representations. In Section II we introduce the first of our proposed action-grounded feature descriptors, and in Section III we describe the second one. Section IV describes our experiments and results. Finally, in Section V we offer concluding thoughts and plans for future work.

†B. Ridge and A. Ude are with the Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. barry.ridge at ijs.si

A. Related Work

Object push affordance learning is an area of robotic learning that has seen quite a number of approaches to its study by various authors [5], [6], [7], [8], [9], [10], [11] since the classic example by Fitzpatrick *et al.* [12], often with different feature representations being used in each case. Angular histograms have been used as object features by Ugur *et al.* [7], [8] in order to describe the shapes of objects in both affordance traversability studies [7], where a robot must push its way past certain obstacles in a mobile environment, and in manipulation studies [8], where a robot must learn push affordances of objects on a table surface.

The idea of grounding features with respect to a local reference frame defined by actions like pushing or grasping has been exploited also by other authors. For instance, Hermans *et al.* [13] used a feature descriptor that takes a segmented object point cloud as input and computes both local and global shape descriptors from the 2D projection of the point cloud onto the supporting table surface encoded in a coordinate frame defined by the object center and a chosen pushing contact location. A notable difference in our work presented here is marked by various efforts to extend such an idea into three dimensions.

Mar *et al.* [14], with their Oriented Multi-Scale Extended Gaussian Image (OMS-EGI) descriptor, described a scenario where angular histograms are computed from octree subdivisions of the axis-aligned bounding boxes (AABB) of tool models with respect to the hand reference frame of an iCub robot, which vary depending on how the tools are grasped. This work is similar to both our previous work [3], [4], and the octree-based feature descriptor we propose in this paper (cf. Sec. III) in the sense in which objects are divided into parts, but our work differs in a number of key respects. Firstly, the OMS-EGI descriptor was extracted from prior models of tools in [14], whereas we apply our descriptors directly to object point cloud segmentations from real world scenes. In addition, the types of features that we encode in each of the sub-parts are different and their design is also motivated by the difference in application.

3D shape descriptors have seen much development in recent years and can be broken down into two categories: *local descriptors*, that are fine-grained, where each point on an object surface carries its own geometric descriptor, such as Spin Images [15] or Signatures of Histograms of Orientations (SHOT) [16]; or *global descriptors*, that are more coarse-grained and typically operate at the object level where the objects have typically been segmented beforehand, like the Global Radius-Based Surface Descriptor (GRSD) [17].

Many of the 3D feature descriptors provided in the popular *Point Cloud Library (PCL)* [18], which formed the basis of the surface geometry descriptors presented in this work, are built on the core conceptual underpinning of *Point Feature Histograms (PFH)* [19]. Here we review some of these descriptors, focusing primarily on those that are, or that can be adapted to be, global descriptors. Given point cloud $P = \{\boldsymbol{p}_i\}$, associated estimated surface normals $N = \{\boldsymbol{n}_i\}$, and query point $\boldsymbol{p}$, the original PFH algorithm [19] involves firstly, finding pairs of points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j\,(i \neq j)$ in a local k-neighbourhood or within a certain radius from a given query point $\boldsymbol{p}$, as well as their associated point normal estimates $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$, and secondly, constructing a Darboux $\boldsymbol{uvw}$ frame coordinate system where

$$\boldsymbol{u} = \boldsymbol{n}_i,\ \boldsymbol{v} = \boldsymbol{u} \times (\boldsymbol{p}_j - \boldsymbol{p}_i)/\|\boldsymbol{p}_j - \boldsymbol{p}_i\|,\ \boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}, \quad (1)$$

thereby allowing for the calculation of normal angular deviations $\langle \alpha, \phi, \theta \rangle$ for each point pair as follows:

$$\alpha = \boldsymbol{v} \cdot \boldsymbol{n}_j,\ \phi = \boldsymbol{u} \cdot \frac{(\boldsymbol{p}_j - \boldsymbol{p}_i)}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|},\ \theta = \arctan(\boldsymbol{w} \cdot \boldsymbol{n}_j, \boldsymbol{u} \cdot \boldsymbol{n}_j). \quad (2)$$

These values are then binned into a histogram of size $5^3 = 125$ in the case of PFH [19], where each of the three feature dimensions are subdivided into five divisions. PFH is normally used as a local descriptor, where such a histogram is generated for each of the points in the point cloud with a relatively small nearest neighbour search radius. However, it may be used as a global descriptor where a histogram is calculated just once for a single point (the object centroid for example) if the search radius is set to the maximum distance between any two points in the point cloud, that is, large enough to encompass the whole object.

The original PFH [19] suffers from computational inefficiency since, not only does it pair the query point $\boldsymbol{p}$ with its neighbours, but it also pairs each of the neighbours with each other. Thus, for a point cloud with n points and local neighbourhoods with k neighbours, it offers a complexity of $O(nk^2)$. This motivated the subsequent the *Fast Point Feature Histogram (FPFH)* algorithm [20] which only considers point pairs between the query point and its neighbours, an efficiency referred to as the *Simplified Point Feature Histogram (SPFH)*, which helps reduce the complexity to $O(nk)$. FPFH constructs a histogram of features by constructing separate histograms in each of the three angular feature dimensions, dividing them into 11 subdivisions, and concatenating them together to form 33 bins. FPFH may also be converted to a global descriptor in the same way as PFH. A more advanced global descriptor formulation based on FPFH came later in the guise of the *Viewpoint Feature Histogram (VFH)* [21]. It is this descriptor that forms the basis of the first of our proposed contributions, as discussed in the following section.

II. ACTION-GROUNDED VIEWPOINT FEATURE HISTOGRAM

The original motivation behind the VFH was to incorporate pose estimation into the object recognition provided by PFH or FPFH, so where those algorithms are invariant to object pose, VFH is divided into both a viewpoint direction component and an extended FPFH component to describe the object geometry. The viewpoint component is calculated by finding the vector between a given viewpoint and the object centroid, translating it to each point in the object cloud, and binning the angles between it and each of their respective normals to a histogram. The final resulting histogram is a

concatenation of the histograms from each of the two components – three 45-bin histograms built from the extended FPFH component and a 128-bin histogram for the viewpoint component – making 263 bins in total. In the PCL, VFH may be optionally extended with a shape distribution component that measures the distances of the points to the centroid, normalises them, and bins them to an additional fourth 45-bin histogram to make 308 VFH bins in total. This idea was used to introduce the *Clustered Viewpoint Feature Histogram (CVFH)* [22] and allows for discrimination between object surfaces that might share similar normal distribution but differ in terms of their point distribution, e.g. elongated versus compact planar surfaces.

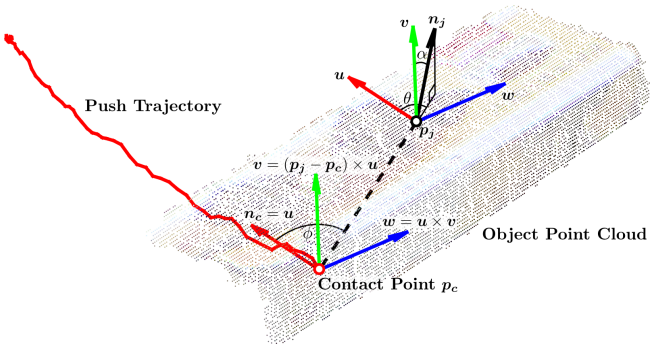

Fig. 1. Action-grounded Darboux frame construction example for the shape component of AGVFH.

In our proposed *action-grounded viewpoint feature histogram (AGVFH)*, we make two key changes to the original VFH descriptor, modifying both the shape component and the viewpoint component respectively, in order to ground the descriptor with respect to the pushing action. In the case of the shape component, instead of using the object centroid as the central point for the SPFH computation as in the original descriptor, we now use the push contact point. As well as that, we use the push direction normal as the basis for forming the Darboux frame. This is visualised in Figure 1 where the Darboux frame is formed between contact point $\boldsymbol{p}_c$ and a given point in the object point cloud $\boldsymbol{p}_j \in P$. The push vector $\boldsymbol{n}_c$ is found by fitting a line to the push trajectory from the contact point onwards using least squares regression. It is worth noting here that although we calculate the push vector using a push trajectory gathered after the fact in the experiments described in this paper, in principle such push vectors could just as easily come from pushes that are planned in advance by a robot. Thus, we construct a $\boldsymbol{uvw}$ Darboux frame coordinate system where

$$\boldsymbol{u} = \boldsymbol{n}_c,\ \boldsymbol{v} = \boldsymbol{u} \times (\boldsymbol{p}_j - \boldsymbol{p}_c)/\|\boldsymbol{p}_j - \boldsymbol{p}_c\|,\ \boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}. \quad (3)$$

The angular histograms are then calculated using:

$$\alpha = \boldsymbol{v} \cdot \boldsymbol{n}_j,\ \phi = \boldsymbol{u} \cdot \frac{(\boldsymbol{p}_j - \boldsymbol{p}_c)}{\|\boldsymbol{p}_j - \boldsymbol{p}_c\|},\ \theta = \arctan(\boldsymbol{w} \cdot \boldsymbol{n}_j, \boldsymbol{u} \cdot \boldsymbol{n}_j). \quad (4)$$

In the case of the viewpoint component, we use the contact point as the viewpoint and replace the vector between the viewpoint and the object centroid with the push vector. Our hypothesis, to be borne out experimentally, is that this type of encoding could provide an advantage in the object affordance learning setting, where the description of the object surface geometry relative to the action-grounded reference frame matters more than an invariant description.

III. ACTION-GROUNDED OCTREE SHAPE FEATURES

Our proposed *action-grounded octree shape features (AGOSF)* representation is similar in nature to our original action-grounded shape feature descriptor as proposed in [3], but uses octrees to decompose the point cloud into part cells instead of separately subdividing along the axes via partitioning planes. An octree is a recursively specified tree data structure that subdivides a three dimensional space into eight octants, each of which may be subsequently subdivided into a further eight octants, and so on. Thus, an arbitrary level of detail may be encapsulated by the representation depending on the octree depth level. This octree subdivision process is illustrated for a segmented object point cloud sample in Figure 2. The octree subdivision is similar in nature to that of the OMS-EGI descriptor by Mar *et al.* [14], but our proposed method differs in the types of features which are encoded in each of the octree cells. Note that the octree octants, which we refer to as *cells* in the remainder, have edge lengths that vary in each dimension depending on the structure of the object point cloud being decomposed. In each of the cells, we derive the following features:

- Local surface normal estimate: x, y, z components.
- Centroid: x, y, z components.
- Point count: 1-dimensional.
- Local curvature estimate: 1-dimensional.

The action-grounded nature of this descriptor is based on describing the above features, in particular the centroid features, with respect to the *action frame*, which is defined in the same way as previously described in [3], [4]. Thus, the action frame has its origin at the contact point on the object, its positive y-axis points in the direction of the pushing motion parallel to the table surface, its positive z-axis points upward from the table surface, and its positive x-axis points to the right of the object.

Our aim with this style of feature design was to capture both the local, fine-grained and the global, coarse-grained shape structure of objects at multiple resolutions. Prior to an object point cloud being decomposed using the octree subdivision, local estimates of surface normals are taken at each of the points in the point cloud using the standard method from the PCL as described in [23] (pp. 45–50). With these local point normal estimates in place, given a particular octree cell, a mean may be taken over the point normals contained within that cell in order to estimate the local surface normal. This method of estimating the local surface normal within the cell is an update from our approach in [3], where RANSAC was used to fit a plane to the points instead. The important point to note is that surface normal estimates within cells can provide the local, fine-grained shape description that we desire within the octree structure.

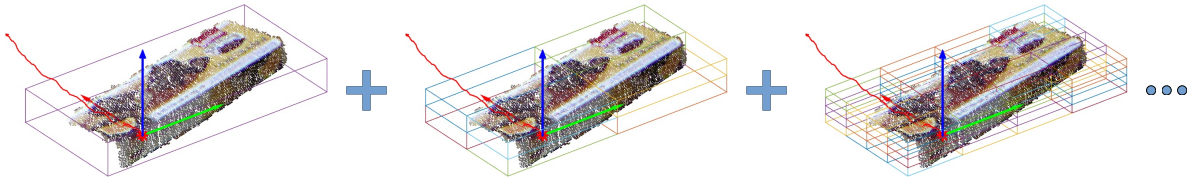

Fig. 2. Action-grounded octree shape feature (AGOSF) 3-level octree decomposition of a sample segmented object point cloud.

Meanwhile, the more global, coarse-grained structure is captured primarily by the centroids of the parts of the point cloud contained within each of the cells. This means that, even though both the point clouds of a tennis ball and a football would be decomposed into the same number of octree cells, the centroids of the cells would capture their respective differences in height, width and breadth. In addition to the cell centroids, we also include cell point counts in an effort to capture the structural mass of different parts of the object. Finally, we also include a local 1-dimensional curvature estimate in each of the cells, which is the mean over point curvature estimates derived by taking the ratio between the minimum eigenvalue and the sum of the eigenvalues of the covariance matrix of local k-neighbours of each point (in our experiments we selected a k value of 9). This method, which is again provided in the PCL and described further in [23] (pp. 45–50), replaces the curvature estimation methods we proposed previously in [24]. Thus, depending on the number of octree depth levels, the AGOSF feature descriptor can have different dimensionality for arbitrarily detailed representations, generalising to $\sum_{l=0}^{D}(8^l \cdot 8)$ for a given maximum depth level D.

There is, however, an issue with the above described representation that we have so far ignored. Sometimes, particularly in cases where an object is irregularly shaped or when the principle axes of a regularly shaped object are not well aligned with the action frame, some of the octree cells will be empty. This poses a problem because it is impossible to find either point normal estimates or a point centroid in an empty cell. Obviously this issue does not arise in the case of our own point counting or with the angular histograms of the OMS-EGI descriptor- in those cases, if a cell is empty, a zero is counted. In the absence of a theoretically obvious solution to this issue, given an instance of an empty cell, we decided to set the cell normal components, centroid components, point count, and curvature estimate, all to zeros.

IV. EXPERIMENTS

In the experiments described below, the goal was to take an affordance dataset of segmented object point clouds, push action trajectories and associated affordance labels, apply multiple different feature descriptors to the object point clouds making use of the push action information where possible for action-grounding, and train classifiers to learn a mapping of the form $f : \mathbb{R}^n \rightarrow \mathbb{N}$ from input feature vectors to affordance class labels, where n is the dimensionality of a given feature descriptor. We used a *random forests (RF)* [25] classification model with 500 trees as the classifier in all cases, as this has proven reliable from previous work [4].

A. Dataset

We used a slightly modified form of the human push affordance dataset described in [4] for our experiments, which consisted of an additional object alongside the five original household objects: four flat-surfaced objects; a book, a marshmallow box, a cookie packet, and a biscuit box, and two curved-surfaced objects; a yoghurt bottle and a coffee cup. These are illustrated in Fig. 3. The dataset was collected as follows. Pushes were performed on the objects by a human wearing a Polhemus Patriot™electromagnetic tracking device on their hand while a Microsoft Kinect sensor recorded point cloud data. The tracking device recorded the trajectory of the fingertip of the human at a frequency of 60Hz while they pushed the objects.

Objects were placed at random start locations and in various poses within the workspace and within view of the Kinect sensor, and the human experimenter would perform straight-line pushes on the objects, attempting to keep the pushes within reasonable limits of 5 different push categories: pushing through the top, bottom, left, right and centre of the objects respectively, from the direction of the field of view of the Kinect. Table I details the number of different affordances produced by each of the objects during the interactions. Certain objects prohibited certain certain affordances. For example, neither the cookie pack nor the book could easily be placed in sideways or upright poses and thus did not produce instances of the toppling affordance. The collected data contained 4 different affordance categories and the samples were hand-labelled with four ground truth labels to reflect this: *left rotation*, *right rotation*, *forward translation* and *forward topple*. For more details on this dataset and the data collection process, see [3].

Fig. 3. Test objects used in our experiments.

TABLE I
OBJECT/AFFORDANCE MATRIX

	Topple	Trans.	Left Rot.	Right Rot.	Total
Cookie Pack	0	6	6	6	18
Mallow Box	12	9	6	6	33
Biscuit Box	12	9	6	6	33
Book	0	6	6	6	18
Yoghurt Bottle	6	6	3	3	18
Coffee Cup	3	6	5	0	14
Total	33	42	32	27	134

B. Feature Descriptors

In our experimental evaluation we compared multiple different feature descriptors, including variations of the two descriptors proposed in this paper. These are listed in the results of Table II, where our proposed methods are italicized, and described here in more detail. In the cases where octrees are used in the feature descriptor, we append the octree depth level to the descriptor name, thus we use the notation AGOSF-2 to denote a 72-dimensional AGOSF descriptor constructed from a two-level octree, AGOSF-3 for a 584-dimensional descriptor constructed with a three-level octree, and so on. The "Action Features" in Table II refers to the addition of six features made up of the x, y and z components of the push contact point and the x, y and z components of the normalised push trajectory vector respectively in world coordinates.

1) PCL Descriptors: In the case of PFH, rather than estimating point feature histograms for every point in the point cloud, we form a global descriptor by estimating a PFH just for the object centroid with a calculation radius encompassing the entire cloud. We also tested a modification where, rather than using the object centroid as a basis for calculation, we use the push contact point. VFH refers to the PCL version of the VFH descriptor with three 45-bin histograms built from the extended FPFH component, a 128-bin histogram for the viewpoint component, and a 45-bin histogram for the shape distribution component to make 308 bins in total. In the "Origin Viewpoint" version, we used the default value of $(0,0,0)$ as the viewpoint. In the "Contact Point Viewpoint" version we use the push contact point as the viewpoint, but without the push vector viewpoint modification discussed in Section II. In all VFH cases, we use the scale variance as proposed in the CVFH paper [22] by keeping the histograms unnormalised. We also investigate two additional global descriptors from the PCL: the *Global Fast Point Feature Histogram (GFPFH)* [26] and *Global Radius-Based Surace Descriptor (GRSD)* [17].

2) OMS-EGI Descriptor: We implemented our own version of the OMS-EGI descriptor in Matlab. Note that our usage of the descriptor does not precisely follow the methodology of the original authors [14], since they applied it to prior models of objects, rather than to partial views of objects from segmented point clouds as in our case. In addition, they used the descriptor for clustering tool pose categories in order to subsequently inform an affordance prediction model, rather than for direct affordance prediction, as in our case.

3) Our Descriptors: AGVFH refers to our proposed action-grounded VFH feature descriptor as described in Section II. AGSF refers to our 35-dimensional action-grounded shape feature descriptor from [3]. Descriptors preceded by "Non-" refer to non-action-grounded equivalents of their counterparts, where the object point clouds are first transformed to the action frame in order to preserve similar object orientation, but whose centroids are then translated to the origin prior to feature extraction. The AGOSF features are as previously described in Section III.

C. Results

We performed $t = 10$ trials of k-fold cross validation with $k = 10$ using the random forests classifier on each of the descriptors and took the mean of the following F_1 score calculation over all trials and folds in order to evaluate their respective performance and produce the results in Table II:

$$\mathrm{TP} = \sum_{i=1}^{t}\sum_{j=1}^{k} \mathrm{TP}^{(i,j)},\ \mathrm{FP} = \sum_{i=1}^{t}\sum_{j=1}^{k} \mathrm{FP}^{(i,j)},\ \mathrm{FN} = \sum_{i=1}^{t}\sum_{j=1}^{k} \mathrm{FN}^{(i,j)},$$

$$\mathrm{F}_1 = (2\cdot\mathrm{TP}) \,/\, (2\cdot\mathrm{TP} + \mathrm{FP} + \mathrm{FN}),$$

where TP denotes the number of true positives, FP denotes the number of false positives, FN denotes the number of false negatives, and i and j index the trials and folds respectively. The multiple trials were conducted in order to mitigate against random variations in the learning process. The results in Table II have been divided for convenience into four groups of the following categories: surface geometry-based descriptors; our descriptors from [3], [4]; 2-level depth octree-based descriptors (OMS-EGI and AGOSF); 3-level depth octree-based descriptors. We present confusion matrices for AGOSF-3, AGVFH, and OMS-EGI-3 in Tables III, IV, and V respectively. In addition, we present the top 20 most relevant features as determined by the random forests algorithm (described as "variable importance" in [25]) averaged over all trials and folds for the AGOSF-3 descriptor in Fig. 4. Overall, as can be seen in Table II, the 3-depth octree-based AGOSF-3 descriptor outperforms all of the

TABLE II
10-FOLD CV RANDOM FORESTS CLASSIFIER RESULTS

Features	**F-Score**
Object Centroid PFH [19] + Action Features	0.4053
Contact Point PFH	0.2566
GFPFH [26] + Action Features	0.4550
GRSD [17] + Action Features	0.4021
Origin Viewpoint VFH [21]	0.3130
Origin Viewpoint VFH [21] + Action Features	0.3948
Contact Point Viewpoint VFH [21]	0.3688
Contact Point Viewpoint VFH [21] + Action Features	0.4725
AGVFH (cf. Sec. II)	0.8889
Non-AGSF [4] + Action Features	0.8706
AGSF [3]	0.9184
OMS-EGI-2 [14]	0.5751
OMS-EGI-3 [14]	0.6480
Non-AGOSF-2 (cf. Sec. III & IV-B) + Action Features	0.8801
AGOSF-2 (cf. Sec. III)	0.9391
Non-AGOSF-3 (cf. Sec. III & IV-B) + Action Features	0.7042
AGOSF-3 (cf. Sec. III)	**0.9526**

other methods with a relatively high mean F-score of 0.9588. We discuss the results in more detail in the following subsections.

TABLE III
10-TRIAL, 10-FOLD CV CONFUSION MATRIX: AGOSF-3

		Prediction			
		Topple	Translate	Left Rot.	Right Rot.
Truth	Topple	280	50	0	0
	Translate	10	410	0	0
	Left Rot.	0	0	320	0
	Right Rot.	0	0	0	270

TABLE IV
10-TRIAL, 10-FOLD CV CONFUSION MATRIX: AGVFH

		Prediction			
		Topple	Translate	Left Rot.	Right Rot.
Truth	Topple	270	60	0	0
	Translate	80	330	10	0
	Left Rot.	10	10	290	10
	Right Rot.	0	20	0	250

TABLE V
10-TRIAL, 10-FOLD CV CONFUSION MATRIX: OMS-EGI-3

		Prediction			
		Topple	Translate	Left Rot.	Right Rot.
Truth	Topple	270	60	0	0
	Translate	70	340	60	50
	Left Rot.	30	100	170	20
	Right Rot.	10	110	0	150

1) Surface Geometry-Based Descriptors: Of all of the surface geometry-based descriptors we tested, few, if any, performed particularly well in our experiments. The one exception is our AGVFH descriptor, which received a mean F-score of 0.8618. As can be seen in the confusion matrix of Table IV, it sometimes confuses topples for forward translations. We speculate that the reason why AGVFH does so much better than the rest of its surface geometry-based counterparts is because action-grounding the reference frame when computing the shape component heavily aids the discriminative capabilities of the descriptor in this type of affordance learning task (cf. Sec. IV-C.4).

2) Octree-Based Descriptors: In the case of the more volumetric octree-based representations, our proposed AGOSF descriptors also do quite a bit better than their OMS-EGI counterparts, at both the less-detailed 2-level depth, and at the more detailed 3-level depth. We note that the difference in performance here is most likely due to the OMS-EGI descriptor lacking a mechanism for describing the distances of octree cells with respect to one another, something that is encoded by the inclusion of centroid features in the AGOSF descriptor. Some evidence for this assertion is provided in the results of Figure 4, which shows that the top 20 most relevant features provided by AGOSF are almost exclusively centroid features. It is also worth noting that the octree-based AGOSF methods significantly outperform the surface geometry-based AGVFH when the third octree depth level is used. We reason that this is due to the multi-scale nature of the representation. Describing the shapes of the objects at multiple granularities is likely to provide an advantage over the angular histogram approach of AGVFH.

3) Action-Grounding: Our results support the general claim that action-grounding can improve performance in an affordance learning task such as the one presented in these experiments. Action-grounding the VFH descriptor provided a substantial performance improvement over its non-action-grounded analogues. Our previously proposed action-grounded features [3], [4] once more proved to be better than their non-action-grounded equivalent on a slightly expanded dataset. The action-grounding paradigm also appeared to work well in the case of octree-based descriptors, particularly as the depth of the octree increased.

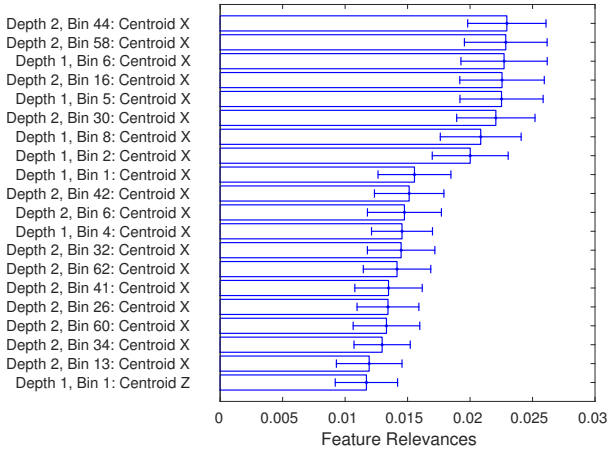

Fig. 4. Top 20 feature relevances for the AGOSF-3 descriptor averaged over all 10 cross validation folds in all 10 trials.

4) Feature Relevance: According to the results of Figure 4, the top 20 most relevant features provided by AGOSF-3 are almost entirely made up of the x components of centroid features from octree cells at various depth levels. This makes sense within the context of the presented affordance learning task, since, in order to be able to distinguish between forward translations, left rotations, and right rotations, when pushing forward towards an object, it is useful to know the relative positions of object parts along the x-axis of the action-grounded reference frame. In addition, we observe that one of the top 20 features is a centroid z component, which indicates some selectivity for features that might predict the toppling affordance.

Figure 5 shows a comparison between feature relevances for the AGVFH descriptor and the contact point viewpoint VFH descriptor with action features. There are noticeable differences here, but in particular, the most relevant features for AGVFH occur at the extremities of the ϕ bin as opposed to the centre of the ϕ bin for VFH. Since, in the case of AGVFH, the ϕ features measure the angular deviations between the push vector and the vectors between the contact point and other object points, it is likely that the most active ϕ features for a given object are on opposite sides of the spectrum for pushes that are instigated on opposite sides

of the object respectively. This may be the reason why the classifier learns to favour features that discriminate between opposite ϕ angles, since opposite ϕ angles indicate opposite pushes, which in turn often indicate opposite affordances, at least in this experimental setting.

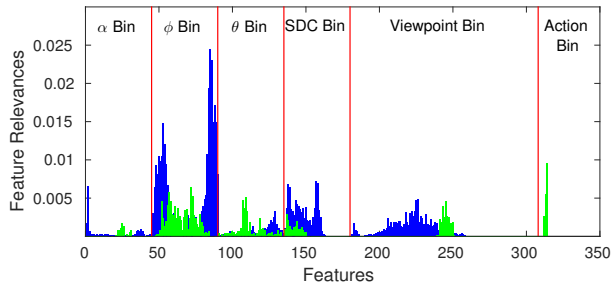

Fig. 5. Feature relevances for AGVFH (blue) vs. Contact Point Viewpoint VFH + Action Features (green).

V. CONCLUSIONS AND FUTURE WORK

We proposed two novel feature descriptors based on the paradigm of action-grounding for object affordance learning tasks. The first of these is based on modifying the viewpoint feature histogram such that both its shape and viewpoint components are grounded with respect to a push action reference frame. The second one is based on subdividing the object point cloud using a multi-scale octree and defining local features within each of the octree cells. We demonstrated the effectiveness of these methods in a cross-comparison with related descriptors in both shape geometry-based and octree-based categories on an object affordance learning dataset.

With regard to future work, we aim to test these feature description methods in affordance learning experiments using a real robot. It would be of particular interest to test the efficacy of the descriptors in affordance regression tasks where the objective is to predict the object motion vector or other continuous variables, such as the distance an object will travel. A wider variety of objects, including balls, cylinders, etc. would inevitably lead to a wider variety of affordances, but they would also be much more difficult to predict and to structure the experiments around such that the learning task is feasible. To this end, we anticipate that a more data-driven approach to the learning apparatus in combination with the essential characteristics of the types of feature descriptors we have proposed in the above could prove fruitful.

REFERENCES

[1] J. J. Gibson, *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[2] A. Stoytchev, "Some basic principles of developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 122–130, 2009.

[3] B. Ridge and A. Ude, "Action-grounded push affordance bootstrapping of unknown objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 2791–2798.

[4] B. Ridge, E. Ugur, and A. Ude, "Comparison of Action-Grounded and Non-Action-Grounded 3-D Shape Features for Object Affordance Classification," in *The 17th Int. Conf. on Advanced Robotics (ICAR)*, Istanbul, Turkey, 2015, pp. 635–641.

[5] D. Omrčen, A. Ude, and A. Kos, "Learning primitive actions through object exploration," in *8th IEEE-RAS Int. Conf. on Humanoid Robots*, Daejeon, Korea, 2008, pp. 306–311.

[6] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory-motor coordination to imitation," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[7] E. Uğur and E. Şahin, "Traversability: A Case Study for Learning and Perceiving Affordances in Robots," *Adaptive Behavior*, vol. 18, no. 3-4, pp. 258–284, June 2010.

[8] 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, July 2011.

[9] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta, "Exploring affordances and tool use on the iCub," in *2013 13th IEEE-RAS Int. Conf. on Humanoid Robots*, Atlanta, Georgia, 2013, pp. 130–137.

[10] M. Björkman and Y. Bekiroglu, "Learning to disambiguate object hypotheses through self-exploration," in *2014 14th IEEE-RAS Int. Conf. on Humanoid Robots*, Madrid, Spain, 2014, pp. 560–565.

[11] V. Högman, M. Björkman, A. Maki, and D. Kragic, "A Sensorimotor Learning Framework for Object Categorization," *IEEE Trans. on Cognitive and Developmental Systems*, vol. 8, no. 1, pp. 15–25, 2016.

[12] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action - initial steps towards artificial cognition," in *IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 3140–3145.

[13] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick, "Learning Contact Locations for Pushing and Orienting Unknown Objects," in *IEEE-RAS Int. Conf. on Humanoid Robotics*, Atlanta, GA, 2013, pp. 435–442.

[14] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Multi-model approach based on 3d functional features for tool affordance learning in robotics," in *IEEE-RAS Int. Conf. on Humanoid Robots*, Seoul, Korea, 2015, pp. 482–489.

[15] A. E. Johnson, "Spin-images: a representation for 3-D surface matching," Ph.D. Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1997.

[16] F. Tombari, S. Salti, and L. D. Stefano, "Unique Signatures of Histograms for Local Surface Description," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer, 2010, no. 6313, pp. 356–369.

[17] Z.-C. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2d–3d categorization and classification for multimodal perception systems," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1378–1402, Sept. 2011.

[18] R. B. Rusu and S. Cousins, "3d is here: Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 1–4.

[19] R. Rusu, N. Blodow, Z. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, 2008, pp. 3384–3391.

[20] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3d registration," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, 2009, pp. 3212–3217.

[21] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 2155–2162.

[22] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. Rusu, and G. Bradski, "CAD-model recognition and 6DOF pose estimation using 3D cues," in *ICCV Workshop on 3D Representation and Recognition (3D RR11)*, Barcelona, Spain, 2011, pp. 585–592.

[23] R. B. Rusu, "Semantic 3D object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science Department, TU Munich, Germany, 2009.

[24] B. Ridge, A. Leonardis, A. Ude, M. Deniša, and D. Skočaj, "Self-Supervised Online Learning of Basic Object Push Affordances," *Int. Journal of Advanced Robotic Systems*, vol. 12, no. 24, 2015.

[25] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] R. B. Rusu, A. Holzbach, M. Beetz, and G. Bradski, "Detecting and segmenting objects for mobile manipulation," in *2009 IEEE 12th Int. Conf. on Computer Vision Workshops*, Sept. 2009, pp. 47–54.