Scalable Real-Time and One-Shot Multiple-Affordance Detection

Eduardo Ruiz and Walterio Mayol-Cuevas

***Abstract*— This paper develops and evaluates a geometry-driven approach that allows for the detection of affordances in an scalable and multiple-instance manner. True to Gibson's idea of *direct* and economical perception, our approach requires little supervision, is straightforward to compute and is agnostic to semantics. The proposed approach is trained with a single interaction example on synthetic data (i.e. simulated), yet is able to generalise to previously unknown RGB-D scenarios without further training. Furthermore, our use of geometric information not only allows to detect *what* a location in the environment affords but also *how* it affords, i.e. object-pose. We show results from several dozens of affordances (80+) predicted simultaneously at high frame rates on indoor environments such as kitchens and offices. Our evaluations show high rates of precision and that the algorithm's predictions align well with crowd-sourced human validations.**

I. INTRODUCTION

Cognitive robots that need to understand and interact with their surroundings can greatly benefit from perceiving and learning based on environment's functional properties or affordances. The concept of affordance was coined by James J. Gibson [1] more than five decades ago in the field of ecological psychology. For Gibson, affordances are action opportunities in the environment that are *directly* perceived by the observer. According to this, the goal of vision was to recognise the affordances rather than elements or objects in the scene. However, perhaps motivated by the top-down view adopted in computer vision research, much of the attention given to the problem of affordances has focused on the recovery of complex representations of the world, internal symbolic relationships or semantic information, which undermines the idea of *direct* and economical perception of affordances proposed by Gibson.

The problems of visually recovering the "valid" properties of the environment that allow to detect affordances are further accentuated in robotics, this is due to the fact that robots need to be able to work in environments that are cluttered, unstructured and unknown. Developing a system that is able to work under these conditions is a difficult problem; more so when traditional affordance detection approaches often need to recognise objects semantically in the environment, or to have previously extensively trained for as many cases (examples) as possible in order to generalise to novel scenarios. Robots would benefit from affordance detection approaches that do not rely on object recognition, nor environment's features costly to estimate; dropping or relaxing such requirements in the perception system can allow robots to have greater generalisation capabilities, enabling them to accomplish their task efficiently.

Authors are with the Department of Computer Science, University of Bristol, Bristol, UK {er13827,cswwmc}@bristol.ac.uk

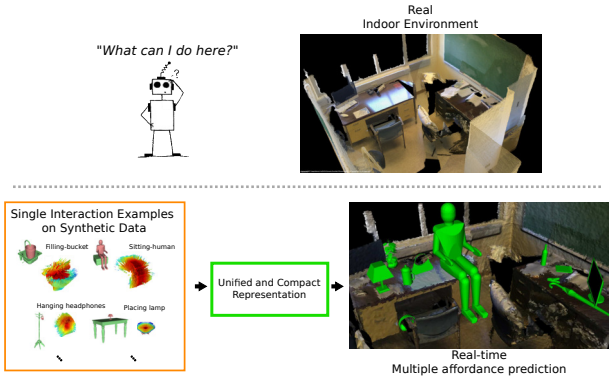

Fig. 1. We propose an algorithm that leverages single-interaction examples to devise a multiple-affordance representation. This unified and compact representation along with the proposed algorithm allow for real-time prediction of multiple affordances simultaneously. The proposed work enables a robot to explore a novel scenario and answer the perceptual question of "*What can I do here?*" or "*What can I afford to do here?*".

In previous work [2], we introduced a novel affordance descriptor that is able to richly characterise the relations between pairs of objects: The Interaction Tensor. In this paper, we expand the capabilities of the approach by first demonstrating the power of the representation to characterise generic pairs of objects. Then, we show that the approach allows to predict several dozens (80+) of interactions at high rates. More importantly, we demonstrate that the approach is able to predict multiple affordances in previously unknown RGB-D environments by *training* from a single example. Examples of the affordances detected with our method are shown in Fig. 1 and 6.

The paper is organised as follows. Section II describes most common approaches studied to date in affordance detection. Section III describes the core of our approach for scalability before we present the evaluation and results of our algorithm in Section IV. Finally, Section V presents our conclusions and avenues for future work.

II. RELATED WORK

Affordance detection has been studied in recent years in the computer vision and robotics fields. Generally speaking, affordance knowledge has been incorporated in learning systems that use data from demonstrations of interactions, robot self-exploration as well as systems that learn from labelled data. In terms of applications, the approaches have considered semantic scene understanding, grasp learning,

gesture recognition, object segmentation and planning in goal-directed tasks.

A big body of research in affordances for robots comes from the developmental robotics field [3], [4], [5]. Generally speaking, the focus of approaches in this field is on the representation and learning of robot's actions and their consequences in the environment. The core of this approaches studies the problem of affordance learning as a problem of structure learning, representing affordances as the (probabilistic) relations between actions, objects, and effects. These models have enabled robots to learn two-object *relational* affordances [6], higher-level manipulation actions [7], and two-arm manipulation [8]. More recently, *relational* affordances have been studied for learning tool usage [9], [10], [11], [12], [13], where the general idea is to learn the effect that one object (i.e. tool) has over or relative to another, allowing to plan and achieve a target effect or object configuration.

Another big body of research has been dedicated to the study of grasping affordances, which is an important skill for robots in order to interact with objects in the environment. Recent examples of this type of work are [14], where a robot learns semantic (constrained) grasps from humans demonstrations, [15], [16] who generate grasp hypotheses based on the detection and fitting of geometric priors; or grasp affordance proposals in a Pick and Place scenarios [17].

Learning what the environment affords to others (e.g. humans) has also been studied in robotics. For instance, methods to anticipate humans activities and assist in everyday tasks [18] or approaches that allow robots to show human-like behaviours [19]. *Human* affordances have also been actively investigated in the computer vision community; for instance, hallucinating humans in indoor environments [20], [21], [22], [23] or learning from labelled data [24]. Overall, these approaches aim to learn suitable locations for humans to sit, stand, walk, etc.

One more area where important work has been carried out is learning single-object and object-part (or tool-part) affordances from labelled examples. [25], [26], [27], [28] characterise single-object affordances such as *containment* or *sittable* using CAD models. In approaches such as [29], [30], [31], affordances of parts of objects are studied; for instance, to learn parts of an object that afford *cutting* and look for potential replacement tools. More recently, deep-learning methods have been exploited for the problem of object affordances [32], [33], [34]; these approaches leverage the ability of deep Convolutional Neural Networks (CNNs) to learn features from large collections of annotated data.

Notably, recent approaches for affordance detection are heavy in terms of requiring multiple learning examples or extensive training phases. Despite the success of human-inspired learning stages, approaches are often limited to a small set of objects and affordances. It is not clear how the models would apply for novel objects or novel realistic environments. Methods that have concentrated efforts in a single type of interaction (i.e. *grasping* or *human* affordances) have achieved remarkable results when facing novel realistic scenarios, yet the question remains open about the generalisation of the approaches for other types of interaction or scenarios that do not require manipulation.

Remarkably, geometric information has proven to be a strong cue for affordance detection in the previous approaches. The advantage of geometric features over alternative representations, such as texture or colour, is that geometry provides a stronger generalisation power since the geometry of everyday objects strongly dictates the physical interactions afforded by objects in the world. In contrast to works such as [35], [36], our method does not build on higher-level geometric primitives nor complex features computed on the environment. Moreover, the general purpose nature of our representation allows to characterise affordances for *simple* objects such as a mug but also enables the representation of more complex interactions like a human riding a motorcycle. Contrary to methods in computer graphics studying functionally analysis [37], [38], the approach introduced in this paper takes into account visually perceived information, does not require highly detailed geometries and is straightforward to compute.

III. SCALABLE AFFORDANCE DETECTION

Our approach expands on [2], where an algorithm to detect geometric affordance locations in indoor scenes is presented. We propose an algorithm to enable scalable multiple-affordance detection. The approach that we proposed allows us to increase the number of affordance-object pairs queried simultaneously at test time without heavily compromising detection rates. Briefly speaking, the approach that we present agglomerates multiple affordance descriptors and performs a grid-based clustering to select a reduced number of keypoints (centroids) required to make predictions at test-time. This is aimed for parallelisation and efficient evaluation. Using this algorithm a robot could answer questions such as "What can be afforded here?" on multiple point locations of an input scene.

In the following subsections we provide details of the proposed algorithm by first briefly summarising our previously introduced affordance representation: the Interaction Tensor.

A. The Interaction Tensor

The Interaction Tensor (iT) [2] is a tensor field representation that characterises affordances between two entities. Using direct, sparse sampling over the iT allows for the determination of geometrically similar interactions from a single *training* example; this sampling comprises what is called *affordance keypoints*, which serve to more quickly judge the likelihood of an affordance at a test point in a scene. The iT is straightforward to compute and tolerates well changes in geometry that provide good generalisation to unseen scenes from a single example. The iT example of any affordance is created with 3D or CAD models of the interacting objects, these objects are placed relative to each other simulating the interaction that they would have on real circumstances. Then, using a dense pointcloud representation of the object the

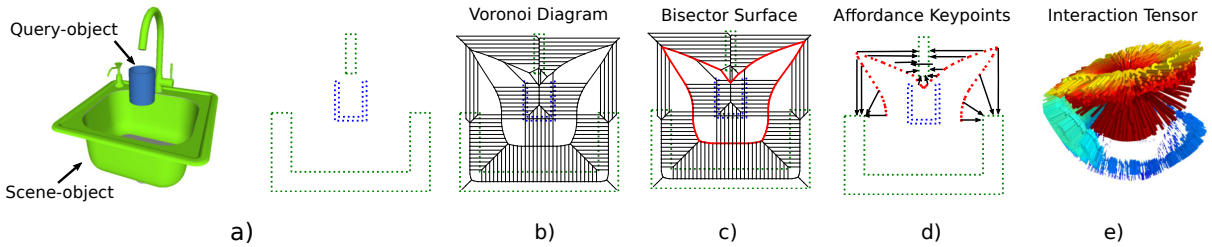

Fig. 2. Computation of an interaction tensor from an affordance of interest. a) Objects are placed simulating the intended interaction in pointcloud representation. b) Voronoi diagram over the whole pointcloud (i.e. both objects). c) Interaction Bisector Surface formed by ridges shared between different objects. d) Affordance keypoints and their associated provenance vectors. e) 3D full interaction tensor for *Filling* a mug.

Interaction Bisector Surface (IBS) is computed; this surface is approximated by computing the Voronoi diagram for the complete pointcloud (i.e. both objects) and retaining only the ridges shared by points from different objects. Information regarding the regions that contributed to the computation of a point on the IBS is preserved in what is called **provenance vectors**. This process generates the iT for the affordance simulated by the two objects. Fig. 2 illustrates the process to obtain an iT for the interaction between a pair of objects.

We name the objects involved in every interaction as **query-object** and **scene-object** (or scene), respectively. A query-object is an object with a *known* affordance and a scene-object is the object (or part) completing the interaction. For instance, in the interaction shown in Fig. 2 (i.e. Filling a mug), the mug represents the query-object and the tap acts a scene-object. The iT is a weighted vector field (*provenance vectors*), where the weight assigned to any location follows the intuition that areas where the objects come closer together are more relevant for the interaction. More precisely, weights are computed as follows

$$w_i = 1 - \frac{|\vec{p_i}|}{|\vec{p}_{\max}|} \tag{1}$$

Eq. 1 assigns higher weights to locations of the interaction where the objects come closer together or touch, these locations have their associated *provenance vectors* $\vec{p_i}$ with a small magnitude. The opposite happens to the weights of locations with larger *provenance vectors*. Examples of weighted iTs can be seen Fig. 3, notice the high-weight (red) assigned to the region corresponding to the *hook* of the coat-hanger. Fig. 3 shows examples of the iT descriptors for two affordances with different objects; this exemplifies the robustness of the iT to changes in the geometry of the interacting objects.

The descriptor for any given affordance is comprised by sampling N *affordance keypoints* from the iT example ($N = 512$ in our experiments). The sampled affordance keypoints represent one orientation of the query-object (the same from the *training* example); we generate a new descriptor by rotating the *affordance keypoints* around the gravity vector $\vec{z}$, which allows us to detect affordance candidate locations at multiple orientations (8 orientations in $[0, 2\pi)$ evenly distributed for our experiments). Thus, the affordance descriptor for a target interaction is comprised by 4096 *keypoints*, where each keypoint is a 6-dimensional feature vector (3 components for xyz coordinates of the IBS point and 3 for the provenance vector components). As shall be shown later, there is no processing impact given our scalable approach.

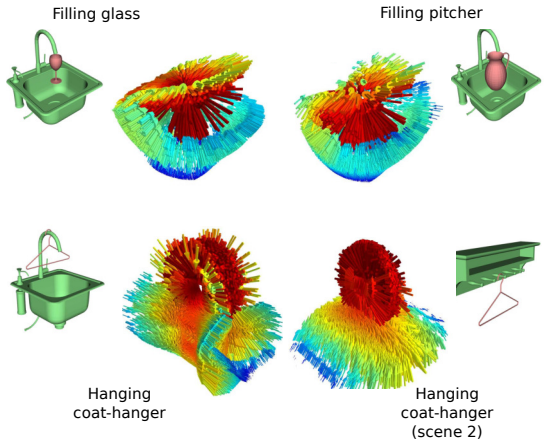

Fig. 3. iT examples for a subset of the objects in our study. Top row shows iT examples for *Filling* objects with different geometries. Bottom row shows iT examples for *Hanging* the same object (coat-hanger) on different scenes (i.e. scene-object).

B. iT Agglomeration

The algorithm that we propose for scalable and multiple-affordance detection follows a one-shot learning approach, i.e. it uses a single example from every affordance to devise the multiple-affordance descriptor. With such representation and algorithm, one can give answer to questions such as "*What can I affford to do here?*" on multiple point locations of an input scene without the need to individually test affordances. The approach allows to increase the number of affordance-object pairs queried simultaneously at test time without heavily compromising detection rates.

For this work, a total of 84 affordance-object pairs are considered, they include CAD models[1] of multiple household items from a wide range of geometries and dimensions, and its inspired by standard robotic manipulation datasets such as [39]. Human models for *Riding* and *Sitting* are also included in order to test "human" affordances. The specific affordance-object pairs are shown along the x-axis in Fig. 9. Note that some objects afford more than one thing e.g. *Fill-Pitcher* and *Hang-Pitcher*. It should also be noted that only objects with bounding box diagonal larger than 10cm long were considered for this experiments; this decision was

[1] `https://3dwarehouse.sketchup.com`

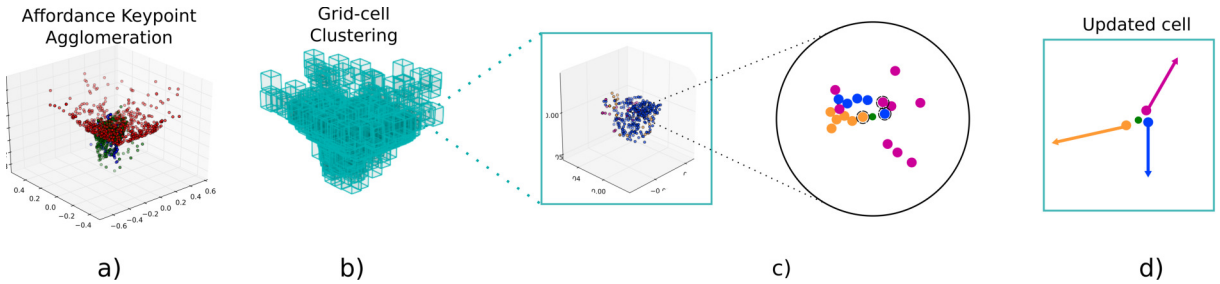

Fig. 4. Agglomeration of affordance descriptors and grid-based clustering of an example agglomeration of 3 affordances. a) single-affordance keypoints are agglomerated (affordances shown as different colours), b) uniform-size cell grid is fitted to the pointcloud, c) one cell can potentially contain many keypoints from multiple affordances, only closest keypoint (per-affordance) to the cell centroid (green) is taken into account during the update process, d) an updated cell with the provenance vectors associated to the keypoints kept after clustering.

taken in the knowledge that standard RGB-D sensors would fail to recover pointclouds for such dimensions; for instance, a screw, a washer or a coin.

For every interaction we compute an iT, i.e. 84 descriptors in total. Once all the descriptors are computed, we agglomerate them in a single pointcloud on which we perform clustering with the algorithm shown in Algorithm 1. First, we fit a grid of uniform-size cells covering every single *affordance keypoint*. Then, we use as seed-points only the centroid of non-empty cells. For every one of these cells, we only keep the one keypoint that is closest to the centroid in a per-affordance basis. For instance, one cell could contain 100 keypoints, all coming from the descriptor of *Placing* a bowl; after the iT clustering process is carried out this cell will only contain the keypoint closest to the cell's centroid. Finally, the descriptor $X_{agglomerative}$ (X_a for short) is obtained by updating the centroids using the keypoints within each cell, and keeping track of the *provenance vectors* associated with them. Fig. 4 illustrates the cell update process of the iT clustering algorithm (steps 6-10 of Algorithm 1).

Algorithm 1 iT clustering

Input: Affordance keypoints $X = \{x_1, ..., x_k\}$, cell size e
Output: Cluster centroids $C = \{c_1, ..., c_j\}$

1: Initialize C with centroids evenly distributed in $[x_{\min}, x_{\max}]$ with increments e.
2: **for all** x in X **do**
3: Assign x to cluster $argmin_j \|x - C_j\|_2$
4: Remove empty clusters
5: Initialize update sets $Y_1, ..., Y_j$ to empty
6: **for all** Clusters C **do**
7: **for all** Affordances $A = \{a_1, ..., a_k\}_{\neq}$ present in C_j **do**
8: Recover all x from affordance a_k
9: Assign $argmin_n \|x_n - C_j\|_2$ to Y_j
10: Update centroids: $c_j \leftarrow \frac{1}{|Y_j|} \sum_{y \in Y_j} y$

C. One-shot Prediction

The clustering process leads to a reduced number of 3D points (centroids) that represent a large number of affordance keypoints. This reduced set and their associated *provenance vectors* are used to compute and predict affordance candidate locations at test time. As discussed earlier, we aim for a method that allows a robot answering the question of "*What can I do here?*" in any given location of an input scene. In order to answer that for up to 84 affordances, we carry out the following procedure:

1) **Uniformly randomly sample a test-point** t_i in the input scene. Uniform random ensures that the algorithm takes into account locations across the whole input scene. Keep in mind that our approach is agnostic to pre-assumed features in the scene affording an interaction. Thus, it is until the interaction is carried out (by hallucinating) that the affordability of the interaction is discovered.
2) Apply the transformation T_{t_i} in order to **align the agglomerative descriptor** $X_{\mathbf{a}}$ **relative to the test-point** in the scene. Since our descriptor already takes into account multiple orientations for any given affordance, the transformation is a straightforward translation. In homogeneous coordinates

$$X'_a = T_{t_i} X_{\mathrm{a}} \tag{2}$$

3) **Compute the 1-NN in the scene for every keypoint** in the agglomerative descriptor X'_a using the voxel surrounding test-point t_i. The dimension of the voxel is proportional to the size of X'_a (diagonal of bounding box). Focusing the NN-search in this voxel alleviates expensive computations associated with this step.
4) **Estimate *test-vectors*** and compare against *provenance vectors* **to produce a score**. Test-vectors are the approximation of the provenance vectors when computed on a novel scenario, a test-vector goes from a keypoint in the descriptor to its nearest neighbour in the vicinity (voxel) of the current test-point. The score is a straightforward vector comparison computed via Eq. 3.

$$s^k = \frac{1}{N^k} \sum_{i=1}^{N^k} \frac{1}{\sqrt{2\pi (w_i^k)^2}} e^{-\frac{(\Delta_i^k)^2}{2(w_i^k)^2}}, \tag{3}$$

with

$$\Delta_i^k = \frac{\|\vec{v_{t_j}} - \vec{p_i^k}\|}{\|\vec{p_i^k}\|}, \quad k \in [0, 84]$$

where w_i^k is the weight of i-th keypoint of affordance k and it is computed from *training* (Eq. 1). Δ_i^k is the

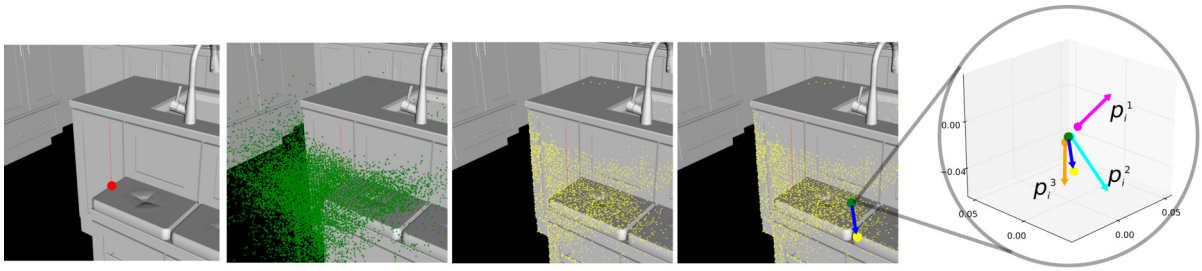

Fig. 5. Illustration of affordance prediction at test-time. a) A test-point is sampled from the input scene (red), b) The agglomerative representation (green) is aligned relative to this test-point, c) The 1-NN in the scene (yellow) for every centroid in the agglomeration, d) An example test-vector (blue) from a cell centroid to its closest scene point, e) A test-vector is compared against the stored provenance vectors p_i^k associated with affordance keypoints in that cell. In this particular cell, 3 scores are obtained.

difference between test vector $\vec{v_{tj}}$ (estimated using the j-th cell-centroid) and provenance vector p_i^k. Fig. 5 illustrates the process followed at test-time in order to predict affordance candidate locations. The intuition behind the scoring function is fitting a Gaussian window to every keypoint in the descriptor. The width of this window is regulated by the keypoint's weight w_i^k; thus, smaller differences between the expected (provenance) vector and the test vector translate into a higher score (per point). Notice that, given that the agglomerated affordance keypoints represent multiple orientations, the prediction algorithm produces a score for multiple oriented interactions. Therefore, we not only are able to determine if a point in the scene *affords or not* but also *how it affords*, i.e. the orientation of the query-object that in conjunction with the scene enables the interaction.

IV. EXPERIMENTS AND EVALUATION

For our experiments, we perform affordance predictions in synthetic scenes and real indoor scenarios. Our one-shot affordance prediction experiments include results of over 150 RGB-D scenes (randomly selected) from ScanNet [40], which include kitchens, living-rooms and offices. Briefly speaking, the experiments consist in executing the steps 1-4 in Section III-C for an input scene. For evaluations, we investigate the effect of the parameters used by our method (e.g. cluster size) in terms of prediction rates and human validations. Fig. 6 shows affordance prediction examples produced with out method for various indoor scenes. These figures are generated offline, checking for collisions among query-objects. The problem of deciding on-line (i.e. at test time) "what happens where? ", or which affordance should take place at a particular location (out of all the possibilities) is regarded as non-trivial and has not been addressed for now.

A. Cluster size and detection rates

One important parameter of our approach for multiple-affordance prediction is the cell size employed to cluster *affordance keypoints*. This is somewhat a compromise of parallelisation capability and framerate operation. One first consideration of this work explored non-uniform spatial representations of the keypoint agglomeration, representations such as those in e.g. Octrees. However, the diversity in

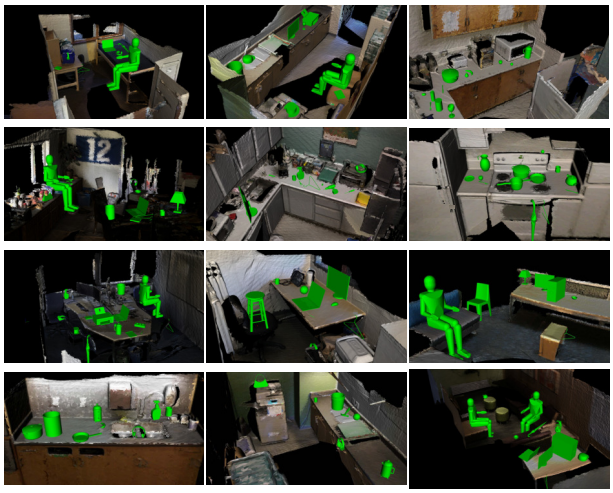

Fig. 6. Multiple-affordance detection examples for RGB-D indoor scenarios. In green are shown query-objects with their predicted poses

dimension and sparsity of the affordances considered in this research made very challenging the selection for the right positioning of centroids, which did not perform as well as sparse yet uniform-sized cells.

Fig. 7 shows the dimensionality of the multiple-affordance representation and the average prediction rates according to the cell size.

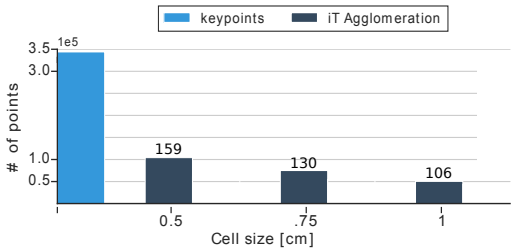

Fig. 7. Bar plot shows the dimensionality reduction achieved with the agglomerative method for different cell sizes. The number of keypoints required to make predictions is reduced up to 6 times. Numbers above each bar show prediction time (milliseconds) per test-point of the input scene.

Looking at this figure, it stands out the large reduction that is achieved with the proposed approach, which is nearly six times smaller (344K vs 60K keypoints). The prediction rates on the same figure show that using grids with a cell

size of 1 cm^3 allows the detection of up to 84 affordances at 10 different locations per iteration on the input scene. This is significantly faster (6x improvement) than predicting affordances by trying descriptors individually at test time one after the other. As noted before, the prediction algorithm performs a NN-search in order to estimate test-vectors and compare them against provenance vectors, the complexity of such operation depends heavily on the dimension of the multiple-affordance representation (i.e. the number of centroids/keypoints). More points in the representation require more computations; thus, reducing the representation allows for faster evaluations at test-time. As shall be shown later, our approach is able to produce top-quality predictions even with such a reduction in dimensionality the approach.

In an effort to further emphasise the scalability of the **iT** agglomeration method, Fig. 8 shows the computation times observed during affordance predictions of 84 affordance-object pairs. The green curve in this figure corresponds to the computation time measured by progressively augmenting the number of affordances represented by an agglomerative descriptor of 0.5cm^3 cells. That is, the time shown in the far right corresponds to an agglomeration of 84 affordances, whereas the first value on the left corresponds to a descriptor computed by agglomerating keypoints of 1 affordance (*Sitting*-human). Observe that the time grows sub-linearly on the number of keypoints added to the agglomeration. This figure also shows the time required to predict affordances by testing individually one after the other, which requires approximately 644 ms per test-point in the input scene.

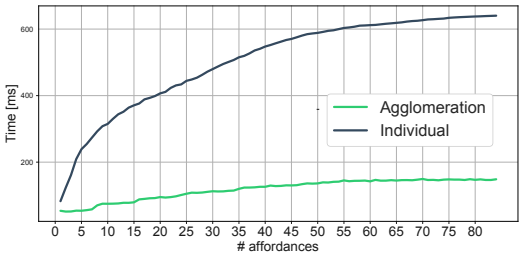

Fig. 8. Plot shows the average time needed to predict multiple affordances in any given scene location. In green is shown the time for agglomerartive representations of various numbers of affordances (1 to 84). In blue is shown the time required to predict affordances by testing one after the other (1 to 84). Notice that green grows sub-linearly with respect to the number of affordances added to the agglomeration.

B. iT Agglomeration vs Single affordance prediction

In our previous work, we demonstrated the outstanding performance of the **iT** method for detecting affordance candidate locations on an individual basis (i.e. querying one affordance at the time). We used this evidence to asses the performance of our multiple-affordance approach. In this sense, the predictions made with the agglomerative representation were compared against those produced in a single-affordance scenario (i.e. as in [2]). This is done with the intuition that achieving a good performance with this baseline would translate into predictions of similar quality, i.e. meaningful affordance predictions according to human criteria.

First, we run the prediction algorithm for every interaction in our study, i.e. 84 sets of affordance predictions; these results are then treated as "ground-truth" in order to compute performance metrics for our multiple-affordance approach.

Fig. 9 presents the performance achieved for all interactions under investigation. In this figure can be seen that 1cm-cells perform better for *Sitting* and *Placing* of medium-to-big objects, the exceptions being *Placing* pc-case. All cell sizes seemed to struggle with *Placing* coat-hanger, fork, tv-remote and pc-mouse. Another interesting result is that *Placing* smaller objects such as a knife, scissors, plate, spoon and pencil are predicted more reliably with the smallest cell size. This is explained by the fact that **iT**s of smaller objects are comprised of shorter *provenance* vectors; these vectors are not well represented when the cell size is increased (e.g. resolution loss). As suspected, smaller objects require a more fine-grained representation, such as the one achieved with 0.5 cm cells.

Table I shows performance metrics for top-level or *generic* affordance categories of the agglomerative approach for various cell sizes. It can be seen that overall the method performs best for *Filling* and *Hanging* affordances, where every single prediction made with the agglomerative representations was also a good location for the single-affordance baseline. It is also worth noticing that *Riding*, which is regarded as the more complex interaction in this study, has the best performance with a cell size of 0.5 cm^3. In contrast, the predictions for *Placing* affordances, which are arguably the least complex interactions, are better when a larger cell size is employed. This can be explained by the fact that *Placing* an object relies on vectors located under the objects; these vectors are very small (i.e. millimetric) when the cell size gets smaller. Vectors estimated at test-time rarely present such magnitudes due to the density of the scene pointcloud. In other words, agglomerative representations of larger cell sizes comprise larger *provenance vectors* that are more easily matched during test-time.

TABLE I
AVERAGE PERFORMANCE FOR DIFFERENT CELL SIZES.

	0.5cm			0.75cm			1cm		
	Precision	**Recall**	**F-1 score**	**Precision**	**Recall**	**F-1 score**	**Precision**	**Recall**	**F-1 score**
Filling	94.28	49.44	0.6261	98.19	27.55	0.4224	99.74	5.6663	0.1063
Hanging	97.08	18.11	0.2792	98.69	10.17	0.1724	98.69	2.2375	0.0418
Placing	92.34	59.48	0.6853	90.48	32.44	0.4613	84.13	5.5334	0.0954
Riding	73.24	60.70	0.6646	65.00	47.26	0.5473	64.30	33.51	0.4406
Sitting	23.85	16.76	0.1968	50.00	14.95	0.2302	91.57	13.15	0.2300
Average	76.16	40.90	0.4904	80.47	26.48	0.3667	87.68	12.02	0.1828

C. Affordance Validation

Affordance predictions on their own are elusive to ground-truth without subjective judgement or evaluation of the likelihood of an interaction. Due to the fact that no prior assumptions are made regarding objects in the scene affording the interactions, the predictions consists of locations that an agent would choose to accomplish an action. As an example, one can afford to place a bowl on a chair as much as one can sit on the kitchen's table. These are arguably valid placings

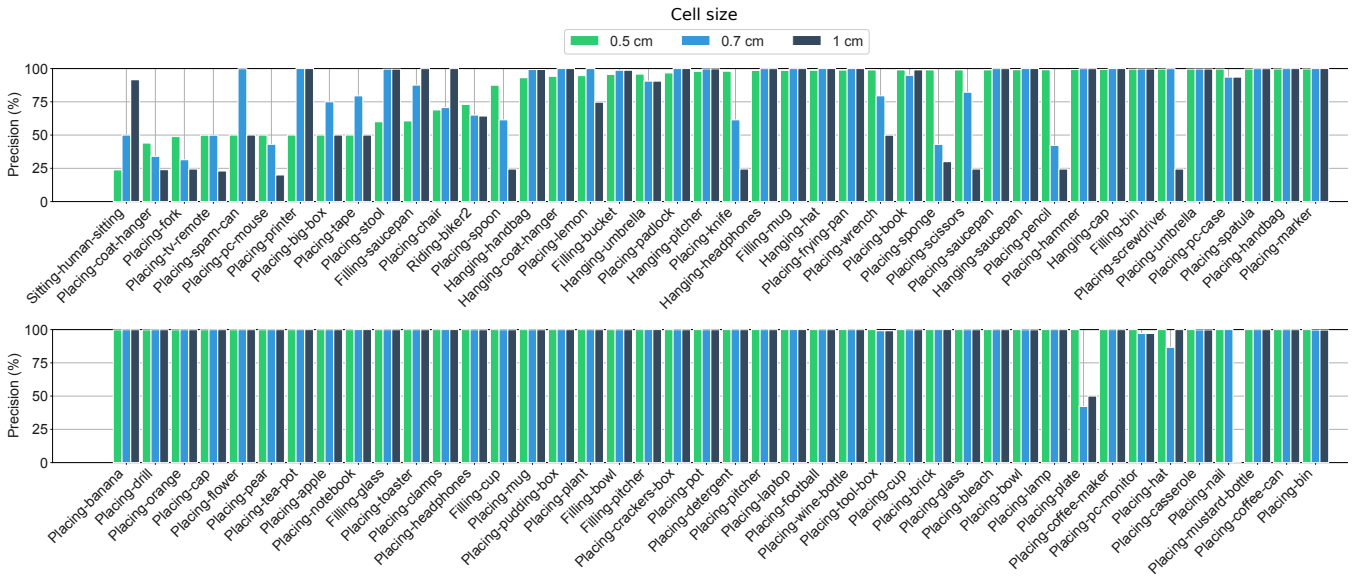

Fig. 9. Precision achieved with agglomerative representations produced by cell sizes of different sizes.

but need to be validated by an "agent", in our case, a human. We use human criteria to validate and asses the quality of our affordance predictions.

In our work, affordance predictions are made by setting a threshold to the output (score) of the one-shot algorithm. First, we use Amazon Mechanical Turk to determine the threshold that produces the best results. Here, people are asked to evaluate the predictions made with the agglomeration algorithm based on the smallest cell size. A total of 2.4K example predictions representing different scores were shown to 42 human evaluators (turkers). These turkers had to select a "winner" from two possible options showing the same affordance-object pair resulting from different scores. A "true" ranking based on human evaluation is computed by fitting a Bradley-Terry model [41] to the pairwise comparisons, with this ranking we asses the performance of the **iT** agglomeration algorithm and we find the optimal threshold that results in the optimal detections. Fig. 10 shows the family of classifiers induced by setting different threshold values at the score of the **iT** agglomeration algorithm.

The ROC plot shows that the method achieves a good performance according to human criteria when considering predictions made with a score above 0.7. In other words, the affordance predictions with a score above this threshold are deemed as good candidates all the time.

V. CONCLUSIONS

We have developed and evaluated a scalable and real-time approach for multiple affordance prediction on RGB-D scenes. Based on a single *training* example per interaction we are able to predict affordance candidate locations on previously unseen scenes for over 80 object-affordance pairs in a single iteration carried out every 106ms. Experiments are carried out on 150 RGB-D scans of indoor environments from a publicly available dataset. We have shown that top-quality affordance detections can be achieved by exploiting

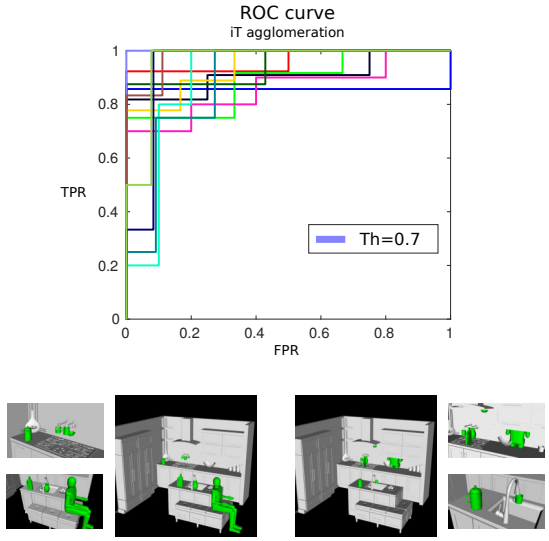

Fig. 10. Mechanical turk evaluation. ROC plot shows the family of classifiers generated by setting different thresholds bands to the prediction score; the best result is obtained with a value of 0.7. Images on the bottom exemplify the type of judgement humans had to make: choosing the image that best depicts the interactions (query-objects in green).

the geometric information involved in the interaction between two entities. One interesting avenue for future work is the investigation of grasping affordances, i.e. the interaction between a hand and other objects. For instance, the model of the robot's hand would serve as a query-object that interacts with a scene-object in the environment.

REFERENCES

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

[2] Ruiz Eduardo and Mayol-Cuevas Walterio. Where can i do this? geometric affordances from a single example with the interaction tensor. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, May 2018.

[3] Angelo Cangelosi and Matthew Schlesinger. *Developmental robotics: From babies to robots*. MIT Press, 2015.

[4] Huaqing Min, Changan Yi, Ronghua Luo, Jinhui Zhu, and Sheng Bi. Affordance research in developmental robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 8(4):237–255, 2016.

[5] Lorenzo Jamone, Emre Ugur, Angelo Cangelosi, Luciano Fadiga, Alexandre Bernardino, Justus Piater, and José Santos-Victor. Affordances in psychology, neuroscience, and robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(1):4–25, 2018.

[6] B. Moldovan, P. Moreno, M. van Otterlo, J. Santos-Victor, and L. De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 4373–4378, 2012.

[7] B. Moldovan, P. Moreno, and M. van Otterlo. On the use of probabilistic relational affordance models for sequential manipulation tasks in robotics. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1290–1295, 2013.

[8] B. Moldovan and L. De Raedt. Learning relational affordance models for two-arm robots. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2916–2922, Sept 2014.

[9] A. Goncalves, G. Saponaro, L. Jamone, and A. Bernardino. Learning visual affordances of objects and tools through autonomous robot exploration. In *Autonomous Robot Systems and Competitions (ICARSC), IEEE International Conference on*, pages 128–133, 2014.

[10] T. Mar, V. Tikhanoff, G. Metta, and L. Natale. Self-supervised learning of grasp dependent tool affordances on the icub humanoid robot. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3200–3206, May 2015.

[11] A. Dehban, L. Jamone, A. R. Kampff, and J. Santos-Victor. Denoising auto-encoders for learning of objects and tools affordances in continuous space. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4866–4871, May 2016.

[12] A. Antunes, L. Jamone, G. Saponaro, A. Bernardino, and R. Ventura. From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5449–5454, May 2016.

[13] G. Saponaro, P. Vicente, A. Dehban, L. Jamone, A. Bernardino, and J. Santos-Victor. Learning at the ends: From hand to tool affordances in humanoid robots. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 331–337, Sep. 2017.

[14] D. Song, C. H. Ek, K. Huebner, and D. Kragic. Task-based robot grasp planning using probabilistic inference. *IEEE Transactions on Robotics*, 31(3):546–561, June 2015.

[15] Andreas ten Pas and Robert Platt. *Localizing Handle-Like Grasp Affordances in 3D Point Clouds*, pages 623–638. Springer International Publishing, Cham, 2016.

[16] Andreas ten Pas and Robert Platt. Using geometry to detect grasp poses in 3d point clouds. In *Robotics Research*, pages 307–324. Springer, 2018.

[17] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morena, P. Qu Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, and A. Rodriguez. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, May 2018.

[18] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016.

[19] T. Shu, X. Gao, M. S. Ryoo, and S. Zhu. Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1669–1676, May 2017.

[20] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 1529–1536, 2011.

[21] Yun Jiang and Ashutosh Saxena. Hallucinating humans for learning robotic placement of objects. In *Experimental Robotics*, pages 921–937. Springer, 2013.

[22] Yun Jiang and Ashutosh Saxena. Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs. In *Robotics: Science and Systems*, pages 1–8, 2014.

[23] L. Piyathilaka and S. Kodagoda. Affordance-map: Mapping human context in 3d scenes using cost-sensitive svm and virtual human models. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2035–2040, Dec 2015.

[24] Anirban Roy and Sinisa Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *European Conference on Computer Vision*, pages 186–201. Springer, 2016.

[25] A. Aldoma, F. Tombari, and M. Vincze. Supervised learning of hidden and non-hidden 0-order affordances and detection in real scenes. In *Robotics and Automation (ICRA), IEEE International Conference on*, pages 1732–1739, 2012.

[26] C. Desai and D. Ramanan. Predicting functional regions on objects. In *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, pages 968–975, 2013.

[27] D.I. Kim and G.S. Sukhatme. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5578–5584, May 2014.

[28] Lap-Fai Yu, Noah Duncan, and Sai-Kit Yeung. Fill and transfer: A simple physics-based approach for containability reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 711–719, 2015.

[29] A. Myers, C. L. Teo, C. Fermller, and Y. Aloimonos. Affordance detection of tool parts from geometric features. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1374–1381, May 2015.

[30] P. Abelha and F. Guerin. Learning how a tool affords by simulating 3d models from the web. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4923–4929, Sep. 2017.

[31] Safoura RezapourLakani and Justus Rodríguez-Sánchez, Antonio J.and Piater. Towards affordance detection for robot manipulation using affordance for parts and parts for affordance. *Autonomous Robots*, Jul 2018.

[32] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915, Sep. 2017.

[33] T. Do, A. Nguyen, and I. Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–5, May 2018.

[34] C. Ye, Y. Yang, R. Mao, C. Fermller, and Y. Aloimonos. What can i do around here? deep functional scene understanding for cognitive robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4604–4611, May 2017.

[35] P. Kaiser, D. Gonzalez-Aguirre, F. Schltje, J. Borrs, N. Vahrenkamp, and T. Asfour. Extracting whole-body affordances from multimodal exploration. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 1036–1043, Nov 2014.

[36] P. Kaiser, C. Mandery, A. Boltres, and T. Asfour. Affordance-based multi-contact whole-body pose sequence planning for humanoid robots in unknown environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3114–3121, May 2018.

[37] Xi Zhao, He Wang, and Taku Komura. Indexing 3d scenes using the interaction bisector surface. *ACM Trans. Graph.*, 33:1–14, 2014.

[38] Ruizhen Hu, Chenyang Zhu, Oliver van Kaick, Ligang Liu, Ariel Shamir, and Hao Zhang. Interaction context (icon): towards a geometric functionality descriptor. *ACM Transactions on Graphics (TOG)*, 34(4):83, 2015.

[39] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A.M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *Robotics Automation Magazine, IEEE*, 22(3):36–52, Sept 2015.

[40] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[41] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.