



Affordance Equivalences in Robotics: A Formalism

Mihai Andries^{1*†}, Ricardo Omar Chavez-Garcia^{2†}, Raja Chatila³, Alessandro Giusti² and Luca Maria Gambardella²

¹ Institute for Systems and Robotics (ISR-Lisboa), Instituto Superior Técnico, Lisbon, Portugal, ² Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, USI-SUPSI, Lugano, Switzerland, ³ Institut des Systèmes Intelligents et de Robotique, Sorbonne Université, Centre National de la Recherche Scientifique, Paris, France

OPEN ACCESS

Edited by:

Tadahiro Taniguchi,
Ritsumeikan University, Japan

Reviewed by:

Ashley Kleinhans,
Ford Motor Company (United States),
United States
Lola Cañamero,
University of Hertfordshire,
United Kingdom

*Correspondence:

Mihai Andries
mandries@isr.tecnico.ulisboa.pt

[†]These authors have contributed
equally to this work.

Received: 30 September 2017

Accepted: 16 May 2018

Published: 08 June 2018

Citation:

Andries M, Chavez-Garcia RO,
Chatila R, Giusti A and
Gambardella LM (2018) Affordance
Equivalences in Robotics: A
Formalism. *Front. Neurobot.* 12:26.
doi: 10.3389/fnbot.2018.00026

Automatic knowledge grounding is still an open problem in cognitive robotics. Recent research in developmental robotics suggests that a robot's interaction with its environment is a valuable source for collecting such knowledge about the effects of robot's actions. A useful concept for this process is that of an affordance, defined as a relationship between an actor, an action performed by this actor, an object on which the action is performed, and the resulting effect. This paper proposes a formalism for defining and identifying affordance equivalence. By comparing the elements of two affordances, we can identify equivalences between affordances, and thus acquire grounded knowledge for the robot. This is useful when changes occur in the set of actions or objects available to the robot, allowing to find alternative paths to reach goals. In the experimental validation phase we verify if the recorded interaction data is coherent with the identified affordance equivalences. This is done by querying a Bayesian Network that serves as container for the collected interaction data, and verifying that both affordances considered equivalent yield the same effect with a high probability.

Keywords: affordance, learning, cognitive robotics, symbol grounding, affordance equivalence

1. INTRODUCTION

Symbolic grounding of robot knowledge consists in creating relationships between the symbolic concepts used by algorithms controlling the robot and the physical concepts to which they correspond (Harnad, 1990). An *affordance* is a concept that allows collection of grounded knowledge. The notion of *affordance* was introduced by Gibson (1977), and refers to the action opportunities provided by the environment. In the context of robotics, an affordance is a relationship between an actor (i.e., robot), an action performed by the actor, an object on which this action is performed, and the observed effect.

A robot able to discover and learn the affordances of an environment can autonomously adapt to it. Moreover, a robot that can detect equivalences between affordances can quickly compute alternative plans for reaching a desired goal, which is useful when some actions or objects suddenly become unavailable.

In this paper, we introduce a method for identifying affordances that generate equivalent effects (see examples in **Figures 1, 2**). We define a (comparison) operator that allows robots to identify equivalence relationships between affordances by analysing their constituent elements (i.e., actors, objects, actions).

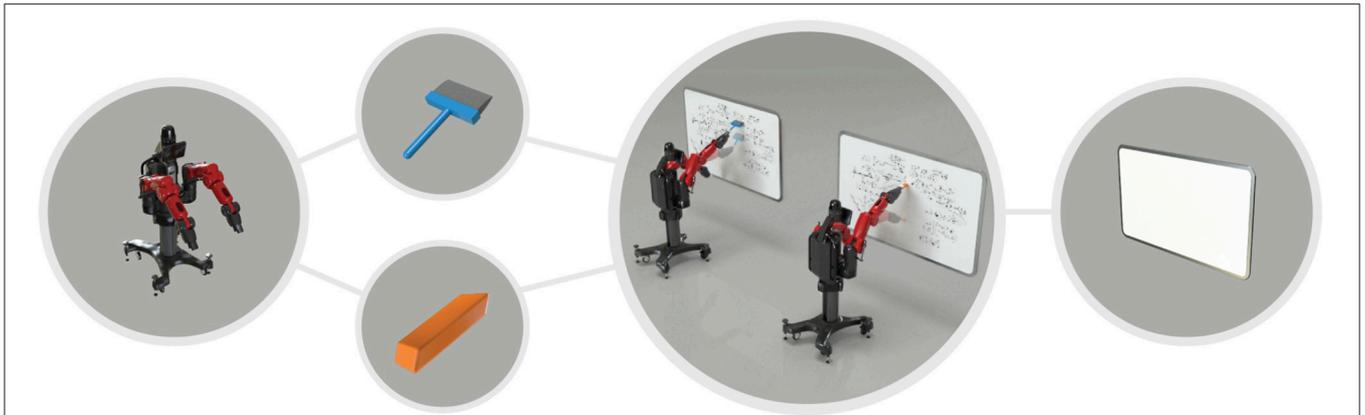


FIGURE 1 | Example of equivalence between two objects for cleaning a whiteboard: a wiper and an eraser. The robot affords to clean the white board by wiping it either with a wiper or an eraser.



FIGURE 2 | Example of equivalence between different actors and their actions for opening a door. A door can be opened by any robot that can interact with the door.

1.1. Affordance Discovery and Learning

All methods proposed in the literature for affordance learning are similar in viewing an interaction as being composed of three components: an action, a target object, and a resulting effect. Different methods were proposed to infer the expected effect, given knowledge about the action and target object.

Several papers approached affordance learning as learning to predict object motion after interaction. For this purpose, Krüger et al. (2011) employed a feedforward neural network with backpropagation which learned so-called *object-action complexes*; Hermans et al. (2013) used Support Vector Machines (SVM) with kernels; while Kopicki et al. (2017) employed Locally Weighted Projection Regression (LWPR) with Kernel Density Estimation and a mixture of experts. Ridge et al. (2009) first used a Self-Organising Map and clustering in the *effect space* to classify objects by their effect, and then trained a SVM which

identified to which cluster an object belongs using its feature-vector description.

Other papers addressed affordance learning from the perspective of object grasping. Stoytchev (2005) employed detection of invariants to learn object grasping affordances. Ugur et al. (2012) used SVMs to study the traversability affordance of a robot for grasping. Katz et al. (2014) used linear SVM to learn to perceive object affordances for autonomous pile manipulation. More details on the use of affordances for object manipulation can be found in the dissertation of Hermans (2014).

Some works followed a supervised training approach, providing hand-labeled datasets which mapped objects images (2D or RGB-D) to their affordances. Myers et al. (2015) learned affordances from local shape and geometry primitives using Superpixel-based Hierarchical Matching Pursuit (S-HMP), and Structured Random Forests (SRF). Image regions (from RGB-D

frames) with pre-selected properties were tagged with specific affordance labels. For instance, a surface region with high convexity was labeled as *containable* (or a variation of it). Varadarajan and Vincze (2012) proposed an Affordance Network for providing affordance knowledge ontologies for common household articles, intended to be used for object recognition and manipulation. An overview of machine learning approaches for detecting affordances of tools in 3D visual data is available in the thesis of Ciocodeica (2016).

Another approach for learning affordances uses Bayesian Networks. Montesano et al. (2008) and Moldovan et al. (2012) employed a graphical model approach for learning affordances, using a Bayesian Network which represents objects/actions/effects as random variables, and which encodes relations between them as dependency links. The structure of this network is learned based on the data of robot's interaction with the world and on *a priori* information related to the dependency of some variables. Once learned, affordances encoded in this way can (1) predict the effect of an action applied to a given object, (2) infer which action on a given object generated an observed effect, and (3) identify which object generates the desired effect when given a specific action.

Yet another popular method for supervised affordance learning uses Deep Learning techniques. For instance, Nguyen et al. (2016) trained a convolutional neural network to identify object affordances in RGB-D images, employing a dataset of object images labeled pixelwise with their corresponding affordances. A similar approach using a deep convolutional neural network was taken by Srikantha and Gall (2016).

Recent comprehensive overviews of affordance learning techniques are available in the dissertation of Moldovan (2015), and in reviews by Jamone et al. (2016), Min et al. (2016), and Zech et al. (2017).

We argue that once affordances are learned, we can find relations between affordances by considering the effects they generate. One of these relations is equivalence, i.e., when two different affordances specify corresponding actions on objects that generate the same effect.

1.2. Affordance Equivalence

Affordance equivalence was studied by Şahin et al. (2007), who considered relationships between single elements of an affordance. Thus, it was possible to identify objects or actions that are equivalent with respect to an affordance when they generate the same effect. Griffith et al. (2012) employed clustering to identify classes of objects that have similar functional properties. Montesano et al. (2008) and Jain and Inamura (2013) treated affordance equivalence from a probabilistic point of view, where, in the context of imitation learning, the robot searches for the combination of action and effect that maximises their similarity to the demonstrated action on an object. Boularias et al. (2015) discovered through reinforcement learning the graspability affordance over objects with different shapes, and indirectly showed equivalence of the grasp action.

Developing this line of thought, we propose a probabilistic method to identify which *combinations of affordance elements* generate equivalent effects. We first present in section 2 the

affordance formalization employed, and based on that we then list in section 2.4 all the possible types of affordance equivalences.

Since the purpose of this study is to identify equivalences between affordances that were already recorded by the robot, we are not seeking to explain how to record these affordances. In this paper we employed the graphical model approach for learning affordances proposed by Montesano et al. (2008). In addition, we rely purely on perception-interaction data, without using *a priori* information (Chavez-Garcia et al., 2016b). To facilitate the experimental setup, we used pre-defined sensorial and motor capabilities for our robots.

The remainder of this paper is organized as follows. In section 2, we introduce our formalization of affordance elements, and define the equivalence relationship in section 2.4. A series of experiments on the discovery of equivalences between affordances is detailed in section 3, together with the obtained results. We conclude and present opportunities for future work in section 4.

2. METHODOLOGY: AFFORDANCE FORMALIZATION

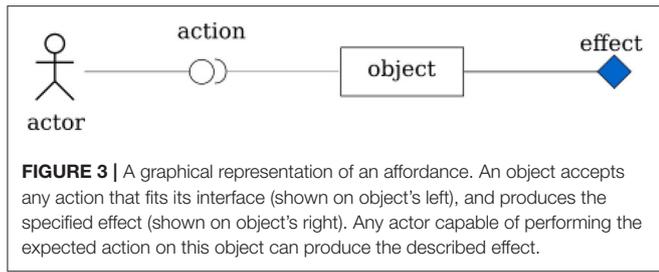
In this section, we present the affordance formalism employed throughout the paper. We follow the definition proposed by Ugur et al. (2011), that we enrich by including the actor performing the action into the affordance tuple (*object, action, effect*). The inclusion of the actor into the affordance allows robots to record affordances specific to their body morphologies. Although we will not focus on this aspect in this paper, it is possible to generalize this knowledge through a change of affordance perspective from robot joint space to object task space (more about this in section 2.1.2).

We define an affordance as follows. Let G be the set of actors in the environment, O the set of objects, A the set of actions, and E the set of observable effects. Hence, when an *actor* applies an *action* on an *object*, generating an *effect*, the corresponding affordance is defined as a tuple:

$$\alpha = (actor, object, action, effect), \text{ for } actor \in G, object \in O, \\ action \in A, \text{ and } effect \in E, \quad (1)$$

and can be graphically represented as shown in **Figure 3**. From actor perspective, it interacts with the environment (the object) and discovers the affordances. From object perspective, affordances are properties of objects which can be perceived by actors, and which are available to actors with specific capabilities. We can also consider observers, who learn by perceiving other actors' affordance acquisition process.

The way in which affordance elements are defined influences the operations that can be performed with affordances. Since we aim to establish equivalence relationships between affordances, we will analyse the definitions of the following affordance elements: actions (from actor and object perspectives), objects (as perceived by robot's feature detectors), and effects (seen as a description of the environment).



2.1. How Are Actions Defined?

Actions can be defined (1) relative to actors, by describing the body control sequence during the execution of an action in joint space; or (2) relative to objects, by describing the consequences of actions on the objects in operational space. We refer to *object perspective* when the actions are defined in the operational/task space, making their definition independent of the actor executing them. We refer to *actor perspective* when the actions are defined in the joint space of the actor, making them dependent of the actor executing them.

This statement comes from the different perspectives obtained from the affordance definition in Equation (1): *actor* and *object* perspective.

2.1.1. Actions Described Relative to Actors

Actions are here described relative to actors and their morphology. They are defined with respect to their control variables in joint space (i.e., velocity, acceleration, jerk), indexed by time τ :

$$\text{action} : \{Q, \dot{Q}, \ddot{Q}\}_\tau \quad (2)$$

As the action is described with respect to the actor morphology and capabilities, comparing two actions requires comparing both the actors performing the actions, and the actions themselves. When the actors are identical, the action comparison is straightforward. However, when there is a difference between actors' morphologies (and their motor capabilities), the straightforward comparison of actions is not possible and a common frame of reference for such comparison is needed.

2.1.2. Actions Described Relative to Objects

When actions are described relative to objects, they represent an action generalisation from the *joint space* of a particular actor (where actions are defined on the actor) to the *operational space* of any actor (where actions are defined on the object).

Thus, when actions are described relative to objects, the *actor* can be omitted from the affordance tuple, to indicate that any actor which has the required motor capabilities is able to generate the action which causes this effect. In addition, the action employed in this representation is defined in *operational space* (and not in *joint space* as before). Hence, dropping the actor from the equation, we can rewrite Equation (1) as:

$$\alpha = (\text{object}, \text{action}, \text{effect}), \text{ for } \text{object} \in O, \text{action} \in A_o, \text{ and } \text{effect} \in E \quad (3)$$

where A_o is the set of all actions in operational space, applicable to object o .

While affordances defined from actor perspective (in joint space, e.g., joint forces to apply) allow to learn using robot's motor and perceptual capabilities, affordances defined from object perspective (in task space, e.g., forces applied on the object) allow to generalise this knowledge.

2.2. How Are Objects Defined?

If an actor has the feature detectors p_1, \dots, p_n corresponding to its perception capacities (such as hue, shape, size), then an object is defined as:

$$\text{object} = \{p_1, \dots, p_n\}, \quad (4)$$

where each feature detector can be seen as function on a perceptual unit (e.g., a salient segment from a visual perception process).

2.3. How Are Effects Defined?

We suppose that an actor g has a set ξ of q effect detectors, that are able to detect changes in the world after an action $a \in A_g$ is applied. For example, when an actor executes action *push* on an object, the object-displacement-effect detector would be a function that computes the difference between two measurements of the object position taken before and after the interaction. Another effect can be the difference in the feedback force measured in the end effector before and after the interaction. Formally, effects are a set of q salient changes in the world ω (i.e., in the target object, the actor, or the environment), detected by robot's effect detectors ξ :

$$\text{effect} = \{\xi_1(\omega), \dots, \xi_q(\omega)\} \quad (5)$$

2.4. Affordance Equivalence Operator

In this section, we introduce the concept of affordance equivalence, based on the formalization presented earlier in section 2. We provide truth tables for two different affordance comparison operators: one for the case where actions are defined in actor joint space, and one for the case where actions are defined in object task space. For each case, we explore the possible types of affordance equivalence.

We have defined an affordance as a tuple of type $(\text{actor}, \text{object}, \text{action}_{\text{joint_space}}, \text{effect})$ when the action is defined relative to the actor, or as a tuple of type $(\text{object}, \text{action}_{\text{operational_space}}, \text{effect})$ when the action is defined relative to the object. Let us now define the truth table for an operator for comparing affordances (one for the actor perspective, and one for the object perspective) and identifying equivalence relationships between them.

We consider equivalent two affordances that generate equivalent effects. To know when two effects are equivalent, an effect-comparison function is required. We define an equivalence function $f(e_a, e_b)$ that yields true if two effects values e_a and e_b are similar in a common frame (e.g., distances for position values, similarity in color models, vector distances for force values). We detect affordance equivalence by (1) feeding the continuous (non-discretised) data on the measured effects to the Bayesian

Network (BN) structure learning algorithm, and then (2) querying the BN over an observed effect to obtain the empirical decision on effect equivalence. Whenever two affordances generate equivalent effects, it is possible to find which affordance elements cause this equivalence. We distinguish several cases of affordance equivalence, depending on the elements which differ in two equivalent affordances, which are detailed below.

2.4.1. Equivalence Between Affordance With Actions Defined Relative to Actors

The comparison cases for affordances with actions described relative to actors are shown in **Table 1**. The 2^4 cases of comparison between the elements of two affordances stem from all the possible (binary) equivalence combinations between the elements. In each case we compare the four components and establish if the elements of affordances are equivalent.

Since actions are defined here relative to the actors, actors with different morphologies cannot perform the same action defined in joint space, because their joint spaces are different. This renders inconsistent cases in which different actors perform the same action: lines (3), (4), (7), and (8) in **Table 1**. This leaves us with five cases of equivalence in **Table 1**, where:

- If different actors using different actions on different objects generate an equivalent effect, then we have (*actor, action, object*) equivalence
- If different actors using different actions on the same object generate an equivalent effect, then we have (*actor, action*) equivalence
- If the same actor using different actions on different objects generates an equivalent effect, then we have (*object, action*) equivalence
- If the same actor using the same action on different objects generates an equivalent effect, then we have *object* equivalence
- If the same actor using different actions on the same object generates an equivalent effect, then we have *action* equivalence.

We assume that the environment is a deterministic system: each time the same actor applies the same action on the same object, it will generate an equivalent effect. Therefore, generating a different effect with the same actor, action, and object is impossible, due to determinism.

Both the effect equivalence and non-equivalence cases provide information about the relationship between two affordances. The affordance equivalence concept is empirically validated in section 3.

2.4.2. Equivalence Between Affordances With Actions Defined Relative to Objects

The comparison cases for affordances with actions described relative to objects are shown in **Table 2**. There are 2^3 cases of comparison, corresponding to the total number of possible (binary) equivalence cases between the elements of a pair of affordances. In this case, three types of equivalence exist:

- If different actions on different objects generate the same effect, then it is (*object, action*) equivalence;

- If same action on different objects generates the same effect, then it is *object* equivalence;
- If different actions on same object generate the same effect, then it is *action* equivalence.

3. EXPERIMENTS AND RESULTS: AFFORDANCE EQUIVALENCE

We designed experiments that would confirm the capability of our affordance representation to detect equivalences and non-equivalences between learned affordances. We employed a Bayesian Network structure-learning approach presented in (Chavez-Garcia et al., 2016a) to describe and learn affordances as relations between random variables (affordance elements). Then we analyse how the learned affordances relate to each case of equivalence presented in **Table 2**.

3.1. Pre-defined Actions

We assume that an agent is equipped, since its conception, with motor and perceptual capabilities that we called *pre-defined*. However, we do not limit the agent's capabilities to the pre-defined set, as through learning the agent may acquire new capabilities. In our scenario, we employed three robotic actors of different morphologies, each with its pre-defined actions:

1. Baxter_{gripper}: the Baxter robot's left arm (7 DOF) equipped with a gripper, with actions:
 - Push (moving with constant velocity without closing the gripper)
 - Pull (closing the gripper and moving with constant velocity)
 - Wipe (closing the gripper and pressing downwards while moving)
 - Move aside (closing the gripper and moving aside)
2. Baxter_{nogripper}: the Baxter robot's right arm with no gripper, with action:
 - Poke (moving forwards with constant acceleration)
3. Katana arm with no gripper (5 d.o.f.), with action:
 - Side push (moving aside with constant velocity)

The actors and their pre-defined sets of actions (motor capabilities) are shown in **Figure 4**.

3.2. Pre-defined Perceptual Capabilities

Our visual perception process takes raw RGB-D data of an observed scene to oversegment the point cloud into a supervoxel representation. This 3D oversegmentation technique is based on a flow-constrained local iterative clustering which uses color and geometric features from the point cloud (Papon et al., 2013). Strict partial connectivity between voxels guarantees that supervoxels cannot flow across disjoint boundaries in 3D space. Supervoxels are then grouped to obtain object clusters that are used for extracting features and manipulation. **Figure 5** illustrates the visual perception process. The objects employed were objects of daily use: toys that can

TABLE 1 | Comparison of two affordances, when actions are described with respect to actors.

#	Actors	Objects	Actions	Effects	Conclusion
1	different	different	different	different	<i>(actor, object, action)</i> non-equivalence
2	different	different	different	equivalent	(actor, object, action) equivalence
3	<u>different</u>	different	<u>same</u>	different	<i>(actor, object)</i> non-equivalence
4	<u>different</u>	different	<u>same</u>	equivalent	(actor, object) equivalence
5	different	same	different	different	<i>(actor, action)</i> non-equivalence
6	different	same	different	equivalent	(actor, action) equivalence
7	<u>different</u>	same	<u>same</u>	different	<i>actor</i> non-equivalence
8	<u>different</u>	same	<u>same</u>	equivalent	actor equivalence
9	same	different	different	different	<i>(object, action)</i> non-equivalence
10	same	different	different	equivalent	(object, action) equivalence
11	same	different	same	different	<i>object</i> non-equivalence
12	same	different	same	equivalent	object equivalence
13	same	same	different	different	<i>action</i> non-equivalence
14	same	same	different	equivalent	action equivalence
15	same	same	same	different	impossible in deterministic systems
16	same	same	same	equivalent	due to determinism

Equivalence cases between affordances are presented in even rows. Inconsistencies are underlined in red. The types of affordance equivalence are shown in bold letters.

TABLE 2 | Comparison of two affordances, when actions are described with respect to objects.

#	Objects	Actions	Effects	Conclusion
1	different	different	different	<i>(object, action)</i> non-equivalence
2	different	different	equivalent	(object, action) equivalence
3	different	same	different	<i>object</i> non-equivalence
4	different	same	equivalent	object equivalence
5	same	different	different	<i>action</i> non-equivalence
6	same	different	equivalent	action equivalence
7	same	same	different	impossible in deterministic systems
8	same	same	equivalent	due to determinism

Equivalence cases between affordances are presented in even rows. The types of affordance equivalence are shown in bold letters.

be assembled, markers, and dusters. The objects were selected so as to be large enough to allow easy segmentation and manipulation.

3.3. Pre-defined Effect Detectors

We used custom hand-written effect detectors for the experimental use-cases, although our experimental architecture allows for an automatic effect detector. An effect detector quantifies the change, if present, in one property of the environment or the actor. For this series of experiments, we developed the following effect detectors: color change in a 2D image (HSV hue) for an object or a region of interest; object's position change (translation only); and the end-effector position. **Figure 6** illustrates the detected effects when *wipe* action is performed. In our previous work we covered changes in joint torques, distance between finger grippers and object speed.

3.4. Affordance Learning

Affordance elements E (effects), O (objects) and A (actions) are represented as random variables of a Bayesian Network (BN) \mathcal{B} . First, in each actor interaction we record the values (discretized) for the random variables representing the objects (section 3.2), actions (section 3.1), and effects (section 3.3). The problem of discovering the relations between E , O , and A can be then translated to finding dependencies between the variables in \mathcal{B} , i.e., $P(\mathcal{B}|\mathcal{D})$ learning the structure of the corresponding network \mathcal{B} from data \mathcal{D} . Thus, affordances are described by the conditional dependencies between variables in \mathcal{B} .

We implemented an information-compression score to estimate how well a Bayesian Network structure describes data \mathcal{D} (Chavez-Garcia et al., 2016b). Our score is based on the Minimum Description Length (MDL) score:

$$MDL(\mathcal{B}|\mathcal{D}) = LL(\mathcal{B}|\mathcal{D}) - |\mathcal{B}| \frac{\log N}{2}, \quad (6)$$

where the first term measures (by applying a log-likelihood score Suzuki, 2017) how many bits are needed to describe data \mathcal{D} based on the probability distribution $P(\mathcal{B})$. The second term counts the number of bits needed to encode \mathcal{B} , where $\frac{\log(N)}{2}$ bits are used for each parameter in the BN. We consider $\frac{\log(N)}{2}$ as factor that penalizes structures with larger number of parameters. For a BN's structure \mathcal{B} , its score is then defined as the posterior probability given the data \mathcal{D} .

We implemented a search-based structure learning algorithm based on the hill-climbing technique, as we did in our previous work. As inputs, this algorithm takes values for the variables in E , O , and A obtained from robot's interaction. This procedure estimates the parameters of the local probability density functions (pdfs) given a Bayesian Network structure. Typically, this is a

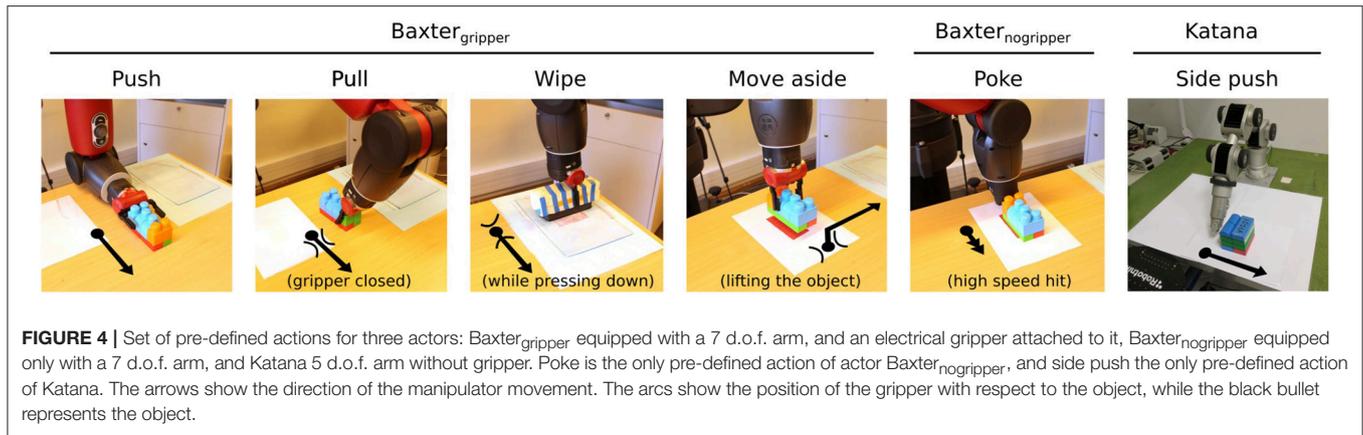


FIGURE 4 | Set of pre-defined actions for three actors: *Baxter_{gripper}* equipped with a 7 d.o.f. arm, and an electrical gripper attached to it, *Baxter_{nogripper}* equipped only with a 7 d.o.f. arm, and *Katana* 5 d.o.f. arm without gripper. Poke is the only pre-defined action of actor *Baxter_{nogripper}*, and side push the only pre-defined action of *Katana*. The arrows show the direction of the manipulator movement. The arcs show the position of the gripper with respect to the object, while the black bullet represents the object.

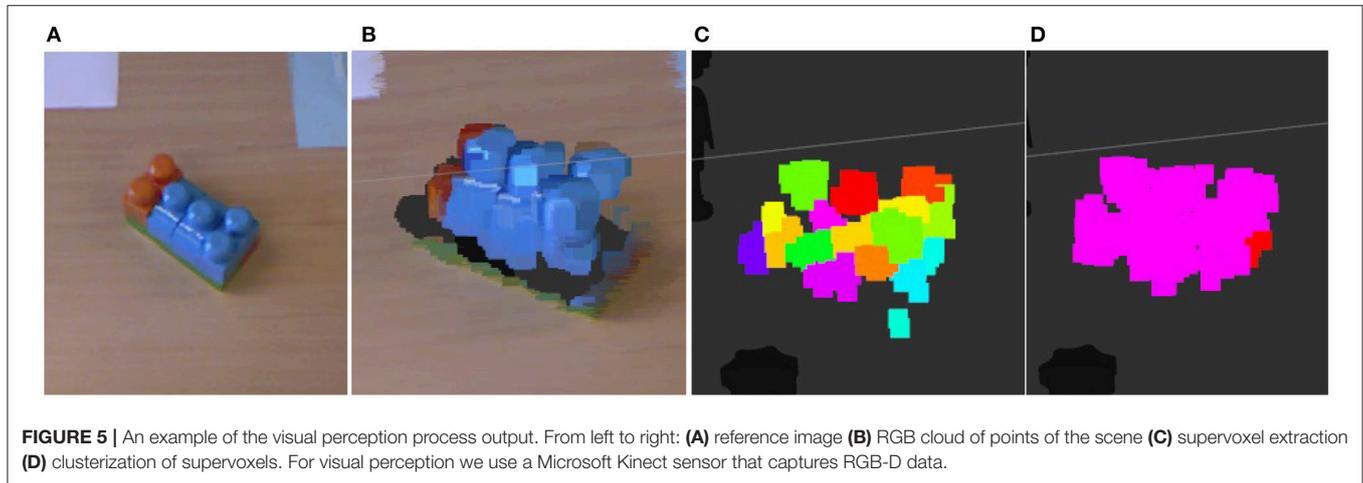


FIGURE 5 | An example of the visual perception process output. From left to right: (A) reference image (B) RGB cloud of points of the scene (C) supervoxel extraction (D) clusterization of supervoxels. For visual perception we use a Microsoft Kinect sensor that captures RGB-D data.

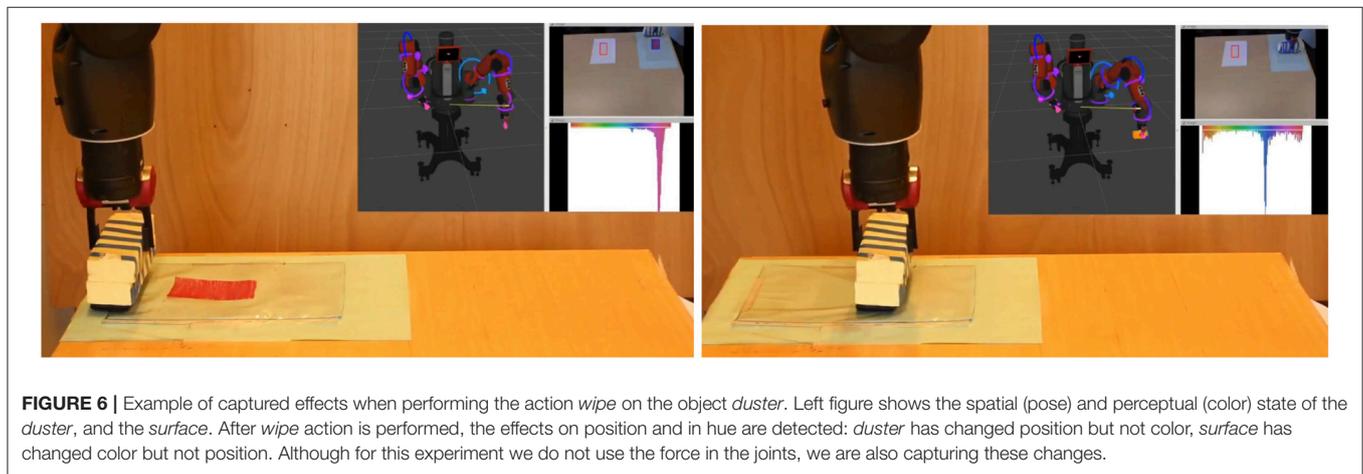


FIGURE 6 | Example of captured effects when performing the action *wipe* on the object *duster*. Left figure shows the spatial (pose) and perceptual (color) state of the *duster*, and the *surface*. After *wipe* action is performed, the effects on position and in hue are detected: *duster* has changed position but not color, *surface* has changed color but not position. Although for this experiment we do not use the force in the joints, we are also capturing these changes.

maximum-likelihood estimation of the probability entries from the data set, which, for multinomial local pdfs, consists of counting the number of tuples that fall into each table entry of each multinomial probability table in the BN. The algorithm’s main loop consists of attempting every possible single-edge addition, removal, or reversal, making the network that increases the score the most the current candidate, and iterating. The

process stops when there is no single-edge change that increases the score. There is no guarantee that this algorithm will settle at a global maximum, but there are techniques to increase its reaching possibilities (we use simulated annealing).

By using the BN framework, we are capable of displaying relationships between affordance elements. The directed nature of its structure allows us to approximate cause-effects

relationships. It also handles uncertainty through the established probability theory. In addition to direct dependencies, we can represent indirect causation.

3.4.1. Detection of Affordance Equivalence

Equivalence between two affordances can be identified by comparing their ability to consistently reproduce the same effect e , judging by the cumulated experimental evidence. The precise type of equivalence between two affordances, which tells which affordance elements' values are equivalent, can be identified by probabilistic inference on the learned BN. Inference allows to identify which (*actor*, *object*, *action*) configurations are more likely to generate the same effect. In practice, this inference is calculated through executing queries to the Bayesian Network, which allow to compute the probability of an event (in our case: the probability of an effect having a value between some given bounds) given the provided evidence data.

Queries have the following form: $P(\textit{proposition}|\textit{evidence})$ where *proposition* represents the query on some variable x , and *evidence* represents the available information for the affordance elements, e.g., the identity of the actor, the description of the action, and the description of the object. In the example of the robot pushing an object, the following query allows to compute the probability of the object displacement falling between certain bounds:

$$P((\textit{position} > \textit{lower bound}) \textit{ and } (\textit{position} < \textit{upper bound}) | \textit{actor} = \textit{Baxter}, \textit{action} = \textit{push}, \textit{object} = \textit{block}) \quad (7)$$

After querying the learned BN with the corresponding elements from **Tables 1, 2** as evidence, if two (*actor*, *object*, *action*) configurations have probabilities of generating an effect that are higher than an arbitrary threshold, then we consider both affordances equivalent:

$$\begin{aligned} &\text{if } P(e|\textit{actor}_1, \textit{object}_1, \textit{action}_1) > \theta \\ &\text{and } P(e|\textit{actor}_2, \textit{object}_2, \textit{action}_2) > \theta \\ &\text{then } (\textit{actor}_1, \textit{object}_1, \textit{action}_1) \equiv (\textit{actor}_2, \textit{object}_2, \textit{action}_2) \end{aligned} \quad (8)$$

For our experiments, we empirically established the equivalence threshold $\theta = 0.85$. The aforementioned querying process connects the learning and reasoning steps, and according to the current goal of an actor, it allows for an empirical threshold selection or an adaptive mechanism.

3.5. Experimental Results

As shown in **Table 1**, affordances composed of 4 elements (*actor*, *object*, *action*, *effect*), which have their actions defined from the actor perspective, have five cases of equivalence (see **Figure 7** for some illustrated examples). We have selected three of them to demonstrate the use of the affordance equivalence operator: (*object*) equivalence, (*action*) equivalence, and (*actor*, *action*) equivalence. In **Figure 7** they correspond to the settings (a), (b), and (c). These experiments are detailed below. For a video demonstration of these experiments, please see the Supplementary Material section at the end of this document.

3.5.1. The (Actor, Action) Equivalence

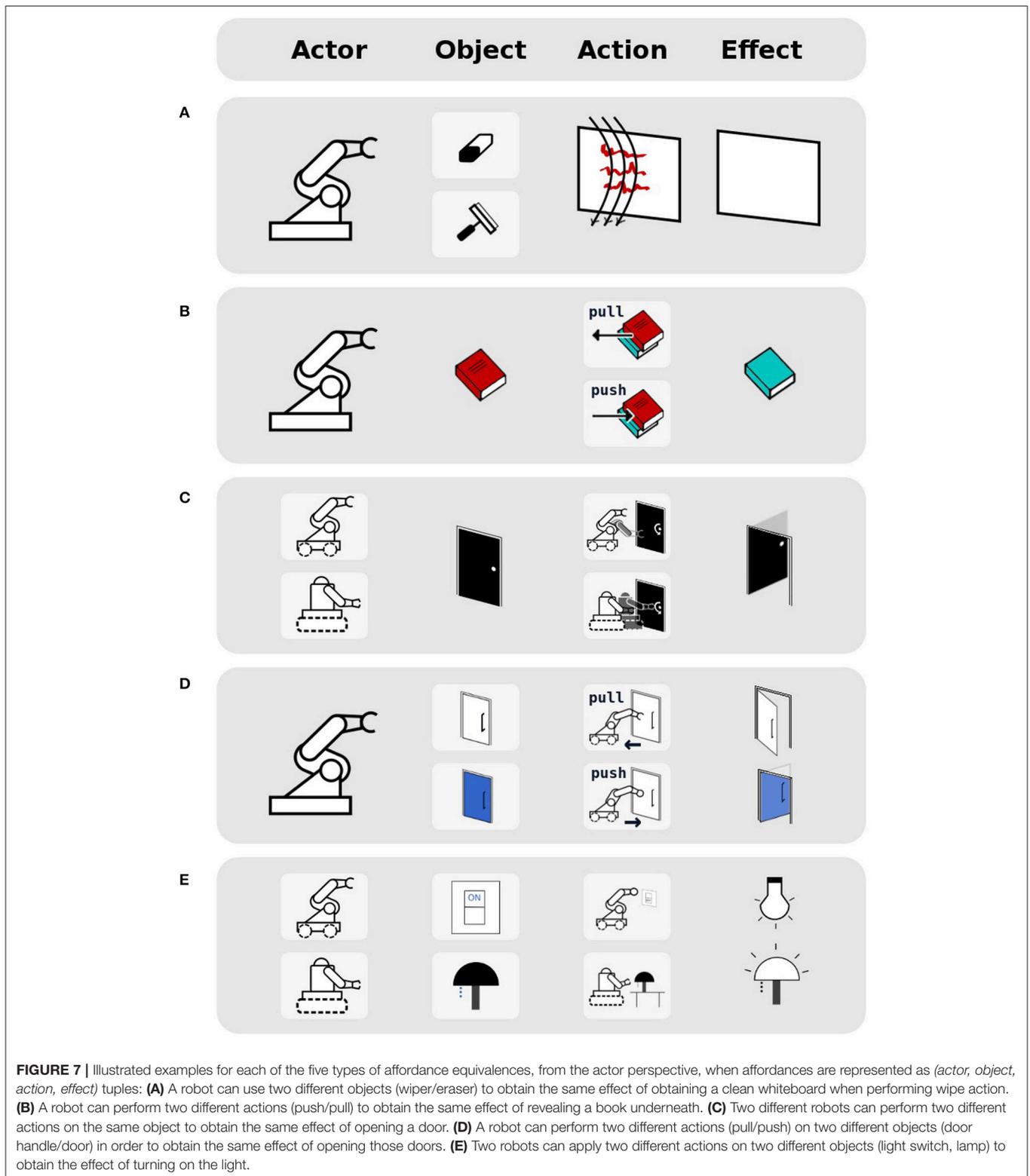
This experiment consisted in discovering the equivalence between (*actor*, *action*) tuples. The goal was to identify configurations that are equivalent in their ability of uncovering a region of interest (a red mark on the table) by moving the object occluding it from robot's camera view (in the case of the Baxter — a toy with features color: blue and shape: box; in the case of the Katana actor — a box with the same perceptual features). In our representation, two objects with the same perceptual features are considered the same. Actor Baxter_{gripper} is equipped with a gripper and can perform action *move_aside*. Actor Baxter_{nogripper} does not have a gripper and can only perform action *poke*. Actor Katana does not have a gripper and can only perform *side_push* action.

The Bayesian Network structure was learned using data from 15 interactions using each (*actor*, *action*) tuple (**Figure 8**). Variables *object_shape* and *object_color* represent the object features, variable *color_mark* captures the presence or absence of a colored mark. Queries performed on the BN suggested that the effect of revealing the red mark is consistently recreated when moving the object toy, with a probability of 0.98 for the action *move_aside* done by the hand with a gripper, 0.97 for the action *poke* done by the hand with no gripper, and 0.94 for the action *side_push* done by the Katana arm on the box object. The probabilities are based on the total number of trials verifying these relationships. Since these affordances consistently recreate equivalent effects while having some equal elements (same toy object for Baxter_{gripper} and Baxter_{nogripper}, and a similar object for Katana), this points that affordance elements that differ between configurations are in fact equivalent in their ability to generate the effect of revealing the red mark, i.e., the tuples (Baxter_{nogripper}, *poke*), (Baxter_{gripper}, *move_aside*) and (Katana, *side_push*). Source code of the experimental setup for the Katana actor is available at https://romarcg@bitbucket.org/romarcg/katana_docker.git.

3.5.2. The (Object) Equivalence

The experiment consisted in determining the equivalence between two visually different whiteboard dusters: *duster_{blue}* and *duster_{orange}*. Actor Baxter_{gripper} applies the same action *wipe* to remove a red marker trace from a blue colored surface, as shown in **Figures 4, 6**. For distinguishing the clean blue colored surface from the surface dirtied with the red marker, the robot's pre-defined effect detector measured the effect on the hue extracted from an HSV histogram.

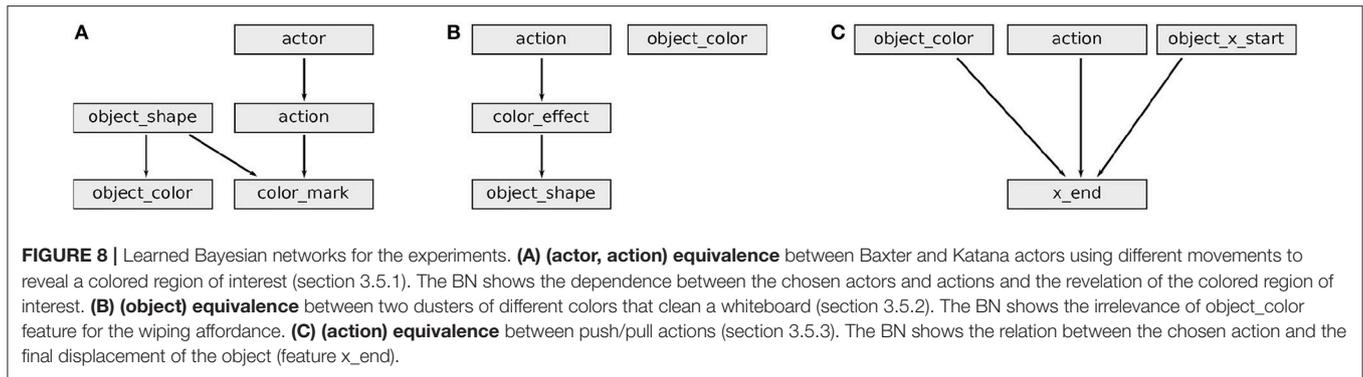
The robot performed 25 trials of the *wipe* action with each duster, and the obtained data was subsequently used to learn the Bayesian Network structure (see **Figure 8B**). Objects are represented in the same way as in section 3.5.1. The effect capturing the change in the wiped area is described by the variable *color_effect*. Queries revealed that the *wipe* action cleans the red marker trace from the blue colored surface with a probability of 0.95 in both cases. Since the observed effects were equivalent, and the actor and action were the same, the objects *duster_{blue}* and *duster_{orange}* are considered equivalent in their ability to reach this effect.



3.5.3. The (Action) Equivalence

In this experiment we analysed equivalence between the actions of an actor. This experiment consisted in placing the same object *toy* into a desired location using two different actions *push* and

pull of the actor *Baxter_{gripper}*. The robot performed 30 trials using each of the push and pull actions. **Figure 8C** shows the learned BN for (action) equivalence. The arrival of the object (described as in previous experiments) to the desired position is described



by the effect variable x_{end} (only the x component of the 3D position was measured). The target location to which we aim to push/pull the object is at x coordinate $0.72 \pm 0.02m$. Variable $object_x_start$ is an object feature representing the object initial position. According to the BN that processed the obtained data, there was a 0.97 probability to *pull* the object to the desired location, and a 0.89 chance to do so by *pushing* it. With all the rest being equal (the actor, object, and effect are the same), and since both actions have a high probability of generating the given effect, these *push* and *pull* actions can be considered equivalent for placing the object *toy* in a desired location.

4. CONCLUSIONS AND FUTURE WORK

We have presented a formalization for affordances with respect to their elements, and the equivalence operator for comparing two affordances from the actor and object perspective. We performed Bayesian Network structure learning to capture affordances as sensorimotor representations based on the observed experimental data. We analysed and validated experimentally the affordance equivalence operator, demonstrating how to extract information on the tuples of actors, actions and objects by comparing two affordances and determining if such tuples are equivalent.

In practice, the learned affordance equivalences can be interchangeably used when some objects or actions become unavailable. In a multi-robot setting, these equivalences can allow an ambient intelligence (an Artificial Intelligence system controlling an environment) to select the appropriate robot for using an affordance to reach a desired effect.

4.1. Future Work

Our future work will focus on the domain of transfer learning. We plan to implement a transformation between the affordances learned by specific robots (in their own joint space) to affordances applicable to objects and defined in their operational space. This will generalise the affordances learned and perceivable by a robot with a specific body schema, making them perceivable (and potentially available) to robots with any type of body schema (morphology).

We are already working on an automatic method for generating 3D object-descriptors. This would allow us to remove human bias from the way in which the robot observes and

analyses its environment. By using an auto-encoder (a type of artificial neural network) that trains on appropriate datasets, it can automatically adapt to changes in objects that the robot interacts with.

Work is also underway on representing robot actions in a continuous space (e.g., using a vector representation of torque forces, or Dynamic Movement Primitives), which would be an improvement from today's discrete representation of actions (e.g., move, push, pull).

Ultimately, we intend to define an *algebra of affordances* detailing all the operations that are possible on affordances, and which would encompass operators such as affordance equivalence, affordance chaining (Ugur et al., 2011), and other operators that are still to explore.

AUTHOR CONTRIBUTIONS

Literature review by MA and RC-G. Methodology and theoretical developments by MA, RC-G, and RC. Experiment design and implementation by MA and RC-G. Analysis of the experimental results by MA, RC-G, RC, AG, and LG. Document writing and illustrations by MA, RC-G, RC, AG, and LG.

ACKNOWLEDGMENTS

This work was funded by ANR RoboErgoSum project under reference ANR-12-CORD-0030, and by Laboratory of Excellence SMART (ANR-11-LABX-65) supported by French State funds managed by the ANR - Investissements d'Avenir programme ANR-11-IDEX-0004-02.

We kindly thank Hugo Simão for his help with the 3D renderings used for illustrating this work. Credit for the 3D models of the Baxter robot used in **Figures 1, 2, 6** goes to Rethink Robotics. Credit for the robot model used in the bottom part of **Figure 2** goes to Dushyant Chourasia (<https://grabcad.com/library/robot-242>).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fnbot.2018.00026/full#supplementary-material>

REFERENCES

- Boularias, A., Bagnell, J. A., and Stentz, A. (2015). "Learning to manipulate unknown objects in clutter by reinforcement," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Learning* (Austin, TX), 1336–1342.
- Chavez-Garcia, R. O., Andries, M., Luce-Vayrac, P., and Chatila, R. (2016a). "Discovering and manipulating affordances," in *International Symposium on Experimental Robotics (ISER)* (Tokyo).
- Chavez-Garcia, R. O., Luce-Vayrac, P., and Chatila, R. (2016b). "Discovering affordances through perception and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon).
- Ciocodeica, S. (2016). *A Machine Learning Approach for Affordance Detection of Tools in 3D Visual Data*. Bachelor's thesis, University of Aberdeen.
- Gibson, J. (1977). "The theory of affordances," in *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, eds R. Shaw and J. Bransford (Hoboken, NJ: John Wiley & Sons Inc.), 67–82.
- Griffith, S., Sinapov, J., Sukhoy, V., and Stoytchev, A. (2012). "A behavior-grounded approach to forming object categories: separating containers from noncontainers," in *IEEE Transactions on Autonomous Mental Development*, 54–69.
- Harnad, S. (1990). The symbol grounding problem. *Phys. D Nonlinear Phenomena* 42, 335–346.
- Hermans, T., Li, F., Reh, J. M., and Bobick, A. F. (2013). "Learning contact locations for pushing and orienting unknown objects," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (Atlanta, GA), 435–442.
- Hermans, T. R. (2014). *Representing and Learning Affordance-Based Behaviors*. Ph.D. thesis, Georgia Institute of Technology.
- Jain, R., and Inamura, T. (2013). Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools. *Artif. Life Robot.* 18, 95–103. doi: 10.1007/s10015-013-0105-1
- Jamone, L., Ugur, E., Cangelosi, A., Fadiga, L., Bernardino, A., Piater, J., et al., (2016). "Affordances in psychology, neuroscience and robotics: a survey," in *IEEE Transactions on Cognitive and Developmental Systems*.
- Katz, D., Venkatraman, A., Kazemi, M., Bagnell, J. A., and Stentz, A. (2014). Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Auton. Robots* 37, 369–382. doi: 10.1007/s10514-014-9407-y
- Kopicki, M., Zurek, S., Stolkin, R., Moerwald, T., and Wyatt, J. L. (2017). Learning modular and transferable forward models of the motions of push manipulated objects. *Auton. Robots* 41, 1061–1082. doi: 10.1007/s10514-016-9571-3
- Krüger, N., Geib, C., Piater, J., Petrick, R., Steedman, M., Wörgötter, F., et al. (2011). Object-action complexes: grounded abstractions of sensory motor processes. *Robot. Auton. Syst.* 59, 740–757. doi: 10.1016/j.robot.2011.05.009
- Min, H., Yi, C., Luo, R., Zhu, J., and Bi, S. (2016). "Affordance research in developmental robotics: a survey," in *IEEE Transactions on Cognitive and Developmental Systems*, 237–255.
- Moldovan, B. (2015). *Relational Affordances and Their Applications*. Ph.D. thesis, KU Leuven.
- Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., and Raedt, L. D. (2012). "Learning relational affordance models for robots in multi-object manipulation tasks," in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN), 4373–4378.
- Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: from sensory - Motor coordination to imitation. *IEEE Trans. Robot.* 24, 15–26. doi: 10.1109/TRO.2007.914848
- Myers, A., Teo, C. L., Fermüller, C., and Aloimonos, Y. (2015). "Affordance detection of tool parts from geometric features," in *IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 1374–1381.
- Nguyen, A., Kanoulas, D., Caldwell, D. G., and Tsagarakis, N. G. (2016). "Detecting object affordances with convolutional neural networks," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on IEEE* (Daejeon), 2765–2770.
- Papon, J., Abramov, A., Schoeler, M., and Wörgötter, F. (2013). "Voxel cloud connectivity segmentation - Supervoxels for point clouds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Portland, OR), 2027–2034.
- Ridge, B., Skočaj, D., and Leonardis, A. (2009). *Unsupervised Learning of Basic Object Affordances from Object Properties*. PRIP, Vienna University of Technology.
- Şahin, E., Çakmak, M., Doğar, M. R., Uğur, E., and Üçoluk, G. (2007). To afford or not to afford: a new formalization of affordances toward affordance-based robot control. *Adapt. Behav.* 15, 447–472. doi: 10.1177/1059712307084689
- Srikantha, A., and Gall, J. (2016). Weakly supervised learning of affordances. *arXiv:1605.02964*.
- Stoytchev, A. (2005). "Toward learning the binding affordances of objects: a behavior-grounded approach," in *Proceedings of AAAI Symposium on Developmental Robotics* (Palo Alto, CA), 17–22.
- Suzuki, J. (2017). A theoretical analysis of the bdeu scores in bayesian network structure learning. *Behaviormetrika* 44, 97–116. doi: 10.1007/s41237-016-0006-4
- Ugur, E., Şahin, E., and Oztop, E. (2012). "Self-discovery of motor primitives and learning grasp affordances," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (Vilamoura)*, 3260–3267.
- Ugur, E., Şahin, E., and Oztop, E. (2011). "Unsupervised learning of object affordances for planning in a mobile manipulation platform," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on IEEE* (Shanghai), 4312–4317.
- Varadarajan, K. M. and Vincze, M. (2012). "Afrob: the affordance network ontology for robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (Vilamoura)*, 1343–1350.
- Zech, P., Haller, S., Lakani, S. R., Ridge, B., Ugur, E., and Piater, J. (2017). Computational models of affordance in robotics: a taxonomy and systematic classification. *Adapt. Behav.* 25, 235–271. doi: 10.1177/1059712317726357

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Andries, Chavez-Garcia, Chatila, Giusti and Gambardella. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.