

A Survey of Knowledge Representation in Service Robotics

David Paulius, Yu Sun*

Department of Computer Sci & Eng, University of South Florida, 4202 E. Fowler Ave, Tampa, FL, 33620, United States



HIGHLIGHTS

- The main contribution of our paper is a focused discussion on what a knowledge representation is as it pertains to the robotics field: a comprehensive tool that encompasses high-level knowledge and low-level features.
- This paper aims to define and discuss knowledge representation for robots, to review the literature for existing solutions, and to identify possible issues in the creation of an effective representation. We also review several tools and/or models that have been successfully applied to smaller sub-problems in robot learning and manipulation, which can be used as components to knowledge representations.
- We discuss key characteristics of representations that allow them to function, in spite of the highly variable nature of a robot's working environment and the objects found within it. We also propose specific requirements that are necessary to build the ideal knowledge representation, drawing on concepts discussed in previous works.

ARTICLE INFO

Article history:

Available online 24 April 2019

Keywords:

Knowledge representation
Robot learning
Task planning
Domestic robots
Service robotics

ABSTRACT

Within the realm of service robotics, researchers have placed a great amount of effort into learning, understanding, and representing motions as manipulations for task execution by robots. The task of robot learning and problem-solving is very broad, as it integrates a variety of tasks such as object detection, activity recognition, task/motion planning, localization, knowledge representation and retrieval, and the intertwining of perception/vision and machine learning techniques. In this paper, we solely focus on knowledge representations and notably how knowledge is typically gathered, represented, and reproduced to solve problems as done by researchers in the past decades. In accordance with the definition of knowledge representations, we discuss the key distinction between such representations and useful learning models that have extensively been introduced and studied in recent years, such as machine learning, deep learning, probabilistic modeling, and semantic graphical structures. Along with an overview of such tools, we discuss the problems which have existed in robot learning and how they have been built and used as solutions, technologies or developments (if any) which have contributed to solving them. Finally, we discuss key principles that should be considered when designing an effective knowledge representation.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

A recent trend in robot learning research has emerged from the motivation of using robots in human-centered environments to develop domestic robots or robot assistants for use in homes and to automate certain processes and tasks which may be inconvenient for us as humans to perform. Assistants are particularly beneficial for the care of the elderly or disabled who cannot perform the actions themselves. In order to develop such robots, researchers aim to create robots which can *learn*, *understand*, and

execute tasks as human beings would. Individually, these are very complicated tasks, as they all require an introspective approach to understanding: (1) how we would perceive our environment (objects, obstacles, and navigation), (2) how we execute actions, (3) how these actions are best designed and implemented, (4) how familiar we are to objects of various types and attributes, and (5) how we ground understanding of the world. To achieve these goals, an integrated solution is needed which allows us to perform a variety of tasks such as object detection, activity recognition, task/motion planning, localization, knowledge representation and retrieval, and the intertwining of computer vision and machine learning. Furthermore, aside from handling variability, such a solution must ensure that robots avoid harming any humans around them and that they maintain a safe environment

* Corresponding author.

E-mail addresses: davidpaulius@mail.usf.edu (D. Paulius), yusun@mail.usf.edu (Y. Sun).

```

kitchen(room) :-
    contains(room, fridge),
    contains(room, stove).

living_room(room) :-
    contains(room, sofa).

bathroom(room) :-
    contains(room, toilet),
    contains(room, sink).

```

Fig. 1. Logical expressions as used in AI and certain robotics applications can be written in different ways such as the Prolog logic programming language. Here, we represent different predicates that a robot can use to determine whether the room it is in is a kitchen, living room or bathroom based on what is found within it.

through the understanding of the intentions of humans and the effects of its own actions. In this paper, we pay special attention to the way knowledge is represented for retention and re-use; a *knowledge representation* can make it easier for a robot to perform its duties and to function in its workspace safely.

1.1. Defining knowledge representation

As established before, several models or tools available at the researcher's disposal have been adopted for learning manipulations or activities to some regard for service or domestic robots. Although this may be true, only a few actually make the distinction of being a *knowledge representation*, and these full-fledged representations cannot be considered equivalent to stand-alone models, tools or classifiers. However, in our field of robotics, there is a lack of a formal definition of what constitutes a knowledge representation for a robotic system. Originally, this concept was derived from the field of artificial intelligence (AI), as representation of knowledge is very crucial to building an artificially intelligent agent or system. As originally defined in AI, a knowledge representation is “concerned with how knowledge can be represented symbolically and manipulated in an automated way by reasoning programs” [1]. The emphasis in AI is heavily placed on the semantic and symbolic representation of knowledge grounded in formal logical expressions. Concepts about the robot's world, its actions, and the consequences of said actions can be described using causal rules and predicates that are connected to one another based on common entities or variables. By identifying instances of such entities and variables, a robot can make propositions to determine whether or not these expressions are true or false through reasoning. An agent (such as a robot) can use these expressions as their source of knowledge so long as they can effectively acquire input and use reasoning to determine the correct outcome. Typically, programmers can take advantage of languages such as Lisp and Prolog to write relations (facts or rules) as clauses that can be used in queries. Fig. 1 shows an example of how logical expressions can be formulated in the Prolog logic programming language. Within such knowledge representations, rules or concepts that are built into the agent are referred to as explicit knowledge, and this knowledge is used to infer new concepts which were initially unknown to the agent as implicit knowledge.

Ideally, a robot's knowledge representation combines several forms of knowledge with reasoning capabilities such that it can devise the optimal plan of action for performing its duties. In robotics, a knowledge representation extends beyond AI's logical formalism of knowledge using only predicates to other components that are important to a robotic system: perception modules

(audio, vision, or touch), sensor inputs, motion primitives and actuation systems, motion and task planning systems, inference systems, et cetera. As an extension to the definition in AI, we define a knowledge representation for robotics as *a means of representing knowledge about a robot's actions and environment, as well as relating the semantics of these concepts to its own internal components, for problem solving through reasoning and inference*. Here, high-level knowledge denotes a semantic, structural representation of relations between different components, while low-level representations on the other hand have no structure nor symbols and instead relate to the robot's control program or physical system. For instance, we would typically describe objects and locations using prepositions or very descriptive grammar, which is unlike the low-level parameters used by robots such as sensor readings, joint angles or points in space. Simply speaking, a knowledge representation gives *meaning* to the inputs that a robot acquires and uses in its tasks. For instance, a robot can follow a certain trajectory or primitive to perform a task, but high-level knowledge can be used to give meaning to it, such as defining what the manipulation is or does. This problem relates to the *symbol grounding problem* [2], which is concerned with understanding how to intrinsically draw meaning (as high-level) to symbols (as low-level) through experience and interaction with the world; a knowledge representation for a robot or artificially intelligent agent has to answer the question of meaning behind what it may see or do in its environment. This symbol grounding problem states that we cannot merely associate symbols to concepts or meaning without considering how these connections are established by the intelligent agent. This along with the problem of symbol grounding is covered extensively in [3]. For human-like manipulations and task planning within our homes and surroundings, if a robot were to communicate with a human user and understand commands and relations as dictated by a human person, then the knowledge representation must have an ambivalent description of objects/tools and manipulations in high- and low-level representations. Furthermore, a robot should also understand similarly to how we as humans understand the world. With the inclusion of logical expressions, knowledge representations still can benefit from the powerful capability of inference and deduction as they were intended for use in AI. Several representations discussed in this paper, for example, ground knowledge in Prolog to make querying relations easier using its programming syntax.

1.2. Overview of paper

This paper aims to discuss the recent trend in knowledge representation for robots, to identify issues in creating effective representations, and to review several tools and models that have been successfully applied to smaller sub-problems in robot learning and manipulation to thus create representations. We discuss key characteristics of representations that allow them to function in spite of the highly variable nature of a robot's working environment and the objects found within it. The main contribution of our paper is a focused discussion on what a knowledge representation is as it pertains to robotics: a comprehensive tool that encompasses high-level knowledge and low-level features. We also describe how models can be used to perform specific tasks that are key to a robot's function.

Our paper is outlined as follows:

- Firstly, we delve into a discussion on exemplary high-level knowledge representations that are comprehensive and complete to be called such in Section 2. Section 2 gives an overview of those knowledge representations, with special attention paid to cloud-based knowledge representations and cognitive architectures.

- Next, we examine numerous specialized knowledge representations in Section 3. These models have been effective in representing properties and concepts that are crucial to activity understanding and task execution, despite that they represent very specific concepts.
- In Section 4, we briefly talk about classifiers designed for important tasks in robotics, despite not explicitly describing rules and semantics for understanding.
- Finally, in Section 5, after reviewing several ways of representing knowledge, we address key issues in developing knowledge representations. Specifically, we propose key concerns that should be addressed to build an effective representation while relating them to discussed works.

2. Comprehensive knowledge representations

In the ideal world, a domestic robot would be directed by commands as input and it should plan its actions and execute a series of tasks to produce a required outcome. To do this, the robot should be equipped with the knowledge it needs to perform all the steps, from receiving its instructions to identifying a motion plan to executing all required manipulations using what is available in its environment. We first begin our discussion on knowledge representations by introducing examples of those that have been introduced by researchers within recent years. These representations should contain an abstraction of low-level features (features that can be acquired by a robotic system) with high-level knowledge that can be interpreted by humans. Knowledge can be constrained to a set of terms or language known as an *ontology*. The purpose of an ontology is to define a scope of concepts and terms used to label and describe the working space of the robot in a format which is also understood by humans.

In this section, we will discuss different approaches to mapping a higher level of knowledge to a robot's understanding of the world to perform tasks. These representations combine several kinds of information. In addition, we discuss larger comprehensive projects that emphasize collaboration and knowledge sharing among a group of robotic systems. These representations are self-contained systems which combine multiple modalities of information such as vision, touch and sound, or they can draw information from several data sets all combine and contribute modality information. A good overview of data sets that offer several modalities of information to researchers related to object manipulation can be found in [4]. This would imply that robots should maintain a constant connection to access past experiences of robots as well as to upload their current and future experiences to build upon that informational source. Other systems in this section also automatically gather data from across the Web.

2.1. High-level representations of tasks

A knowledge representation innately represents skills or actions in an ontology, emphasizing reliability, safety and usability for and by robots. In order for the robot to perform effectively, a knowledge representation should contain concepts or rules based on different modalities of data, it should allow for expansion or learning new concepts, it should be used to reason logically and statistically, and that it appropriately defines objects, actions/skills, and states needed for manipulations. As we will address in more detail in Section 5, a knowledge representation should encompass all of these ideas so that a robot can reason as well as perform skills on its own. When representing skills, it is not only important to consider motion-related details such as trajectories, but we should also consider the semantic meaning behind the skills (i.e. what exactly is the motion, what changes or consequences do its actions cause on the robot's

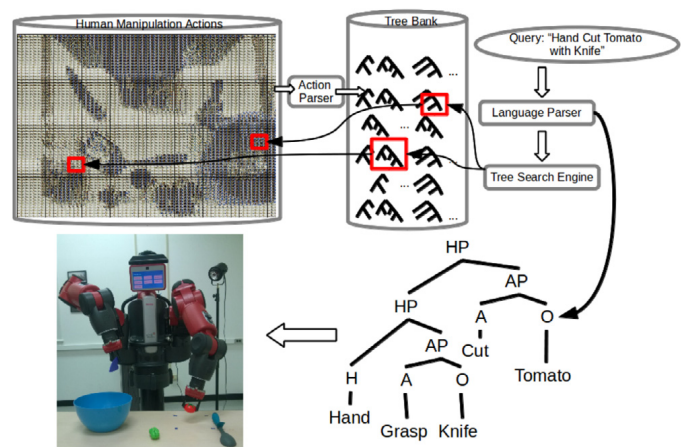


Fig. 2. Illustration of the pipeline for manipulation action tree banks, as originally presented in [10]. The action tree bank comprises of knowledge from demonstrations, and this knowledge base is searched for a tree which a robot can use for manipulation execution.

environment, etc.). Many researchers do not consider the representation of states within their work [5], which is important for recognizing when an action is complete. We pay particular attention to knowledge representations that ground tasks as high-level knowledge through the integration of different structures and models.

We discuss our first example of a high-level knowledge representation, which was originally proposed by Ramirez-Amaro et al. [6–8]. These researchers used learning by demonstration to teach robots about manipulations obtained directly from demonstrations, and they describe it as a transfer of skills from the demonstrator to the robot; upon observation of a demonstration of a skill, the robot then imitates the action performed by a human demonstrator. This sense of “transfer learning” however is different to the traditional sense within the machine learning community [9]. They can create semantic graphs as trees with knowledge in the form of transferable skills needed to execute three challenging kitchen tasks. This knowledge is directly extracted from human demonstrators and it allows the robot to perform the exact methods needed to imitate the demonstrator in manipulating objects. Human activities are learned based on several attributes: (1) the motion made by the hand, (2) the object(s) being moved and manipulated by the hand, and (3) the object(s) which these held items are being used and acted on, and they are presented as ordered pairs to train their inference engine. Once obtained, these semantic rules, grounded in Prolog, can be used in reasoning and future understanding of demonstrations through inference; these properties were applied to a decision tree classifier to automatically gain knowledge and rules from new demonstrations.

Another exemplary study in representing skills from demonstrations was presented by Yang et al. In [11], the researchers presented a method for representing observed manipulations in the form of combinatory categorial grammar (CCG) using rules originally described in [12]. This grammar vividly describes a specific action, the items being used, as well as the consequence of performing such an action [13], and each action can also be effectively broken down into smaller sub-actions or sub-activities. Using said context-free grammars, they also developed *manipulation action tree banks* [10] to represent action sequences as tree data structures that can be executed by a robot. Equipped with action tree banks, a robot can use the knowledge gathered from multiple demonstrations to determine the actions it needs

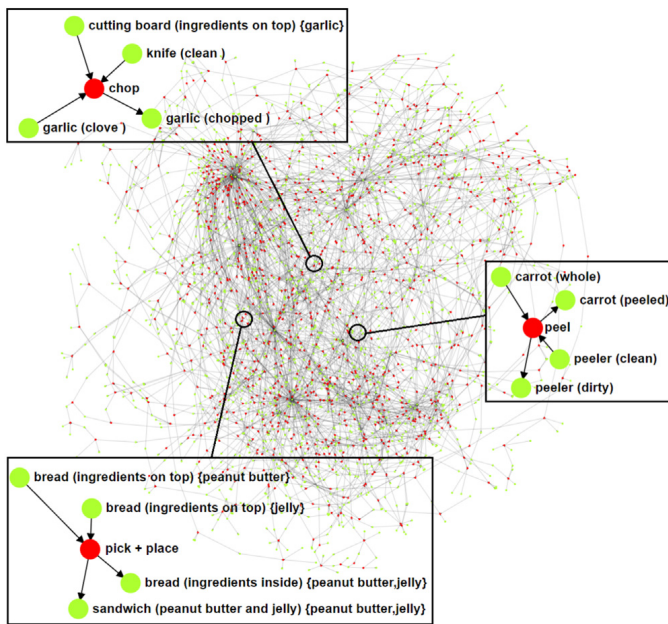


Fig. 3. Illustration of a universal FOON as a result of combining knowledge by merging subgraphs from 65 videos. Three examples of functional units are shown, each describing an atomic manipulation.

to take, in the form of a tree, for a given manipulation problem (illustrated in Fig. 2).

Similar to the prior representation, our group also introduced a semantic knowledge representation called the *functional object-oriented network* [14–16] (FOON) which serves as a basis for both learning and representing manipulation knowledge for domestic tasks. The bipartite FOON structure, akin to Petri Nets [17], contains object and motion nodes to capture the concept of object affordances. Affordances (based on J. J. Gibson’s theory of affordance [18]) are described by edges drawn between an object and a motion/action type. This network combines input object nodes, output object nodes and a motion node to create what we refer to as *functional units* (shown in Fig. 3). Individually, a functional unit describes a singular or atomic action that results in an object’s change of state before and after a motion is executed upon them (so as to emphasize state change as an indicator of the end of an action [5]); collectively, these units describe an entire activity as a subgraph. Typically, a *universal FOON* will comprise of many functional units (shown in Fig. 3) obtained from annotating several instructional videos. Through task tree retrieval (based on graph searching concepts), a robot can find a task tree which contains a sequence of steps that achieve a given goal. This is done by identifying a target node and backtracking to find steps whose requirements we satisfy (i.e. we have all the objects needed to perform actions), much like the “firing” of transitions in Petri Nets. For visualizations of annotated subgraphs, we refer readers to our website.¹ To create new FOON subgraphs, in [16], we proposed a pipeline to annotate videos using a universal FOON. This deep learning pipeline is capable of suggesting objects and motions as functional units that match activities as seen in new video demonstrations. The annotations that may be obtained from this process can be merged with our current network to add new knowledge semi-automatically.

¹ FOON Website – <http://www.foonets.com>.

2.2. Cognitive architectures

A *cognitive architecture* is a model that attempts to explain the psychological theories behind human cognition and the processes, mechanisms and modules that are pivotal to cognition and behavior. Many architectures have been proposed and extensively studied based on different cognitive theories, including the likes of ACT-R [19], Soar [20,21], CLARION [22], and EPIC [23]. Every architecture differs from one another in how knowledge is acquired, represented, retained (through long- and short-term memories), and transmitted within its internal cognitive structures since they all follow their own respective theories of human cognition; despite their differences, however, they all aim to answer the question of how human cognition works. Much like knowledge representations for robots, cognitive architectures modularize different components that are central to thoughts, perception, inference, and action and have them connected with one another based on cognitive theories. For an extensive overview of such architectures, we encourage readers who are interested in learning more about them to refer to a review done by Kotseruba and Tsotsos [24]. Within robotics, cognitive architectures are extensively studied in the field of developmental robotics as a means of understanding how we develop sensory and motor skills as an infant. Such studies look at how each internal component is connected with one another so that a robot can develop new skills and acquire knowledge about its problem domain. Readers can learn more about developmental robotics through [25–27].

Typically, cognitive architectures emphasize on how knowledge is retained and used as either long- or short-term memory, where long-term memory can refer to concepts (goals or descriptions of an entity’s environment), while short-term memory refers to instances of such concepts. Each of these concepts and skills are learned, retained, and activated once their arguments have been fulfilled by identifying them through the robot’s perception system. Architectures such as Soar [28] and ICARUS [29–31] have been used to illustrate how skills and concepts are learned from navigating throughout its environment or through human interaction. They can learn concepts such as object instance identification and location. These skills (i.e. its abilities to manipulate objects) are learned as long-term components which can be used in conjunction with other skills for performing tasks on the robot’s environment. Through a developmental approach, works such as [32–36] investigate how skills are developed through human–robot interaction (HRI). In such studies, a human assistant would interact with robots to teach them about the relationships between its actions, effects, and its own internal representation of the world. For example, in Ivaldi et al. [34], a robot was taught about object affordance through the assistance of a caregiver who supervises the robot’s actions much like a parent would. The caregiver can give the robot different commands such as looking, grasping, pushing or more complex actions like placing objects on top of another. Through HRI, the robot learns how to identify novel objects by showing the robot what the items of focus are without any prior knowledge of what they look like. Once the robot has the knowledge of those objects, the robot can proceed to learn about actions and the action’s effects while gaining positive or negative feedback to indicate whether the robot performed the task correctly or not. In summary, cognitive architectures not only aim to equip robots with the necessary knowledge to perform its duties, but they ultimately aim to explore how our cognition as humans work. Retaining a memory of experiences is crucial to learn semantic and symbolic concepts and to reason for solving problems.

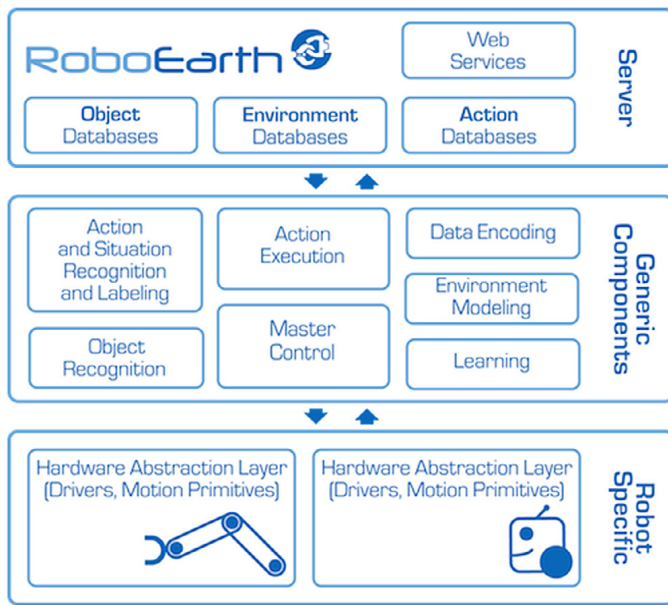


Fig. 4. Illustration of the pipeline for RoboEarth [43,44]. As taken from <http://www.i6.in.tum.de/Main/ResearchRoboEarth/>.

2.3. Distributive and collaborative representations

There is major potential to improve a service robot's performance using cloud computing [37–39]. Thanks to cloud computing and powerful hardware, it has become easier for us to perform high-intensive processing tasks with parallel programming and distributed systems. A good overview of cloud robotics and relevant technologies can be found at [39]. Robots can take advantage of off-board computations to allow for longer usage of simpler robotic systems and to reduce their computing power, thus maximizing the overall performance of robots. Furthermore, certain initiatives such as the Million Object Challenge [40,41] show that we make it easier to learn certain tasks in parallel with many robots performing tasks at once. Highlighted as an important, emerging technology [42], the Million Object Challenge² involves learning how to grasp using Baxter robots working in a distributed manner. In [41], robots can learn how to grasp using a simple monocular lens camera; using such a basic vision system can be applied to other systems to simplify grasping by using more widely accessible and less complicated technologies. These projects aim to learn how to grasp objects of different shapes and sizes by tasking a group of robots to repeatedly try to pick up a set of objects until they have successfully grasped and placed it in another container. The important thing to note here is that the *distribution of tasks* can greatly benefit robot learning.

When it comes to comprehensive knowledge representations that are based over the cloud, prominent examples include RoboEarth and RoboBrain. RoboEarth [43,44], referred to as the “World Wide Web for robots”, is an ongoing collaborative project aiming to create a cloud-based database for robots to access knowledge needed to solve a task. RoboEarth was first proposed as a proof-of-concept to show that cloud robotics would greatly simplify robot learning. RoboEarth provides an ontology for storage of semantic concepts and a method for accessing knowledge through a software as a service (SaaS) interface, where computational inferences and mappings can be done remotely.

As a collaborative system, RoboEarth allows robots to archive its own experiences (such as object types observed, motion plans successfully or unsuccessfully used, robot architectures, etc.) for recall and reuse by other capable robots. This database would contain massive amounts of data (in the form of object models, SLAM maps [47], semantic maps, etc.) which can be used for tasks such as object recognition, instance segmentation and path-planning. An illustration of the RoboEarth pipeline is shown as Fig. 5. Related to RoboEarth is another promising project headed by Rapyuta Robotics,³ a company which now deals with cloud robotics solutions. Rapyuta Robotics's system called *Rapyuta*, named after the movie from Japan's Studio Ghibli, was first introduced in [48] and then in [49] as a platform as a service (PaaS) interface for robots. It acts as an open-source middleware for accessing resources from the web such as the aforementioned RoboEarth repository and ROS (Robot Operating System) packages. Additionally, it reduces the processing done by the robot by offloading computations to the Cloud. Robots can also communicate and share information with one another through this PaaS system. This project has since evolved into the development of their cloud robotics platform for corporate solutions.

Results from RoboEarth led into the development of openEASE,⁴ also by Beetz et al. [50] (EASE being an abbreviation for *Everyday Activity Science and Engineering*). builds upon RoboEarth as a web-based interface and processing service that equips robots with knowledge from prior experiences (similar to accessing memory) and reasoning capabilities in the form of semantically labeled activity data. A robot using openEASE will have access to: (1) knowledge about a robot's hardware, its capabilities, its environment and objects it can manipulate, (2) memorized experiences which a robot can use for reasoning (why it did an action, how it did it, and what effects the action caused), (3) annotated knowledge obtained from human demonstrations. Queries and statements are formulated using ProLog, which can be sent through a web-based graphical interface or through a web API usable by robots; they allow robots to acquire semantic information and meaning to sensor input and to data structures used for control purposes. As a component to this project, Tenorth et al. [45,51,52] presented KnowRob (illustrated in Fig. 4) as a knowledge processing system for querying the openEASE knowledge base using Prolog predicates. KnowRob combines various sources of knowledge such as web pages (methods from instructional websites, images of usable objects, etc.), natural language tasks, and human observations. A robot can ground the knowledge from KnowRob to a robot's perception/action system and its internal data structures through a symbolic layer referred to as “virtual knowledge bases”. Through ProbCog [53], a statistical relational learning and reasoning system, models such as Bayesian Logic Networks [54] or Markov Logic Networks can be built for representing the state of the robot's current context. KnowRob is built within another tool known as CRAM (short for *Cognitive Robot Abstract Machine*) [55,56], a software toolbox for the design, implementation and deployment of robots using its own CRAM Plan Language (CPL). CPL is inspired by Common Lisp and Prolog for the expressive specification of concurrent, sensor-guided reactive behavior, or in simpler terms, how a robot should react to certain sensory events or changes in belief state.

Another noteworthy technology that deals with knowledge gathering and sharing is RoboBrain⁵; Saxena et al. [57] introduced RoboBrain in 2014 as a means of massively collecting concepts which are learned from automatic gathering of data

³ Rapyuta Robotics – <https://www.rapyuta-robotics.com/>.

⁴ OpenEASE – <http://www.open-ease.org/>.

⁵ RoboBrain – <http://robobrain.me>.

² Million Object Challenge – <http://h2r.cs.brown.edu/million-object-challenge/>.

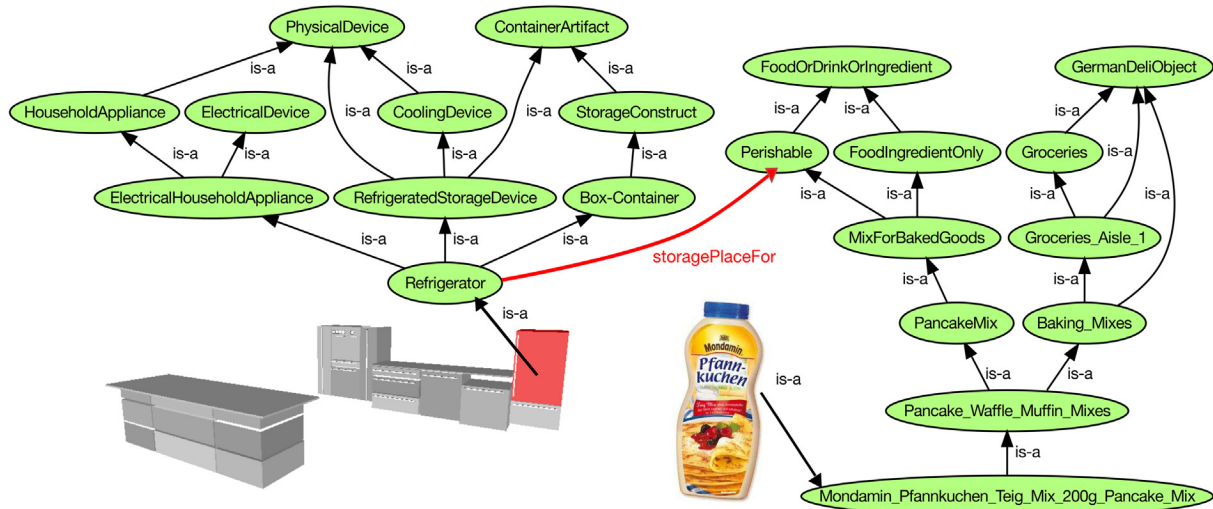


Fig. 5. Illustration of the reasoning steps used by KnowRob [45,46] for inferring the location (i.e. in fridge) of a specific object (i.e. pancake mix) needed for cooking. The semantic graph shows relations and properties about the objects and the current location.

from the Internet, simulations, and robot experiments. This differs to the RoboEarth/openEASE representation in the fact that RoboBrain uses graphs for encoding knowledge, while RoboEarth and its internal components use propositional logic and statements for defining concepts and relations in a working space. The information is represented as a graph, where nodes represent concepts (such as images, text, videos, haptics data, affordances, deeply-learned features, etc.) and edges represent the relationships between such concepts. RoboBrain connects knowledge from popular sources such as WordNet [58], Wikipedia, Freebase, and ImageNet [59]. They manage errors in knowledge collection using crowd-sourcing feedback as well as beliefs that reflect the trust given to certain knowledge sources and the correctness of concepts and their relations. To retrieve knowledge, the Robot Query Language (RQL) can be used for obtaining a subgraph describing the activity of executing a certain task. Unlike the case with openEASE, it was not demonstrated how a robot can execute the method reflected by a subgraph; however, the knowledge gathered nevertheless can be quite useful for task planning, instance identification, and inference (see Fig. 6).

2.3.1. Remarks on cloud-based systems

To conclude, we have covered several knowledge representations that take different approaches to obtaining and retaining knowledge for robotics. Although they all have their differences, the common goal is to develop a representation that is not specific to a single robot. In fact, there are several ongoing projects whose aim is to take advantage of cloud computing platforms and technologies, and the abstraction of knowledge would therefore be essential. Using distributed resources will also facilitate easier programming of robots through experience sharing and communication protocols that allow robots to work collaboratively. However, the challenge lies in developing a generalized platform that can be used by different robots; since these robots are being built by a variety of companies and groups, it is up to them to foster seamless integration with these cloud services. Close communication among researchers is crucial to ensure that research challenges are solved. Additionally, defining and adopting universal standards or protocols would be key for the success of comprehensive and distributed representations.

3. Specific knowledge representations

In contrast to full-fledged knowledge representations, there are other tools that can be useful to a robot in its execution

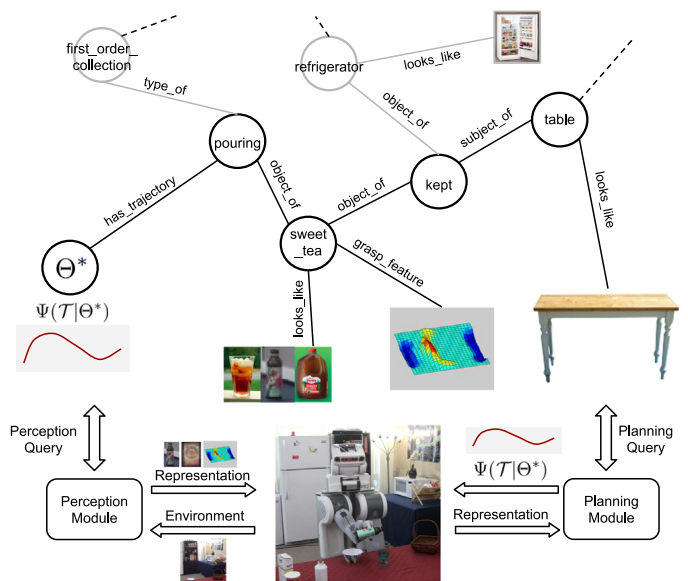


Fig. 6. An illustration of how RoboBrain can be used by a robot for solving problems as originally shown in [57].

of human-like manipulations and activities. To create the ideal knowledge representation, such a representation must be comprehensive enough to tie high-level knowledge to low-level features/attributes; these models can be put together to integrate their strengths and to create a representation suitable for service robotics. Researchers have extensively studied several types of models that can be used for representing different types of knowledge. Such types of knowledge are not limited to visual cues for grasp points or activity recognition, object–action–effect relationships to relate objects to their uses, object detectors and instance classifiers, motion identifiers, and object–object relationships. The papers selected in this section consider these kinds of relationships when building their models.

3.1. Probabilistic graphical models

In this section, we focus on learning approaches that use probabilistic models as their base of knowledge, which assume

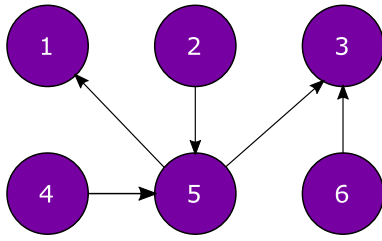


Fig. 7. A simple example of a Bayesian network.

that a robot’s world and the actions it can possibly execute are not discrete but indeterminate by nature. In other words, the robot’s world is governed by likelihoods and uncertainty, and these likelihoods are captured as probabilities grounded in such models. These models therefore can be used for representing knowledge needed by robots when it comes to recognizing activities through a basal understanding of the effects of its own actions on its environment. Although these models are examples of machine learning algorithms, they are fit to learn high-level concepts and rules for inference; other machine learning techniques that do not focus on these high-level rules would be considered as implicit representations of those rules, which will be covered in Section 4. We first begin by covering the different types of models used in recent works, and then we explore how these models can be applied to learning and representing activities and actions. We recommend that readers refer to works by Koller et al. [60,61] for further reading on the theory of these probabilistic models.

3.1.1. Bayesian networks

Bayesian Networks (BN) are directed, acyclic graphical models whose nodes can represent entities or concepts and edges denote probabilistic dependencies between such nodes and, furthermore, a flow of influence that may be incidental between them. Each node in the network represents a conditional probability distribution, where probabilities are grounded by Bayes’ Rule. If we have an unknown variable, given that we know the other variables, we can use this equation to find the unknown probability or variable. These variables are usually discrete and reflect observed phenomena within a setting or environment when applied to robotics; nodes are typically used to represent observable variables for properties or objects. In simple terms, once there is an instance of a path between one variable to another, we can then say that there is a flow of influence since one node is dependent to the other; however, if there is no path connecting two nodes, such as the nodes labeled 1 and 6 or 2 and 4 in Fig. 7, then we can say that these nodes are independent. In mathematical notation, given that these node variables are N_1 and N_2 with no parents (hence, we say \emptyset), this is written as: $(N_1 \perp N_2) | \emptyset$. More accurately, this independence is best regarded as *conditional independence*. Conditional independence assumes that two or more nodes follow a local Markov assumption, where a variable is said to be conditionally independent from nodes which are not its children given its parent node. The presence of edges between a pair of variables indicate a flow of independence, while the absence of edges between a similar pair of variables indicate that they are completely independent. It is because of this flow of influence that BNs are well-suited for inference about unobserved variables given those which we may have observed.

3.1.2. Markov networks

Bayesian Networks differ to another category of graphical methods referred to as *Markov Random Fields (MRF)* or *Markov Networks (MN)* with the stark difference in the graph’s edges: there are no directed edges present in these structures. Since

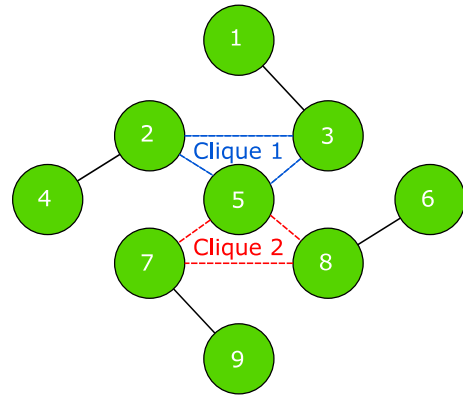


Fig. 8. A simple example of a Markov network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

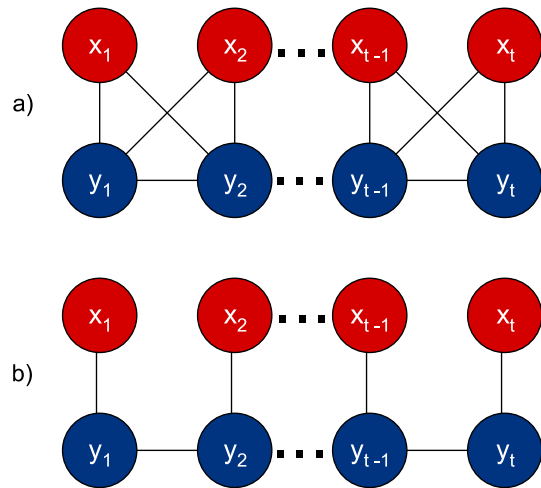


Fig. 9. An example of conditional random fields. The CRF labeled (a) and (b) differ in terms of the states’ dependencies on the observed variables. CRF (a) illustrates a model which uses several measurements for predicting a given state (which may be used for recognition), while CRF (b) shows a simpler model which would only use the observation at a time t to predict a state at time t .

there is no directional flow of influence, a MN can adequately capture cyclic dependencies. This means that there is an equal flow of influence by the pair of connected nodes. As with BNs, we can also apply the local Markov assumption to denote conditional independence. However, each variable in a MRF do not necessarily follow probability distributions. Instead, we parameterize a Markov Network using *factors*, which are functions representing the relationship for each *clique* (i.e. a subset of nodes in which all nodes are connected to one another – such as sets {2, 3, 5} and {5, 7, 8} in Fig. 8 represented by red and blue dashes in the graph respectively). These factors, when combined together, can represent the entire state of the graph and make up a distribution. These models can be used for representing the phenomenon reflected by given data just as with their directed variants, especially for representing transitions of states which are not necessarily irreversible.

A Markov Network can be specialized further for specific purposes and functions. *Conditional random fields (CRF)* [62], another instance of a Markovian model, are a special instance of MNs that encode conditional distributions (shown in Fig. 9). Given a set of observed variables X (representing data or measurements) and target variables Y (representing states), a CRF models a distribution based on X , i.e. $P(Y|X)$. The definition of the distribution for

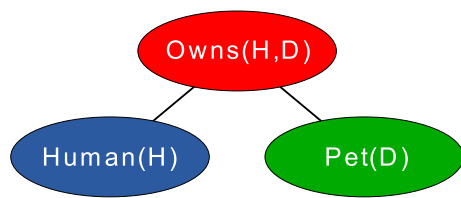


Fig. 10. An example of a Markov Logic Network. This MLN represents the first-order logic statement that a human H can own a pet D (or more formally, $\forall H, D : Owns(H, D) \rightarrow Human(H) \leftrightarrow Pet(D)$). For example, if we have instances of a human “Ryan” who owns a dog “Wilfred”, then this would follow this logical statement where $Owns(Ryan, Wilfred)$, $Human(Ryan)$ and $Pet(Wilfred)$.

a CRF would not solely consider factors (or cliques) with those observable variables (i.e. a given C_i must not comprise of only variables in X). CRFs are typically used for modeling predictions of sequential data, where the observable variables represent concepts that are used for predicting a certain outcome as target variables. A *Markov Logic Network* (MLN) [63] uses the structure of Markov models combined with first-order logical statements or predicates, which describe the state of the world and those of objects and entities that are present within it. These logical statements consist of four parts: constants, variables, functions, and predicates. Each node is labeled with first-order logical statements to describe a probabilistic scenario, and they are assigned weights which reflect the likelihood of them being active. A very simple example of a MLN based on a single first-order logic statement is shown in Fig. 10. This differs from the regular usage of first-order statements: if a statement or condition is not true, then in the MLN, it means that it has a low probability of being true in that situation instead of being assigned zero probability.

3.1.3. Applications of probabilistic models

In this section, we discuss how these probabilistic models have been used for learning about activities and manipulations. Bayesian Networks in research studies are mainly used for capturing the effects of actions upon objects in a robot’s environment. When capturing such effects, a robot would be presented with demonstrations of observable action and effect pairs in order to learn the relationships between them. These relationships can be taught through learning by demonstration, and it can be classified into two subcategories: *trajectory-level* learning and *symbolic-level* learning [64]. Trajectory-level learning is a low-level representation approach which aims to generalize motions on the level of trajectories and to encode motions in joint, torque or task space, while symbolic-level learning looks at a high-level representation which focuses on the meaning behind skills in activity learning. The robot’s interaction with its environment serves to either learn new motor primitives or skills (trajectory-level) or to learn new properties associated with the type of grasp they make or the skills they use, the object’s physical features, and the effects that occur from executing an action (symbolic-level). In works such as [65–70], a robot can use basic, pre-programmed motor skills (viz. grasping, tapping or touching) to learn about relationships between an object’s features (such as shapes, sizes or textures) and features of its actions (such as velocities and point-of-contact). The controllers of these skills are tuned by the robot’s experiences and exploration with its environment, and the causality of these actions and their effects upon the world, based on object features, can be represented through a BN. The robot can use the knowledge it has gathered from interacting with objects and performing fine-tuning to select the appropriate action that achieves a human demonstrator’s result. Similarly, Jain et al. [54] and Stoytchev et al. [71,72] used these networks to learn about the effects of actions on objects based

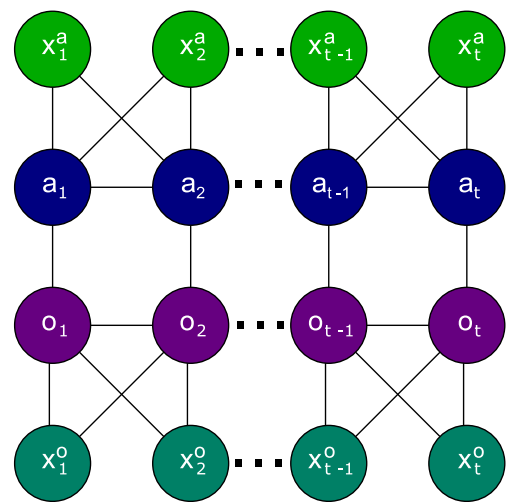


Fig. 11. Illustration of a factorial CRF as used in Kjellström et al. [75] to predict actions a and objects o based on observed features x^a and x^o (for both action and object) at some given time t .

on demonstrations with tools. Their BNs were built based on geometric features relevant to a tool’s function (and tools similar to it), which they coined as functional features, for predicting the effects of tools unknown to the robot with the learned model. For instance, objects used for cutting have a sharp edge as a functional feature, and those used as containers have a non-convex shape for holding matter or substances; once the robot can identify these features, it can use the trained model to predict the results of specific actions. The tools’ effects are given by the target object’s displacement, the initial position of the tool relative to the target object, and the target velocity after impact was made on the tool’s functional feature. A BN can also be used with other modalities of data such as speech input for grounding actions to their effects [73]. Instead of learning object–action effects, BNs can also describe the likelihoods of object–object interaction between specific pairs of objects as learned from observing human behavior [74]. These affordance models are particularly useful in improving both activity recognition and object classification and teaching robots to perform tasks using paired objects.

With regard to Markov Networks, instead of a cause–effect relationship as inherently represented in Bayesian Networks, researchers can focus on learning dependencies between concepts useful for learning from demonstrations and identifying future cases of actions or activities, which can particularly be useful for a robot in learning new concepts or safely coordinating with others working in the scene. As an example of activity recognition with MNs, Kjellström et al. [75,76] used CRFs to perform both object classification and action recognition for hand manipulations. Their reasoning behind the simultaneous classification of both objects and manipulations comes from: (1) the sequence of an object’s viewpoints and occlusion from the hand indicate the type of action taking place, and (2) the object’s features suggest the way the hand will be shaped to grasp it. They use factorial conditional random fields (FCRF) [77] to map this two-way relationship between object features and possible action types (shown in Fig. 11). FCRFs have the advantage of mapping the relationship between the data level (features found in observations) and the label level (object types and properties and their relatedness with actions), thus effectively capturing affordances suggested by the hands and objects. A similar approach is taken in [78] to identify activities based on objects and their proximity to hands in a scene using CRFs. Prior to this work, in [79], this

association was described using text descriptions, which they denote as functional object string descriptors, which significantly perform better than using purely appearance-based descriptors for recognizing similar events or activities. Using CRFs to represent the spatio-temporal relationship between objects, denoted by functional classes, improved over their previous approach of performing activity recognition with string kernels.

Another example of spatio-temporal representation of activities to objects was done by Koppula et al. [80], who used MRFs to describe the relationships present in the scene between activities and objects. By segmenting the video to the point of obtaining sub-activity events, they can extract a MRF with nodes representing objects and the sub-activities they are observed in and edges representing: (1) affordance-sub-activity relations (i.e. where the object's affordance depends on the sub-activity it is involved in), (2) affordance-affordance relation (i.e. where one can infer the affordance(s) of a single object based on the affordances of objects around them), (3) sub-activity change over time (i.e. the flow of sub-activities which make up a single activity), and (4) affordance change over time (i.e. object affordances can change in time depending on the sub-activities they are involved in). They proposed using this model for the purpose of recognizing full-body activities occurring in videos collected for their Cornell Activity Dataset⁶ (CDA). Following [80], they investigated how they can anticipate or predict human behavior using CRFs to ensure that a robot reacts safely in [81]. A special CRF, called the anticipatory temporal CRF (ATCRF), can be built after identifying object affordance for a particular action type and can effectively describe all possible trajectories of human motion and sub-activities that are likely to be taken as time goes by.

Using first-order logic statements are effective for reasoning and inference; taking advantage of such logical expressions, a MLN effectively represents a systematic encoding of the robot's world. A MLN can be thought of as a knowledge base, as these logical statements can be used for reasoning and drawing conclusions based on what a robot sees and observes. For instance, with regards to activity recognition using affordance, Zhu et al. [82] used a knowledge base, in the form of a Markov Logic Network, for inferring object affordances in activities which are suggested by the pose of the human demonstrator in videos. They can do the inverse by predicting the objects and actions occurring in a given scene based on the pose of the human demonstrators with relatively great performance. This could only be done after they collected a large amount of information about these usable objects and affordances as features, but there is no need for training multiple classifiers for each object-based task to identify each type of detectable activities as typically done with other machine learning approaches. A MLN such as theirs can be used alongside other components for activity recognition to predict human intention and to enforce safe behavior within a human-robot collaborative environment. KnowLang [83,84], proposed by Vassev et al. is a knowledge representation that was developed for cognitive robots where the power of AI's logical expressions of the world with what they actually perceive in their world. It also combines first-order logic with probabilistic methods which they can use for defining explicit knowledge for the robot. However, when making certain decisions in which lies uncertainty, statistical reasoning through the use of Bayesian Networks makes the process more reliable through reasoning on beliefs. Experiences can be reflected through probabilities, and such distributions are likely to change based on what the robot sees or acts.

3.2. Semantic graphs

Graphs are very popular for representing information quite simply because we can display knowledge in a graphical form

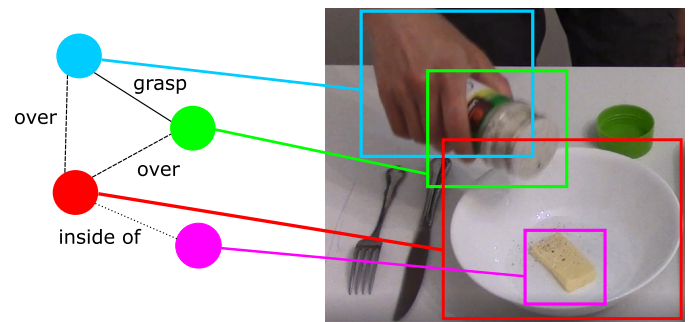


Fig. 12. Illustration of semantic understanding of a certain scenario. In this case, we are creating a semantic graph whose nodes and edges describe the objects in the scene and their relative positions.

that can be interpreted and verified visually by humans. As previously discussed, probabilistic models can also be represented graphically and make excellent inference tools. With probabilistic graphical models, edges would describe the likelihoods of certain variables as nodes causing others to occur. However, in this section, we will be referring to another subset of graphs referred to as *semantic graphs*, whose nodes and edges describe semantic concepts and details between entities as observed in demonstrations. Spatial concepts, for instance, can be described by semantic graphs, where nodes can describe objects within a scene, and edges describe commonality or contextual relationships between objects in terms of position (one object may hold another object, one object may be on top of another object, et cetera); such an example is shown as Fig. 12. Some graphs also embody temporal relations, where two or more particular events are related by time (e.g. one event must occur before another). Basically, these networks can be used for compressing details and capturing relationships as needed by a robot.

3.2.1. Activity recognition & inference with graphs

One of the major problems in robot learning has been in learning to recognize activities to facilitate the transfer of knowledge to robotic systems. A major component of activity recognition and understanding is predicting an ongoing activity or action as seen in a video demonstration. Knowledge extraction is mainly done through the processing of activity-based videos and images or through interpreting sensor data from demonstrations either done by the robot or human demonstrators. Techniques such as optical flow can also be used for identifying motion patterns to then recognize motion types or primitives. These elements can be used as context clues for inference. Previous work focused on understanding the activity taking place with the use of the primary portion of such videos to recognize the likely activity and results which would be implied by it [85–87], especially for predicting the next action which would take place in a sequence of actions [88]. Semantic graphs have been used for representing affordances based on how objects are used with one another based on visual cues and spatio-temporal relatedness.

Segmentation techniques can be applied to images to identify the objects being used in demonstrations. These segmented “patches” can be used for labeling nodes in semantic graphs (such as in Fig. 12). For example, Aksoy et al. [89,90] created these semantic graphs after segmentation. Their focus was in understanding the relationship between objects and hands in the environment and generalizing graphs for representing activities and identifying future instances of these events. This approach can be classified as unsupervised learning since there is no explicit indication of what the objects are; objects instead are solely encoded based on manipulations in matrices, which they refer

⁶ Cornell Activity Dataset – <http://pr.cs.cornell.edu/humanactivities/>.

to as *semantic event chains* (SEC). These structures capture the transitions in segment relations (temporal information), which are then generalized by removing any redundancies in activities, to be used in recognizing similar events. They characterized spatial relations of objects as non-touching, overlapping, touching, or absent within each matrix entry and as edges which connect image segments. Sridhar et al. [91] also used segmentation to separate the scene into “blobs” (similar to the patches in [89,90]) and cluster them as a semantic graph, based on the objects’ usage in videos, for affordance detection. Their semantic graphs are called *activity graphs*, structures which describe the spatial (whether objects are disconnected, found in the surrounding area, or touching) and temporal (relativity with respect to episodic events) relationships in a single video. With such graphs, similarity between activities can be measured even with varying object instances, orientations, hand positions, and trajectories. Zhu et al. [92] focused on segmenting the tool and the object it is being used on to create a spatial-temporal *parse graph*. Within these graphs, they capture the pose taken by a demonstrator, the observed grasping point of the tool, the functional area of the tool that affords an action, the trajectory of motion, and the physical properties (such as force, pressure or speed) that govern the action. These graphs can then be used to infer how objects can be used based on prior demonstrations.

3.2.2. Representing sequences of skills or events for tasks

Semantic graphs may also be used for task execution in the form of skills, containing knowledge that can be used by robots for manipulations. In these structures, nodes represent objects and action types. Several researchers have taken approaches to learning object affordance and representing them in this manner. As discussed before, notable examples of such graphs include [7,8,14,15], the latter being inspired by Petri Networks (or simply Petri Nets) [17]. Petri Nets were originally intended for illustrating chemical reactions, and they have been shown to be applicable to other domains such as robotics and assembly. Petri Nets are networks with two types of nodes: *place* nodes and *transition* nodes. Place nodes represent states of objects or entities, and transition nodes represent events or actions which cause a change in state. The term for state change with respect to Petri Nets is *firing* of transitions. Typically, all place nodes must be present for transitions to fire, therefore enforcing an implicit ordering of actions and behaviors. Costelha et al. [93,94] used Petri Nets for representing robot tasks over other methods such as Finite State Automata (FSA) which require more memory and a larger space of representation and its limitation to single-robot systems. Petri Nets, on the other hand, can represent concurrent system actions and sharing of resources. The implicit ordering of events allows them to filter out specific plans which can never happen or those which should be avoided. They created *Petri Net Plans* (PNP), which are essentially a combination of ordinary actions and sensing actions using control operators.

Similar to context-free grammars are *object-action complexes* (OAC, pronounced like “oak”) [95–98]. This representation’s purpose is to capture changes in state of the environment in a formal structure which can be used for task execution. OACs combine high-level planning and representation of the world with low-level robot control mechanisms called instantiated state transition fragment (ISTF). An ISTF can be seen as minute, lower-level constructs, which can be put together like context-free grammars, for a concrete understanding of an action’s effects before and after a motor program (i.e. action) is executed; OACs can be created after learning a variety of ISTFs. ISTFs are generalized to only contain the object-state changes which are relevant to an action tuple (identified through methods described in [99]), as ISTFs can contain object-states which may or may not be affected

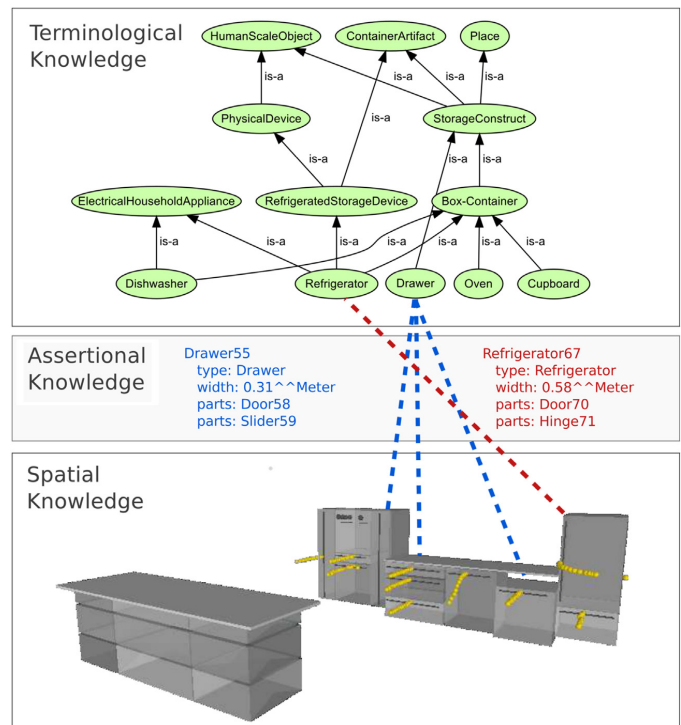


Fig. 13. Illustration of various properties captured in the semantic object map (SOM) representation as presented in [101,102]. SOM ties geometric data to semantic data that can be used in reasoning.

or caused by a specific action. Given a set of object affordances and relations learned, an associative network can be used for encoding and retrieving the state change that will occur from a certain OAC permutation.

Other approaches so far have attempted to map high-level manipulation skills to graphical structures. Instead of focusing on manipulations, Konidaris et al. [100] chose a different representation for trajectories as skills. These researchers introduced an algorithm for learning skills from demonstrations, focusing primarily on motion trajectories from tasks, called CST (for *Constructing Skill Trees*). Motion trajectories can be broken down using change-point detection, and these smaller trajectory components are referred to as skills. The aim of change-point detection is to find the point(s) at which there is a significant or observable change in trajectory. After successfully segmenting the trajectories into skills, these learned skills can be combined together as *skill trees* for potentially novel manipulations by appending skills into one executable skill.

3.2.3. Combining semantic and physical maps

Semantic graphs can also take the form of *semantic maps*, which are special graphs that relate spatial or geometrical details (such as morphology of space, position and orientation of objects, geometry of objects as models, and any positions of special places of interest) to semantic descriptions (such as the purpose of objects). These spatial features can be acquired from SLAM modules, including properties such as sensor readings or features, orientation, and absolute or relative positioning of objects or landmarks; through SLAM, the robot can obtain a map of the environment that uses the contextual information to particularly highlight instances of different objects or points of interest that lie there and to identify where they are, for instance. Semantic maps have also been used in identifying grasps by using geometrical features about the objects [103,104]. An example of

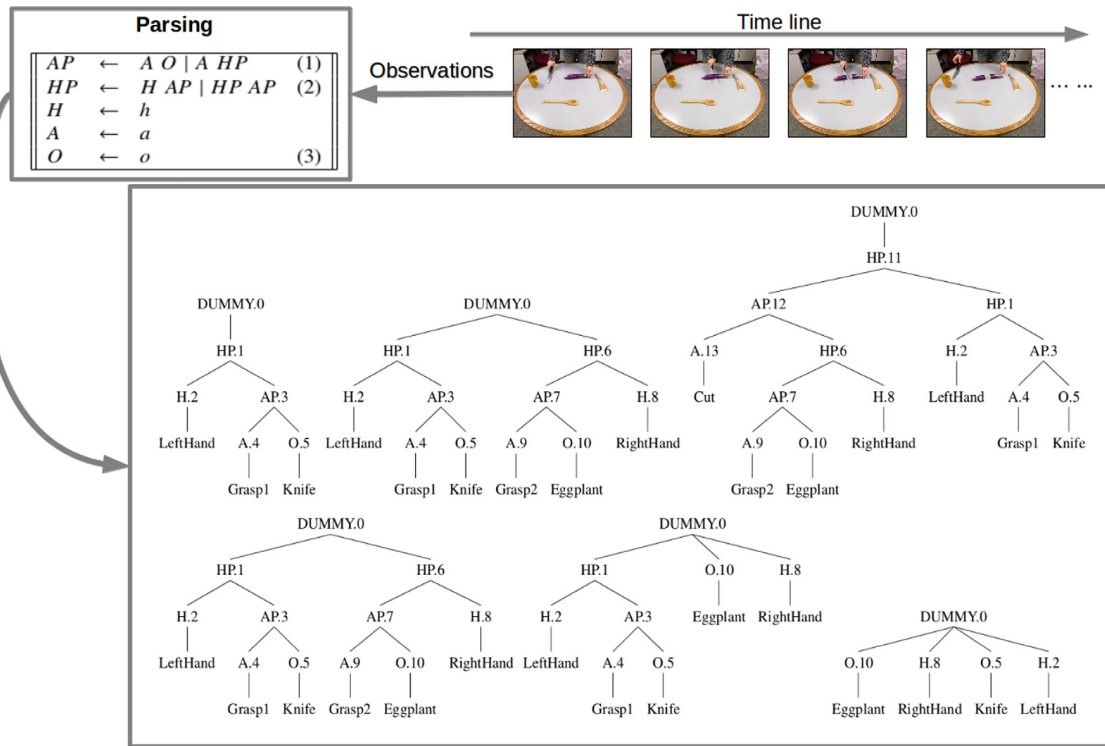


Fig. 14. Illustration of how a context-free grammar can be used to construct parse trees as shown in [11]. The rules for their context-free grammar is described by production rules 1 to 3. Their manipulation action grammar includes the non-terminal symbols *A* describing the action, *O* describing the objects or tools being used within the activity, *H* describing the hand that is manipulating the object(s), *AP* and *HP* to describe the context of the action, hand and objects together. Actions based in this grammar can then be used in robot task planning.

how semantic maps can be created was proposed by Galindo et al. [105], which integrates causal knowledge (how actions affect the state of the robot’s environment) and world knowledge (what the robot knows about objects around, their properties, and their relations), using two levels of information: the spatial box (S-Box) and terminological box (T-Box); they mark the physical location of objects at the sensor level as well as note the free space in rooms with S-Box, while the innate properties of these objects are linked using ontologies with T-Box. *Semantic object maps* (SOM) [101,102] also serve a similar purpose to combine geometric data with semantic data to answer queries to determine whether a certain action can be executed given present circumstances in its environment. For example, as in Fig. 13, a Room instance can be inferred to be a kitchen if there are items within the environment that are typical of a kitchen, such as a stove or a fridge. With regards to creating semantic maps through human interaction, works such as [106–108] aimed to develop HRI systems to impart knowledge to a robot about its surroundings: what lies around it and the conceptual knowledge tied to these elements. Both systems use audio for interacting with robots; in addition to this speech recognition system, [108] combined a tangible user interface and a vision system with a robot’s modules for motion planning and mapping to compile and create a knowledge base which a robot can then use for its navigation through its environment.

3.3. Context-free grammars

Context-free grammars are also an effective way of constructing or representing semantic graphs or structures, as they guarantee completeness, efficiency and correctness. A context-free grammar defines rules to creating semantic structures and sub-structures as strings using its own symbols (called *terminals*) defined within a finite set called an *alphabet*. These terminal

symbols can be used when substituting variable symbols called *non-terminals*; the substitution process is described by production rules, which allow us to generate sentences. With such a formal definition of a context-free grammar, researchers have been able to define rules that describe concepts such as manipulation/action types which can then be useful for defining plans that robots can use for execution and also for the composition of skills into sub-skills. One such example of a context-free grammar was proposed by Yang et al. [10,12,13,109], who studied how manipulations in activities can be represented through grammar and then broken down into visual semantic graphs. These parse trees (as described in Fig. 14) are built from demonstrations can then be used to form manipulation action tree banks. The high-level representation serves as a symbolic representation of the manipulations which describe each step required to solve a given problem. In a different approach that also uses context-free grammars (CFGs), Dantam et al. [110,111] also formulated robot primitives and control policies using their own representation called *Motion Grammars* (MG). A parse tree can be constructed to reflect the procedures being executed and they can be broken down by a motion parser to create sub-tasks until they have been satisfied similar to the representation introduced by Yang et al. The experiments conducted in their work showed that a robot can effectively execute game-related manipulations while maintaining a safe environment for humans playing with it. Even though there was no discussion on how these can be used for service robotics manipulations such as domestic or industrial tasks, the use of MG can ensure safe, real-time control of robot tasks.

4. Model-level representations

The aim of machine learning algorithms is to find a suitable model or function for discriminating classes or for predicting

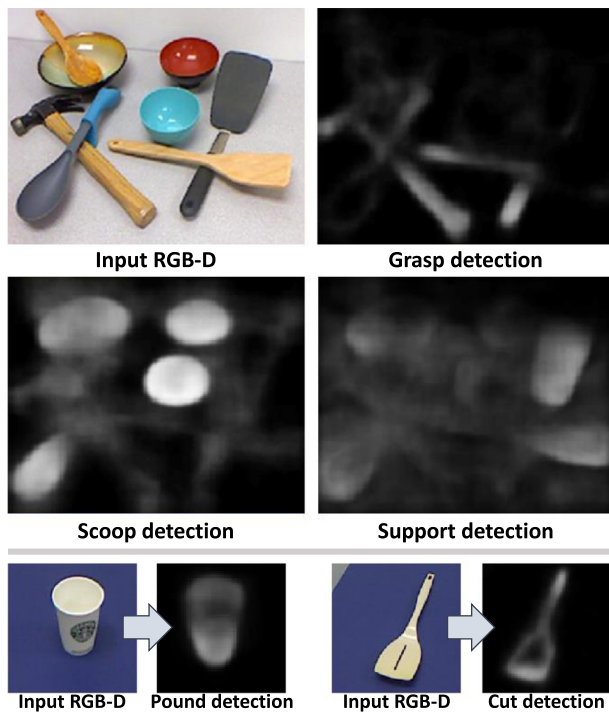


Fig. 15. An example of affordance-based pixel hot spot detection as done by Myers et al. [112]. In their project, they can look at a scene and identify the types of operations a robot can perform using objects found there by highlighting tools that afford a specific action or purpose. A robot can use this to further decide its actions.

effects or labels; techniques such as neural networks and probabilistic graphical methods can all be considered as machine learning algorithms in addition to those discussed within this section. Although we have previously discussed some of these models, they were used in the context of learning high-level rules or inter-relationships between concepts (such as the object–action–effect relationship). They are equally suitable along with other machine learning approaches to learn lower-level concepts that are not sufficient to explicitly represent knowledge and to perform reasoning tasks. However, when applied, classifiers can implicitly represent rules that describe classes needed for a robot’s tasks. Therefore, such learning models complement knowledge representations in performing specialized tasks, such as image understanding, segmentation and processing, to then further guide a robot’s future actions.

4.1. Classical machine learning techniques

Major successes in reinforcement and imitation learning can be attributed to machine learning methods [113–116], proving them to be useful for training robots in a bottom-up fashion. Just as we have addressed before with other models such as probabilistic models, imitation learning aims to teach a robot about skills that can be incrementally learned and used for task planning. Machine learning algorithms have been used in this regard to learn new motor primitives or skills [99,117–119] or to successfully classify or predict the outcomes of actions on objects [120–122]. Works like [99] and [119] apply machine learning algorithms to learn how to grasp objects based on their features. In [120], the researchers aimed to represent the effects of the robot’s motions on objects in its environment using a support vector machine (or SVM, a model which can be used to discriminate between classes by representing data in high

dimensions and finding a function or hyperplane that optimally separates class instances), which, once trained, can be useful in predicting the change in an object’s state that is reflected by its displacement in the scene when performing actions (e.g. whether the objects would roll or respond to pulling). SVMs are also used in [121,122] to learn about object effects (such as push-ability and stack-ability) based on physical properties of objects. To better understand the consequences of its actions, they tweaked various parameters such as the hand speed and the tactile sensors in the robot’s hand. For the task of affordance detection, [112] used two classifier variants, hierarchical mapping pursuit (or HMP, which are similar to neural networks) and random forests (an ensemble of decision trees or similar classifiers), to identify pixel hot spots within a scene that suggest features related to affordances; an illustration of this task is shown as Fig. 15.

4.2. Neural networks for deep learning

A vast number of studies have taken advantage of the effective method of *deep learning* and the use of neural networks for tasks such as handwritten character recognition [123,124] and image processing and recognition [125–127]. The power in deep learning techniques lies in its property as a “universal approximator” [128], where it can theoretically learn any model or function given the right number of layers (meaning one or more hidden layers). A deep neural network represents a feature detector or classifier which can be stored and reused by processing systems for specific purposes such as identifying affordances, grasping points, object recognition, instance segmentation, and many more. The amount of training time can be significantly reduced by using pre-trained models and fine-tuning them with other data sets, since the networks will retain valuable information that can be extended to other domains. For instance, it is customary to use pre-trained versions of well-established networks such as VGG [126], ResNet [129] and GoogLeNet [130], which have all been successfully applied to the task of labeling images. We refrain from going too deep into this machine learning technique in this paper, and so we refer readers to [131,132] for extensive reviews on deep learning and neural networks.

Neural networks have been proven to be capable of learning very effectively in both supervised and unsupervised learning for robotics. As a component to a knowledge representation, these models can be used by a robot to handle decisions or disputes among a variety of choices that may not be easily grounded in explicit rules. One task for which they have particularly been effective for is grasp synthesis; in [133], the authors review the research in data-driven grasp synthesis, where the ideal or optimal grasp type is selected among a sampling set of grasps. A neural network (such as those in [134–136]) would be used to recommend the ideal point to grasp objects, even those that have never been seen before [41,137]. In addition to robotic grasping, these networks are also effective for scene understanding [138–140], specifically affordance detection within a scene [141–143]. They have also been applied to the generation of textual descriptions of what they observe in their environment [144] for communication between robots and humans. The identification of what is around the robot can also be referenced to underlying rules, facts, or beliefs. Deep learning works well in activity recognition due to their exceptional performance in image segmentation, object recognition and instance identification [126, 145,146], even in real-time [147]. Such networks can be used for learning task plans from demonstrations to build upon a robot’s knowledge base [148,149]; these learned plans can be used for predicting future activities to anticipate the consequences and forces exerted from those actions [150]. Finally, neural networks can be used in learning control and the dynamics/physics of actions [151–155].

4.3. Summary

To conclude, many of these works have shown how machine learning algorithms are effective in the classification of a wide array of applications such as detecting object affordances and grasp points. Traditionally, algorithms such as SVMs have been used for a wide array of tasks and are now being replaced with deep learned networks since neural networks are easy to train. Neural networks are especially very useful for extending applications to simpler systems, and they have proved to work exceptionally well in image processing tasks. We can use trained networks to help us identify key features that can be adopted to a programmed approach or to other rule-based classifiers. Despite their great success and trending usage among the robotics community, we should not consider them as the “magic bullet” for all problems [156]. They simply cannot serve as stand-alone knowledge representations due to the lack of explicit knowledge and meaning behind what they implicitly represent; nevertheless, they are quite useful for guiding the decision-making process for a robot.

5. Selecting knowledge representations

Previously, we examined several different approaches that have been implemented successfully to solve a wide array of sub-problems needed to build a complete and comprehensive knowledge representation. Each technique comes with its own limitations and strengths, and they should be used accordingly with one another to create an ideal knowledge representation. In this section, we would like to pay closer attention to the key features of knowledge representations as observed in the works discussed in this paper. We argue that the following components (depicted as Fig. 16) are needed for a sufficient knowledge representation: (1) the representation of motions or skills for task planning at the semantic level of understanding; (2) a perception system, which integrates multiple channels of input such as vision, haptics, natural language, audio or sensors to localize objects, tools, or entities (humans, animals, etc.), to perceive observable states as well as possible obstacles, and to communicate with other robots or humans; (3) the grounding of perception or control to logical statements, symbols, or values, which can be used to connect concepts and solve problems through logical reasoning; (4) the retention of experiences as beliefs to represent uncertainty, which can influence a robot to consider other possibilities in task planning as well as tailoring the robot’s actions to the human beings or environment it has been built to operate under (through statistical reasoning); (5) the committing of newly acquired knowledge and concepts for re-use as well as the ability to learn new concepts; (6) a suitable definition and representation of the (household) environment. A majority of these things are unfortunately not a focus in several of the representations discussed in this paper, as these components are usually built independently from others. Perhaps the closest representation to this idea is KnowRob through openEASE [157]; nevertheless, these systems still require a lot of work to get them to the standard of working reliably among humans. In the following subsections, we discuss these different ideas and why they are important for knowledge representations used in service robotics.

5.1. Component #1: Representing motions for action

Perhaps the most important of all components, a knowledge representation gives meaning to a robot’s actions and environment; the way we represent these aspects are very crucial to building an effective knowledge representation. In particular, we



Fig. 16. Overview of Section 5. When designing or selecting a knowledge representation, researchers should take into consideration as many of these concepts as possible; this however is a very challenging problem, as there is no fundamental way to integrate these components together into one representation.

should focus on meaning behind a robot’s actions and motions taken for the understanding the consequences of its actions. A representation of actions, whether grounding them as motion primitives or denoting an action as a combination of skills, can be used for signifying meaning behind the transition between object states (or quite simply, the consequences of performing those actions). When reasoning and determining its goal, the robot needs to find the right actions to get to that goal. Saving these primitives as skills also facilitates re-use, and they can always be tuned to different parameters based on what the robot perceives. The organization of activities or tasks as motion/skill trees makes it easy to combine skills or primitives according to the current problem and the present state of the robot’s world. Whether it be representative of trajectories as in [100] or individual skills [6–8,10,95–98], the idea that ties them together is the ability to combine multiple units together to suit the problem at hand. Using combinatorial representations such as [10,12,13,100,109] and [110,111]) also emphasize the idea of compositionality and flexibility of structures for task planning.

5.2. Component #2: Perception

An important component to any intelligent agent is perception, as an agent such as a robot needs to acquire input to determine how to proceed with future actions. For instance, a robot would need to acquire input to determine how it would trigger conditions for rules such as in Fig. 1. Many solutions discussed in this paper under Section 4 have been shown to be effective in processing input to determine grasp points, object affordance hot-spots, object recognition and instance segmentation, etc. Other classical computer vision techniques have also been used for processing input through images or videos to perform similar tasks. A knowledge representation can also store useful object descriptors in forms such as 3D models or point clouds,

which are made available (implicitly or explicitly) in works such as [43,44,57,149] and [134–136]. Now that we have shown that we can detect and process all manner of input, we should instead focus on how this can be done in *real-time*. In addition to this, the models used for perception should be trained to handle uncertainty as best as they can by training on a wider set of samples for the corresponding household tasks the robot is given; this brings us to another point in selecting suitable training samples to give a general representation of several concepts to work with. For instance, one possible limitation to machine learning models is the likelihood of overfitting to the training set, causing the model to develop a bias towards the data rather than learning what is really important for classification.

5.3. Component #3: Grounding concepts with logic

As a basic premise of AI, knowledge representations allow for reasoning using logical statements and expressions by using explicit knowledge to make inferences to acquire implicit knowledge describing the current state of the world that then reflects the next likely action to be taken. Structures such as MLN (as used in [82–84]) integrate logical rules with probabilities. However, many of the learning models that have been developed lack the grounding of such concepts to explicit statements. Without doing so, it becomes a challenge to define expectations of actions, such as the change in the state of the robot's environment; when ignoring states, it becomes difficult for a robot to determine whether it has completed the action successfully or not [5], and we cannot always assume that the robot will solve problems without fail. Rules that imply an order to actions and tasks make it easy to modularize actions. For instance, in [14–16], states are explicitly reflected in object nodes within the network, and these can be used for indicating whether a robot has successfully performed the manipulation or not while also representing precedence of its actions. Logical formulations of concepts also help us to translate human-understood concepts into a “mid-level” knowledge that can be understood by both humans and robots alike.

5.4. Component #4: Probabilities & beliefs

Realistically, an agent cannot fully depend on logic to understand its world, as predictive modeling does not always give a straightforward answer. It is through the use of beliefs that a robot can statistically reason to make decisions about its future actions. Several classifiers, for instance, cannot give a 100% accurate answer about what it is identifying, and so weights are assigned to give the best guess of what is presented to the robot. Another example would be with a robot's sensors: innately, sensors capture noise from the input it gathers from its surroundings. A robot should leverage its beliefs or estimates with measurements because it will not always have a completely correct understanding of its world. The use of probabilities also makes for strong inferences through probabilistic models as discussed in prior sections. Beliefs can also be updated to make stronger inferences about what a robot sees around itself to make smarter decisions. For instance, by using information about object affordances, a robot can select the right tool for a particular job; identification of said tools can be done using learned classifiers such as [112,141–143]. With respect to previously discussed works, probabilistic and statistical reasoning methods have been used in learning how its actions affect its environment. Leveraging explicit knowledge using a robot's beliefs about its surroundings, through both logical and statistical reasoning, would allow it to make the best decision possible. For example, in the task of SLAM, the joint task of location estimation and update of its belief

and location is important for the robot to proceed with navigating its environment with as minimal error as possible. Weights may also aid the robot's system in deciding between multiple courses of action so as to reduce the likelihood of task failure.

5.5. Component #5: Learning from experiences

To guide a robot in performing its duties, it is beneficial for it to retain knowledge from prior experiences in solving a problem (or failing to do so). This is a key advantage that is contained in knowledge representations such as openEASE, RoboEarth and KnowRob, where a robot can record details about the manipulations it has performed over a distributed system. Other robots that are attempting to perform the same manipulation can then access those prior experiences to adapt its actions to its problem; details such as trajectories can help to guide the current robot to follow similar patterns. More specifically, in the KnowRob representation from Section 2.3, one of the key aspects of the platform is to record the experiences of performing a specific task or manipulation. As in their recent work in [52], they illustrated how one robot can use the experience of opening a fridge from another robot to perform the same task; however, the exact trajectory as learned from the previous robot cannot be completely used due to differences in the state of the targeted object and its environment. Therefore, it is not simply imitating what a previous robot has done, but rather, it refines the motion to better suit the current problem. As another example of alternatives for future task planning, aside from others in Section 2.1, the FOON representation draws from multiple demonstrations to potentially create novel manipulation sequences. This is done through the merging of subgraphs for each demonstration into one universal FOON. With machine learning models, the overall idea of training these models is to adapt them to accurately predict future instances. Probabilistic models, for instance, have been used for predicting the effect of actions on objects, whether physically or semantically. The ideal representation therefore should retain experiences to guide the robot's actions.

5.6. Component #6: Descriptive problem statement

Finally, the major difficulty in creating the ideal knowledge representation is the inability to define a robot's world precise enough to solve every possible problem. As we talked about before, logical expressions can be useful for grounding causal rules, while probabilistic expressions can help in answering ambiguities using beliefs. However, to vividly paint the picture of the robot's world in different structures and databases in this way really is no simple matter. A clear definition of the problem would help to identify those areas in robot manipulation and learning that we have done well enough and those areas that we still have a long way to go with. For instance, by clearly defining or constraining the environment in which a robot operates in, we can build tools or models that will function well even in variability. The idea of building more robust classifiers is mentioned in [158] through the idea of *open set recognition* [156,159,160], where the classification is redefined to account for robust detection of known positive classes while rejecting both known and unknown negative classes. The formulation of what the robot needs to identify versus what does not concern it for a particular action therefore is very important to clearly define.

5.7. Final analysis

Given that we keep the previously mentioned components in mind, we can develop more effective representations for robots. It is important that we as researchers consider how we can develop a robot that understands its environment and its own actions to induce smarter behavior. Instead of confining a robot to a single environment by programming it in a “hard-coded” fashion, through a properly designed knowledge representation, a robot can work in variants of the same environment it is trained to work among (for example, one robot trained to work in kitchens can recognize how a kitchen looks like and what it expects to find within a kitchen). Humans can map their own commands and understanding in that terminology, because we do not understand the world in low-level representations as used in machine learning algorithms or numbers for color values, depths, et cetera. As in AI, a robot will be able to intelligently gather new knowledge and make its own decisions and actions based on its experiences (or those of other robots). Several sub-tasks in robot problem solving can rely on machine learning models or tools to derive a solution, and we believe that these tools should adequately be used in unison with others. However, it is still important to derive representations filled with meaning, as implicit representations of knowledge can lead to unpredictable results, and machine learning models are not solutions for intelligent behavior in robots. An ongoing issue is the development of a standard for representations and ontology [161] and it is important to determine a common goal in the development of independent knowledge representations. This would eventually lead to promising developments for the future of robotics to integrate multiple platforms and to sustain cross-architecture communication and collaboration between robots through a common language, interface and/or network. In addition, we suggest that researchers focus more on building representations that can be built and accessed remotely through databases or cloud systems to make robot learning easier while reducing the strain on a robot’s on-board system to facilitate working in real-time scenarios, such as RoboEarth and openEASE. The ability to reproduce actions or sequences and the retention of experiences as memory would make training robots more efficient and less time consuming, since we do not necessarily have to learn tasks or skills from scratch.

6. Concluding remarks

To summarize, this paper provides an overview of technologies and research that is being used in robot learning and representation for service robots. To suitably define a knowledge representation, it is important to distinguish representation from learning. Machine learning models, for instance, are not sufficient to be representations alone and are better referred to as specialized learning models or classifiers. However, the effectiveness of such models should not be disregarded; rather, we should emphasize the *integration* of multiple components together with high-level knowledge to fully equip a robot with the means to understand its environment and actions to perform meaningful tasks. It is important to note that this paper has not talked about the efficiency of representation and retrieval algorithms, but instead this paper tackles the principles behind an effective knowledge representation for a robot to use to its fullest potential. Ideally, a knowledge representation should encompass all of the necessary information needed by a robot to solve a problem and understand its environment in order to do so effectively. It should allow the robot to successfully perform several actions such as perception/vision, knowledge acquisition, motion and task planning, and reasoning. However, due to the highly

variable nature of domestic environments and households as well as the question of safe operation among humans, building a suitable comprehensive knowledge representation becomes a very challenging and complex problem. In order to build suitable knowledge representations, we should strive to break this larger problem into smaller problem sets; we should take advantage of the many models that solve several sub-problems that a robot would typically be confronted with in manipulation problems. To further improve on the usefulness of knowledge representations, researchers can take advantage of large-scale distributed systems and networks for sharing knowledge and experiences that would help to teach robots how to do certain tasks from scratch without the need for reprogramming or rebuilding solutions. These distributed representations, such as those discussed in Section 2.3 can be hosted and built upon through the cloud or on the Internet. Although challenging and ambitious, a cloud computing interface for robots can greatly advance robotics technology. The most important takeaway from this paper is that there is always a need for standards to be developed and adopted by researchers in the realm of service robotics in order for safe and efficient robots to be deployed among humans.

Acknowledgments

We would like to acknowledge the National Science Foundation, United States for supporting our work under Grants No. 1421418 and 1560761. We would also like to thank our reviewers for their insightful and thorough comments that helped to greatly improve this manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] R.J. Brachman, H.J. Levesque, Knowledge Representation and Reasoning, 2003.
- [2] S. Harnad, The symbol grounding problem, *Physica D* 42 (1–3) (1990) 335–346.
- [3] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, H. Asoh, Symbol emergence in robotics: a survey, *Adv. Robot.* 30 (11–12) (2016) 706–728.
- [4] Y. Huang, M. Bianchi, M. Liarokapis, Y. Sun, Recent data sets on object manipulation: A survey, *Big data* 4 (4) (2016) 197–216.
- [5] A.B. Jelodar, M.S. Salekin, Y. Sun, Identifying Object States in Cooking-Related Images, 2018, arXiv preprint arXiv:1805.06956.
- [6] K. Ramirez-Amaro, T. Inamura, E. Dean-León, M. Beetz, G. Cheng, Bootstrapping humanoid robot skills by extracting semantic representations of human-like activities from virtual reality, in: *Humanoid Robots (Humanoids)*, 2014 14th IEEE-RAS International Conference on, IEEE, 2014, pp. 438–443.
- [7] K. Ramirez-Amaro, M. Beetz, G. Cheng, Understanding the intention of human activities through semantic perception: observation, understanding and execution on a humanoid robot, *Adv. Robot.* 29 (5) (2015) 345–362.
- [8] K. Ramirez-Amaro, M. Beetz, G. Cheng, Transferring skills to humanoid robots by extracting semantic representations from observations of human activities, *Artificial Intelligence* 247 (2017) 95–118.
- [9] S.J. Pan, Q. Yang, et al., A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.* 22 (10) (2010) 1345–1359.
- [10] Y. Yang, A. Guha, C. Fermuller, Y. Aloimonos, Manipulation action tree bank: A knowledge resource for humanoids, in: *Humanoid Robots (Humanoids)*, 2014 14th IEEE-RAS International Conference on, IEEE, 2014, pp. 987–992.
- [11] C.F. Yezhou Yang, E.E. Aksoy, Learning the Semantics of Manipulation Action, CoRR abs/1512.01525, 2015, URL <http://arxiv.org/abs/1512.01525>.
- [12] Y. Yang, A. Guha, C. Fermuller, Y. Aloimonos, A cognitive system for understanding human manipulation actions, *Adv. Cognit. Syst.* 3 (2014) 67–86.
- [13] Y. Yang, C. Fermuller, Y. Aloimonos, Detection of manipulation action consequences (MAC), in: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

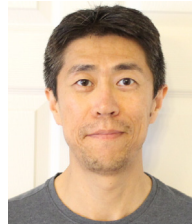
- [14] D. Paulius, Y. Huang, R. Milton, W.D. Buchanan, J. Sam, Y. Sun, Functional object-oriented network for manipulation learning, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2016, pp. 2655–2662, <http://dx.doi.org/10.1109/IIROS.2016.7759413>.
- [15] D. Paulius, A.B. Jelodar, Y. Sun, Functional object-oriented network: Construction & expansion, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 5935–5941, <http://dx.doi.org/10.1109/ICRA.2018.8460200>.
- [16] A.B. Jelodar, D. Paulius, Y. Sun, Long activity video understanding using functional object-oriented network, *IEEE Trans. Multimed.* (2018).
- [17] C.A. Petri, W. Reisig, Petri net, *Scholarpedia* 3 (4) (2008) 6477, revision 91646.
- [18] J. Gibson, The theory of affordances, in: R. Shaw, J. Bransford (Eds.), *Perceiving, Acting and Knowing*, Hillsdale, NJ: Erlbaum, 1977.
- [19] J.R. Anderson, *Rules of the mind*, Psychology Press, 2014.
- [20] J.E. Laird, A. Newell, P.S. Rosenbloom, Soar: An architecture for general intelligence, *Artif. Intell.* 33 (1) (1987) 1–64.
- [21] J.E. Laird, *The Soar Cognitive Architecture*, MIT press, 2012.
- [22] R. Sun, E. Merrill, T. Peterson, From implicit skills to explicit knowledge: a bottom-up model of skill learning, *Cognit. Sci.* 25 (2) (2001) 203–244.
- [23] D.E. Kieras, D.E. Meyer, An overview of the EPIC architecture for cognition and performance with application to human-computer interaction, *Human-Comput. Interact.* 12 (4) (1997) 391–438.
- [24] I. Kotseruba, J.K. Tsotsos, A review of 40 years of cognitive architecture research: Core cognitive abilities and practical applications, 2016, arXiv preprint arXiv:1610.08602.
- [25] M. Lungarella, G. Metta, R. Pfeifer, G. Sandini, Developmental robotics: a survey, *Connect. Sci.* 15 (4) (2003) 151–190.
- [26] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, C. Yoshida, Cognitive developmental robotics: A survey, *IEEE Trans. Autonomous Mental Dev.* 1 (1) (2009) 12–34.
- [27] H. Min, C. Yi, R. Luo, J. Zhu, S. Bi, Affordance research in developmental robotics: a survey, *IEEE Trans. Cognit. Dev. Syst.* 8 (4) (2016) 237–255.
- [28] J.E. Laird, K.R. Kinkade, S. Mohan, J.Z. Xu, Cognitive robotics using the soar cognitive architecture, in: *Workshops At the Twenty-Sixth AAAI Conference on Artificial Intelligence, Cognitive Robotics*, 2012.
- [29] P. Langley, D. Choi, A unified cognitive architecture for physical agents, in: *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1469.
- [30] K. Kim, J.-Y. Lee, D. Choi, J.-M. Park, B.-J. You, Autonomous task execution of a humanoid robot using a cognitive model, in: *Robotics and Biomimetics (ROBIO)*, 2010 IEEE International Conference on, IEEE, 2010, pp. 405–410.
- [31] K. Kim, D. Choi, J.-Y. Lee, J.-M. Park, B.-J. You, Controlling a humanoid robot in home environment with a cognitive architecture, in: *Robotics and Biomimetics (ROBIO)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 1754–1759.
- [32] D. Vernon, G. Metta, G. Sandini, The icub cognitive architecture: Interactive development in a humanoid robot, in: *Development and Learning*, 2007. IC DL 2007. IEEE 6th International Conference on, Ieee, 2007, pp. 122–127.
- [33] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. Von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, et al., The icub humanoid robot: An open-systems platform for research in cognitive development, *Neural Netw.* 23 (8) (2010) 1125–1134.
- [34] S. Ivaldi, N. Lyubova, D. G erardeaux-Viret, A. Droniou, S.M. Anzalone, M. Chetouani, D. Filliat, O. Sigaud, Perception and human interaction for developmental learning of objects and affordances, in: *Humanoid Robots (Humanoids)*, 2012 12th IEEE-RAS International Conference on, IEEE, 2012, pp. 248–254.
- [35] J.G. Trafton, L.M. Hiatt, A.M. Harrison, F.P. Tamborello II, S.S. Khemlani, A.C. Schultz, Act-r/e: An embodied cognitive architecture for human-robot interaction, *J. Human-Robot Interact.* 2 (1) (2013) 30–55.
- [36] S.M. Anzalone, S. Boucenna, S. Ivaldi, M. Chetouani, Evaluating the engagement with social robots, *Int. J. Social Robot.* 7 (4) (2015) 465–478.
- [37] S.L. Remy, M. Blake, Distributed service-oriented robotics, *IEEE Internet Comput.* 15 (2) (2011) 70–74.
- [38] E. Guizzo, Robots with their heads in the clouds, *IEEE Spectr.* 48 (3) (2011).
- [39] B. Kehoe, S. Patil, P. Abbeel, K. Goldberg, A survey of research on cloud robotics and automation., *IEEE Trans. Autom. Sci. Eng.* 12 (2) (2015) 398–409.
- [40] J. Oberlin, M. Meier, T. Kraska, S. Tellex, Acquiring Object Experiences at Scale.
- [41] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, D. Quillen, Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, *Int. J. Robot. Res.* (2016) 0278364917710318.
- [42] A. Schaffer, 10 breakthrough technologies 2016: Robots that teach each other, *MIT Technology Rev.* (2016).
- [43] M. Waibel, M. Beetz, J. Civera, R. d’Andrea, J. Elfring, D. Galvez-Lopez, K. H ussermann, R. Janssen, J. Montiel, A. Perzylo, et al., Roboearth, *IEEE Robot. Autom. Mag.* 18 (2) (2011) 69–82.
- [44] L. Riazuelo, M. Tenorth, D. Di Marco, M. Salas, D. G alvez-L opez, L. M osenlechner, L. Kunze, M. Beetz, J.D. Tard os, L. Montano, et al., Roboearth semantic mapping: A cloud enabled knowledge-based approach, *IEEE Trans. Autom. Sci. Eng.* 12 (2) (2015) 432–443.
- [45] M. Tenorth, M. Beetz, KnowRob: Knowledge processing for autonomous personal robots, in: *Intelligent Robots and Systems*, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, 2009, pp. 4261–4266.
- [46] M. Tenorth, M. Beetz, Knowrob: A knowledge processing infrastructure for cognition-enabled robots, *Int. J. Robot. Res.* 32 (5) (2013) 566–590.
- [47] L. Riazuelo, J. Civera, J. Montiel, C2tam: A cloud framework for co-operative tracking and mapping, *Robot. Auton. Syst.* 62 (4) (2014) 401–413.
- [48] D. Hunziker, M. Gajamohan, M. Waibel, R. D’Andrea, Rapyuta: The roboearth cloud engine, in: *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, IEEE, 2013, pp. 438–444.
- [49] G. Mohanarajah, D. Hunziker, R. D’Andrea, M. Waibel, Rapyuta: A cloud robotics platform, *IEEE Trans. Autom. Sci. Eng.* 12 (2) (2015) 481–493.
- [50] M. Beetz, M. Tenorth, J. Winkler, Open-EASE, in: *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on, IEEE, 2015, pp. 1983–1990.
- [51] M. Tenorth, M. Beetz, Representations for robot knowledge in the KnowRob framework, *Artificial Intelligence* 247 (2017) 151–169.
- [52] M. Beetz, D. Be ler, A. Haidu, M. Pomarlan, A.K. Bozcuo lu, G. Bartels, Knowrob 2.0a 2nd generation knowledge processing framework for cognition-enabled robotic agents, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 512–519.
- [53] D. Jain, L. Mosenlechner, M. Beetz, Equipping robot control programs with first-order probabilistic reasoning capabilities, in: *Robotics and Automation*, 2009. ICRA’09. IEEE International Conference on, IEEE, 2009, pp. 3626–3631.
- [54] R. Jain, T. Inamura, Bayesian Learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools, *Artif. Life Robot.* 18 (1–2) (2013) 95–103.
- [55] M. Beetz, L. M osenlechner, M. Tenorth, CRAMA cognitive robot abstract machine for everyday manipulation in human environments, in: *Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 1012–1017.
- [56] M. Beetz, D. Jain, L. Mosenlechner, M. Tenorth, L. Kunze, N. Blodow, D. Pangercic, Cognition-enabled autonomous robot control for the realization of home chore task intelligence, *Proc. IEEE* 100 (8) (2012) 2454–2471.
- [57] A. Saxena, A. Jain, O. Sener, A. Jami, D.K. Misra, H.S. Koppula, Robobrain: Large-scale knowledge engine for robots, arXiv preprint arXiv:1412.0691.
- [58] C. Fellbaum, *WordNet: An Electronic Lexical Database*, Bradford Books, 1998.
- [59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: *CVPR09*, 2009.
- [60] D. Koller, N. Friedman, L. Getoor, B. Taskar, Graphical models in a nutshell, *Introd. Statist. Relational Learn.* (2007) 13–55.
- [61] D. Koller, N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT press, 2009.
- [62] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: *Proceedings of the Eighteenth International Conference on Machine Learning*, in: *ICML ’01*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289, URL <http://dl.acm.org/citation.cfm?id=645530.655813>.
- [63] M. Richardson, P. Domingos, Markov Logic networks, *Mach. Learn.* 62 (1) (2006) 107–136, <http://dx.doi.org/10.1007/s10994-006-5833-1>.
- [64] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Robot programming by demonstration, in: *Springer Handbook of Robotics*, Springer, 2008, pp. 1371–1394.
- [65] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, Modeling affordances using bayesian networks, in: *Intelligent Robots and Systems*, 2007. IROS 2007. IEEE/RSJ International Conference on, IEEE, 2007, pp. 4102–4107.
- [66] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, Learning object affordances: from sensory-motor coordination to imitation, *IEEE Trans. Robot.* 24 (1) (2008) 15–26.
- [67] B. Moldovan, P. Moreno, M. van Otterlo, On the use of probabilistic relational affordance models for sequential manipulation tasks in robotics, in: *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, IEEE, 2013, pp. 1290–1295.
- [68] B. Moldovan, L. De Raedt, Learning relational affordance models for two-arm robots, in: *Intelligent Robots and Systems (IROS 2014)*, 2014 IEEE/RSJ International Conference on, IEEE, 2014, pp. 2916–2922.
- [69] B. Moldovan, P. Moreno, D. Nitti, J. Santos-Victor, L. De Raedt, Relational affordances for multiple-object manipulation, *Auton. Robots* (2017) 1–26.
- [70] F. Stramandinoli, V. Tikhonoff, U. Pattacini, F. Nori, Heteroscedastic regression and active learning for modeling affordances in humanoids, *IEEE Trans. Cognit. Dev. Syst.* (2017).

- [71] A. Stoytchev, Behavior-grounded representation of tool affordances, in: *Robotics and Automation*, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE, 2005, pp. 3060–3065.
- [72] J. Sinapov, A. Stoytchev, Learning and generalization of behavior-grounded tool affordances, in: *Development and Learning*, 2007. IC DL 2007. IEEE 6th International Conference on, IEEE, 2007, pp. 19–24.
- [73] V. Krunić, G. Salvì, A. Bernardino, L. Montesano, J. Santos-Victor, Affordance based word-to-meaning association, in: *Robotics and Automation*, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 4138–4143.
- [74] Y. Sun, S. Ren, Y. Lin, Object-object interaction affordance learning, *Robot. Auton. Syst.* (2013).
- [75] H. Kjellström, J. Romero, D. Kragic, Visual object-action recognition: Inferring object affordances from human demonstration, *Comput. Vis. Image Underst.* 115 (2010) 81–90.
- [76] H. Kjellström, J. Romero, D. Martínez, D. Kragić, Simultaneous visual recognition of manipulation actions and manipulated objects, *Comput. Vis.–ECCV 2008* (2008) 336–349.
- [77] C. Sutton, A. McCallum, K. Rohanimanesh, Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data, *J. Mach. Learn. Res.* 8 (Mar) (2007) 693–723.
- [78] A. Pieropan, C.H. Ek, H. Kjellström, Recognizing object affordances in terms of spatio-temporal object-object relationships, in: *2014 IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2014, pp. 52–58.
- [79] A. Pieropan, C.H. Ek, H. Kjellström, Functional object descriptors for human activity modeling, in: *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, IEEE, 2013, pp. 1282–1289.
- [80] H.S. Koppula, R. Gupta, A. Saxena, Learning human activities and object affordances from rgb-d videos, *Int. J. Robot. Res.* 32 (8) (2013) 951–970.
- [81] H.S. Koppula, A. Saxena, Anticipating human activities using object affordances for reactive robotic response, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (1) (2016) 14–29.
- [82] Y. Zhu, A. Fathi, L. Fei-Fei, Reasoning about object affordances in a knowledge base representation, in: *European Conference on Computer Vision*, Springer, 2014, pp. 408–424.
- [83] E. Vassev, M. Hinchey, Knowledge representation for cognitive robotic systems, in: *2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, IEEE, 2012, pp. 156–163.
- [84] E. Vassev, M. Hinchey, Knowlang: Knowledge representation for self-adaptive systems., *IEEE Comput.* 48 (2) (2015) 81–84.
- [85] M.S. Ryoo, Human activity prediction: Early recognition of ongoing activities from streaming videos, in: *Computer Vision (ICCV)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 1036–1043.
- [86] M. Ryoo, T.J. Fuchs, L. Xia, J.K. Aggarwal, L. Matthies, Robot-centric activity prediction from first-person videos: What will they do to me?, in: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, ACM, 2015, pp. 295–302.
- [87] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Mark Siskind, S. Wang, Recognize human activities from partially observed videos, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2658–2665.
- [88] C. Vondrick, H. Pirsiavash, A. Torralba, Anticipating the future by watching unlabeled video, *arXiv preprint arXiv:1504.08023*.
- [89] E.E. Aksoy, A. Abramov, F. Wörgötter, B. Dellen, CaTegorizing object-action relations from semantic scene graphs, in: *Robotics and Automation (ICRA)*, 2010 IEEE International Conference on, IEEE, 2010, pp. 398–405.
- [90] E.E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, F. Wörgötter, Learning the semantics of object-action relations by observation, *Int. J. Robot. Res.* 30 (10) (2011) 1229–1249.
- [91] M. Sridhar, A.G. Cohn, D.C. Hogg, Learning functional object categories from a relational spatio-temporal representation, in: *ECAI 2008: 18th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications)*, IOS Press, 2008, pp. 606–610.
- [92] Y. Zhu, Y. Zhao, S. Chun Zhu, Understanding tools: Task-oriented object modeling, learning and recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2855–2864.
- [93] H. Costelha, P. Lima, Modelling, analysis and execution of robotic tasks using petri nets, in: *Intelligent Robots and Systems*, 2007. IROS 2007. IEEE/RSJ International Conference on, IEEE, 2007, pp. 1449–1454.
- [94] H. Costelha, P. Lima, Robot task plan representation by petri nets: modelling, identification, analysis and execution, *Auton. Robots* 33 (4) (2012) 337–360.
- [95] C. Geib, K. Mourao, R. Petrick, N. Pugeault, M. Steedman, N. Krueger, F. Wörgötter, Object action complexes as an interface for planning and robot control, in: *IEEE RAS International Conference on Humanoid Robots*, 2006.
- [96] R. Petrick, D. Kraft, K. Mourao, N. Pugeault, N. Krüger, M. Steedman, Representation and integration: Combining robot control, high-level planning, and action learning, in: *Proceedings of the 6th International Cognitive Robotics Workshop*, Citeseer, 2008, pp. 32–41.
- [97] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, et al., Object-action complexes: Grounded abstractions of sensory-motor processes, *Robot. Auton. Syst.* 59 (10) (2011) 740–757.
- [98] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, F. Wörgötter, R. Dillmann, Action sequence reproduction based on automatic segmentation and object-action complexes, in: *Humanoid Robots (Humanoids)*, 2013 13th IEEE-RAS International Conference on, IEEE, 2013, pp. 189–195.
- [99] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, N. Krüger, Early reactive grasping with second order 3D feature relations, in: *Recent Progress in Robotics: Viable Robotic Service to Human*, Springer, 2007, pp. 91–105.
- [100] G.D. Konidaris, S. Kuindersma, R. Grupen, A. Barto, Robot learning from demonstration by constructing skill trees, *Int. J. Robot. Res.* 31 (3) (2012) 360–375.
- [101] R.B. Rusu, Z.C. Marton, N. Blodow, A. Holzbach, M. Beetz, Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments, in: *Intelligent Robots and Systems*, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, 2009, pp. 3601–3608.
- [102] D. Pangercic, B. Pitzer, M. Tenorth, M. Beetz, Semantic object maps for robotic housework-representation, acquisition and use, in: *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 4644–4651.
- [103] H. Dang, P.K. Allen, Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task, in: *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 1311–1317.
- [104] H. Dang, P.K. Allen, Semantic grasping: planning task-specific stable robotic grasps, *Auton. Robots* 37 (3) (2014) 301–316.
- [105] C. Galindo, J.-A. Fernández-Madriral, J. González, A. Saffiotti, Robot task planning using semantic maps, *Robot. Autonomous Syst.* 56 (11) (2008) 955–966.
- [106] E. Bastianelli, D. Bloisi, R. Capobianco, G. Gemignani, L. Iocchi, D. Nardi, Knowledge representation for robots through human-robot interaction, *arXiv preprint arXiv:1307.7351*.
- [107] T. Kollar, V. Perera, D. Nardi, M. Veloso, Learning environmental knowledge from task-based human-robot dialog, in: *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, IEEE, 2013, pp. 4304–4309.
- [108] G. Randelli, T.M. Bonanni, L. Iocchi, D. Nardi, Knowledge acquisition through human-robot multimodal interaction, *Intell. Serv. Robot.* 6 (1) (2013) 19–31.
- [109] Y. Yang, C. Fermüller, Y. Li, Y. Aloimonos, Grasp type revisited: A modern perspective on a classical feature for vision, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 400–408.
- [110] N. Dantam, P. Kolhe, M. Stilman, The motion grammar for physical human-robot games, in: *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 5463–5469.
- [111] N. Dantam, M. Stilman, The motion grammar: Analysis of a linguistic method for robot control, *IEEE Trans. Robot.* 29 (3) (2013) 704–718.
- [112] A. Myers, C.L. Teo, C. Fermüller, Y. Aloimonos, Affordance detection of tool parts from geometric features, in: *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on, IEEE, 2015, pp. 1374–1381.
- [113] J. Peters, S. Vijayakumar, S. Schaal, Reinforcement learning for humanoid robotics, in: *Proceedings of the Third IEEE-RAS International Conference on Humanoid Robots*, 2003, pp. 1–20.
- [114] J. Kober, J. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, *Int. J. Robot. Res.* 32 (11) (2013) 1238–1274.
- [115] M.P. Deisenroth, G. Neumann, J. Peters, et al., A survey on policy search for robotics, *Found. Trends. Robot.* 2 (1–2) (2013) 1–142.
- [116] B.D. Argall, S. Chernova, et al., A survey of robot learning from demonstration, *Robot. Auton. Syst.* 57 (5) (2009) 469–483.
- [117] J.R. Peters, Machine Learning of Motor Skills for Robotics (Ph.D. thesis), University of Southern California, 2007.
- [118] J. Kober, J. Peters, Learning motor primitives for robotics, in: *Robotics and Automation*, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 2112–2118.
- [119] E. Ugur, E. Şahin, E. Oztop, Self-discovery of motor primitives and learning grasp affordances, in: *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 3260–3267.
- [120] V. Tikhonoff, U. Pattacini, L. Natale, G. Metta, Exploring affordances and tool use on the icub, in: *Humanoid Robots (Humanoids)*, 2013 13th IEEE-RAS International Conference on, IEEE, 2013, pp. 130–137.
- [121] E. Ugur, S. Szedmak, J. Piater, Bootstrapping paired-object affordance learning with learned single-affordance features, in: *Development and Learning and Epigenetic Robotics (ICDL-Epirob)*, 2014 Joint IEEE International Conferences on, IEEE, 2014, pp. 476–481.
- [122] E. Ugur, J. Piater, Emergent structuring of interdependent affordance learning tasks, in: *Development and Learning and Epigenetic Robotics (ICDL-Epirob)*, 2014 Joint IEEE International Conferences on, IEEE, 2014, pp. 489–494.

- [123] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural Comput.* 1 (4) (1989) 541–551.
- [124] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324.
- [125] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.
- [126] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*.
- [127] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: *European Conference on Computer Vision*, Springer, 2014, pp. 818–833.
- [128] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (5) (1989) 359–366.
- [129] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [130] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
- [131] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Netw.* 61 (2015) 85–117.
- [132] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, *Neurocomputing* 234 (2017) 11–26.
- [133] J. Bohg, A. Morales, T. Asfour, D. Kragic, Data-driven grasp synthesis survey, *IEEE Trans. Robot.* 30 (2) (2014) 289–309.
- [134] J. Mahler, F.T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, K. Goldberg, Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards, in: *Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, 2016*, pp. 1957–1964.
- [135] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J.A. Ojea, K. Goldberg, Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics, *arXiv preprint arXiv:1703.09312*.
- [136] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, K. Goldberg, Dex-Net 3.0: Computing Robust Robot Suction Grasp Targets in Point Clouds using a New Analytic Model and Deep Learning, *arXiv preprint arXiv:1709.06670*.
- [137] I. Lenz, H. Lee, A. Saxena, Deep learning for detecting robotic grasps, *Int. J. Robot. Res.* 34 (4–5) (2015) 705–724.
- [138] J. Chen, C.-W. Ngo, Deep-based ingredient recognition for cooking recipe retrieval, in: *Proceedings of the 2016 ACM on Multimedia Conference, ACM, 2016*, pp. 32–41.
- [139] Y. Zhang, M. Bai, P. Kohli, S. Izadi, J. Xiao, Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1192–1201.
- [140] S. Aditya, Y. Yang, C. Baral, Y. Aloimonos, C. Fermüller, Image understanding using vision and reasoning through scene description graph, *Comput. Vis. Image Underst.* 173 (2018) 33–45.
- [141] A. Nguyen, D. Kanoulas, D.G. Caldwell, N.G. Tsagarakis, Detecting object affordances with convolutional neural networks, in: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, IEEE, 2016*, pp. 2765–2770.
- [142] A. Srikantha, J. Gall, Weakly Supervised Learning of Affordances, *arXiv preprint arXiv:1605.02964*.
- [143] T.-T. Do, A. Nguyen, I. Reid, Affordancenet: An end-to-end deep learning approach for object affordance detection, in: *International Conference on Robotics and Automation (ICRA)*, 2018.
- [144] S. Cascianelli, G. Costante, T.A. Giarfuglia, P. Valigi, M.L. Fravolini, Full-GRU natural language video description for service robotics applications, *IEEE Robot. Automat. Lett.* 3 (2) (2018) 841–848, <http://dx.doi.org/10.1109/LRA.2018.2793345>.
- [145] S. Gupta, R. Girshick, P. Arbeláez, J. Malik, Learning rich features from RGB-d images for object detection and segmentation, in: *European Conference on Computer Vision*, Springer, 2014, pp. 345–360.
- [146] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [147] N. Dvornik, K. Shmelkov, J. Mairal, C. Schmid, Blitznet: A real-time deep network for scene understanding, in: *ICCV 2017-International Conference on Computer Vision*, 2017, p. 11.
- [148] Y. Yang, Y. Li, C. Fermüller, Y. Aloimonos, Robot learning manipulation action plans by “watching” unconstrained videos from the world wide web, in: *AAAI*, 2015, pp. 3686–3693.
- [149] J. Sung, S.H. Jin, A. Saxena, Robobarista: Object part based transfer of manipulation trajectories from crowd-sourcing in 3d pointclouds, in: *Robotics Research*, Springer, 2018, pp. 701–720.
- [150] C. Fermüller, F. Wang, Y. Yang, K. Zampogiannis, Y. Zhang, F. Barranco, M. Pfeiffer, Prediction of manipulation actions, *Int. J. Comput. Vis.* 126 (2–4) (2018) 358–374.
- [151] J. Wu, I. Yildirim, J.J. Lim, B. Freeman, J. Tenenbaum, Galileo: Perceiving physical object properties by integrating a physics engine with deep learning, in: *Advances in Neural Information Processing Systems*, 2015, pp. 127–135.
- [152] C. Finn, I. Goodfellow, S. Levine, Unsupervised learning for physical interaction through video prediction, in: *Advances in Neural Information Processing Systems*, 2016, pp. 64–72.
- [153] P. Battaglia, R. Pascanu, M. Lai, D.J. Rezende, et al., Interaction networks for learning about objects, relations and physics, in: *Advances in Neural Information Processing Systems*, 2016, pp. 4502–4510.
- [154] I. Yildirim, T. Gerstenberg, B. Saeed, M. Toussaint, J. Tenenbaum, Physical problem solving: Joint planning with symbolic, geometric, and dynamic constraints, *arXiv preprint arXiv:1707.08212*.
- [155] A. Byravan, D. Fox, Se3-nets: Learning rigid body motion using deep neural networks, in: *Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE, 2017*, pp. 173–180.
- [156] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, et al., The limits and potentials of deep learning for robotics, *Int. J. Robot. Res.* 37 (4–5) (2018) 405–420, <http://dx.doi.org/10.1177/0278364918770733>.
- [157] M. Tenorth, D. Jain, M. Beetz, Knowledge processing for cognitive robots, *KI-Künstliche Intell.* 24 (3) (2010) 233–240.
- [158] W.J. Scheirer, L.P. Jain, T.E. Boulton, Probability models for open set recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (11) (2014) 2317–2324.
- [159] A. Bendale, T. Boulton, Towards open world recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1893–1902.
- [160] A. Bendale, T.E. Boulton, Towards open set deep networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1563–1572.
- [161] E. Prestes, J.L. Carbonera, S.R. Fiorini, V.A. Jorge, M. Abel, R. Madhavan, A. Loco, P. Goncalves, M.E. Barreto, M. Habib, et al., Towards a core ontology for robotics and automation, *Robot. Auton. Syst.* 61 (11) (2013) 1193–1204.



David Paulius received his B.S. degree from the University of the Virgin Islands, St. Thomas, VI, in 2014. Since August 2014, he has been a Ph.D. student in the Department of Computer Science and Engineering at the University of South Florida and is a graduate researcher in the Robot Perception and Action Lab (RPAL). He is soon due to complete his Ph.D. His research interests include robotics, knowledge representations, artificial intelligence, and graph theory.



Dr. Yu Sun is an Associate Professor and Associate Chair of Graduate Affairs in the Department of Computer Science and Engineering at the University of South Florida. He was a Visiting Associate Professor at Stanford University. He was the recipient of the 2018 USF Excellence in Innovation Award. He has served as reviewer/panelist for the U.S. National Science Foundation, the French National Research Agency, the Kingdom of Saudi Arabia RDO, and IEEE Admission & Advancement (A&A). He has served as Senior Editor or Associate Editor for several journals and conferences

such as *IEEE Transactions on Robotics*, *ICRA*, and *IROS*. He has published numerous papers and holds nine US patents in intelligent systems, robotics, deep learning, computer vision, virtual reality, and medical applications.