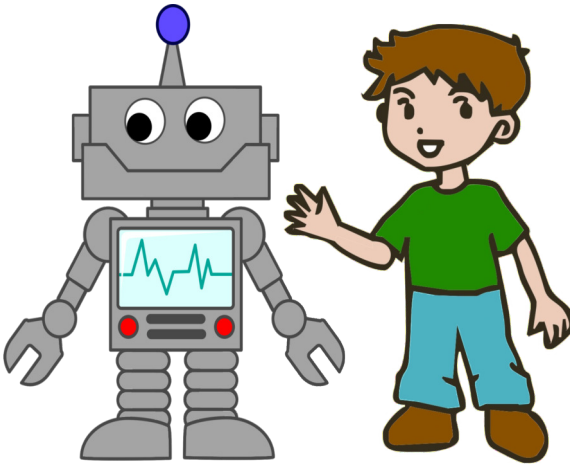
TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



Gestures and Object Affordances for Human–Robot Interaction

Giovanni Saponaro

Supervisor: Doctor Alexandre José Malheiro Bernardino
Co-Supervisor: Doctor Giampiero Salvi
Co-Supervisor: Doctor Lorenzo Jamone

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with Distinction

2019

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Gestures and Object Affordances for Human–Robot Interaction

Giovanni Saponaro

Supervisor: Doctor Alexandre José Malheiro Bernardino
Co-Supervisor: Doctor Giampiero Salvi
Co-Supervisor: Doctor Lorenzo Jamone

Thesis approved in public session to obtain the PhD Degree in
Electrical and Computer Engineering

Jury final classification: Pass with Distinction

Jury

Chairperson: Doctor João Manuel Lage de Miranda Lemos, Instituto Superior Técnico, Universidade de Lisboa
Members of the Committee:
Doctor José Alberto Rosado dos Santos Vítor, Instituto Superior Técnico, Universidade de Lisboa
Doctor Ana Maria Severino de Almeida e Paiva, Instituto Superior Técnico, Universidade de Lisboa
Doctor Luís Filipe de Seabra Lopes, Universidade de Aveiro
Doctor Alexandre José Malheiro Bernardino, Instituto Superior Técnico, Universidade de Lisboa
Doctor Emre Uğur, Engineering Faculty, Boğaziçi University, Turkey

Funding Institutions:
Fundação para a Ciência e a Tecnologia
European Commission

2019

ABSTRACT

Personal and service robots may benefit society in different activities and challenges, thanks to their increasingly advanced mechanical and decision-making capabilities. However, for that to happen one of the essential conditions is to have coherent, natural and intuitive interfaces so that humans can make a fruitful and effective use of these intelligent machines. In general, the goal of reaching intuitive interfaces for human–robot interaction has not yet been attained, partly due to the fact that robots are not yet widespread in public spaces, partly due to the technical difficulties in interpreting human intentions.

Making service robots that understand their surroundings entails that they should possess capabilities that allow them to operate in unstructured environments and under unpredictable conditions, unlike industrial robots which usually operate in highly structured, controlled and repeatable environments. To tackle these challenges, in this thesis we develop computational models based on findings from developmental psychology (object affordances) and from neuroscience (mirror neuron system). The proposed models stem from the consideration that objects carry information about actions and interactions.

We contribute a modular framework for visual robot affordance learning, based on autonomous robot exploration of the world, sensorimotor data collection and statistical models. We combine affordances with communication (gestures and language), to interpret and describe human actions in manipulative scenes by reusing previous robot experience. We show a model that deals with multiple objects, giving rise to tool use, including the link from hand affordances (i.e., action possibilities by using the hands) to tool affordances (i.e., action possibilities by using tools). We illustrate how affordances can be used for planning complex manipulation tasks under noise and uncertainty. In two appendixes, we describe a robot model for recognizing human gestures in manipulation scenarios, and we report a study about how people perceive robot gestures when the facial information is turned off.

Keywords: gestures, object affordances, human–robot interaction, iCub robot, machine learning

RESUMO

Os robôs pessoais e de serviço podem beneficiar a sociedade em atividades e desafios diferentes, graças às suas capacidades mecânicas e de tomadas de decisão cada vez mais avançadas. No entanto, para isso acontecer, uma das condições essenciais é a existência de interfaces coerentes, naturais e intuitivas para que os humanos possam usar estas máquinas inteligentes de uma forma frutuosa e eficaz. No geral, o objetivo de alcançar interfaces intuitivas para a interação homem–robô (*human–robot interaction*) não foi ainda alcançado, em parte pelo facto de que os robôs não estão ainda difusos nos espaços públicos, em parte devido às dificuldades técnicas na interpretação das intenções humanas.

Construir robôs de serviço, que percebam o que os rodeia, implica que os mesmos possuam capacidades que lhes permitam funcionar em ambientes não estruturados e em condições imprevisíveis, ao contrário dos robôs industriais que operam em ambientes altamente estruturados, controlados e repetíveis. Para enfrentar estes desafios, nesta tese desenvolvemos modelos computacionais baseados em descobertas da psicologia do desenvolvimento (potencialidades de objetos, *object affordances*) e da neurociência (sistema de neurónios espelho). Os modelos propostos resultam da consideração que os objetos transportam informação sobre ações e interações.

Propomos uma biblioteca de *software* modular para aprendizagem de potencialidades visuais em robôs, baseada na exploração autónoma do mundo, recolha de dados sensório-motores e técnicas estatísticas. Juntamos potencialidades com comunicação (gestos e linguagem), para interpretar e descrever ações humanas em cenários de manipulação, reutilizando a experiência prévia do robô. Mostramos um modelo que lida com objetos múltiplos, permitindo o uso de ferramentas, e também a ligação de potencialidades de mãos (i.e., possibilidade de ações usando as mãos) para potencialidades de ferramentas (i.e., possibilidade de ações usando ferramentas). Explicamos como as potencialidades podem ser usadas para o planeamento de ações complexas de manipulação, sob ruído e incerteza. Em dois apêndices, descrevemos um modelo robótico para reconhecer gestos humanos em contextos de manipulação, e relatamos um estudo de como as pessoas percebem gestos robóticos quando a informação facial dos mesmos é desligada.

Palavras-chave: gestos, potencialidades de objetos (*object affordances*), interação homem–robô, robô iCub, aprendizagem automática

ACKNOWLEDGMENTS

First of all, I would like to give a warm thank you to my main supervisor, Prof. Alexandre Bernardino, for his constant guidance during the years that led to the preparation of this work, and for encouraging me to gain my independence. My co-supervisors, Prof. Giampiero Salvi and Prof. Lorenzo Jamone, have also supported me countless times with scientific insights, challenges, and suggestions, for which I am grateful.

I extend my gratitude to Prof. José Santos-Victor, director of the Computer and Robot Vision Laboratory (VisLab) in Lisbon, for making me feel welcome, for making me feel an important "player" in a research team, and for creating an environment where it is possible to conduct research without worries. Thank you to all my laboratory mates as well (too many to list!).

This research has been made possible with funding from the Portuguese Government (Fundação para a Ciência e a Tecnologia, doctoral grant SFRH/BD/61910/2009, project grants PEst-OE/EEI/LA0009/2011 and UID/EEA/50009/2013) and from the European Commission (POETICON++ project, FP7-ICT-288382).

The picture on the thesis cover was elaborated by me, modifying two open-access clipart images available from `https://openclipart.org`, respectively drawn by Sirrob01 (robot image) and by an anonymous user (boy image).

And finally, a heartfelt thanks to Cláudia for her unconditional patience and support all along.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI Artificial Intelligence

ASR Automatic Speech Recognition

AI Artificial Intelligence

BDE Bayesian Dirichlet likelihood-equivalence

BN Bayesian Network

CFG Context-Free Grammar

CPD Conditional Probability Distribution

CRF Conditional Random Field

DAG Directed Acyclic Graph

DOF Degree of Freedom

DMP Dynamic Movement Primitive

EM Expectation–Maximization

GMM Gaussian Mixture Model

GRT Gesture Recognition Toolkit

GT Ground Truth

HMM Hidden Markov Model

HPN Hierarchy Planning in the Now

ISO International Organization for Standardization

LBP Local Binary Pattern

LOC Lines of Code

MAP Maximum A Posteriori

MCMC Markov Chain Monte Carlo

MRF Markov Random Field

NLU Natural Language Understanding

OAC Object–Action Complex

PCA Principal Component Analysis

PDDL Planning Domain Definition Language

PDF Probability Density Function

RNN Recurrent Neural Network

ROS Robot Operating System

SIFT Scale-Invariant Feature Transform

YARP Yet Another Robot Platform

1

MOTIVATION AND SCOPE OF THE THESIS

In recent years, it is becoming increasingly common to use robots in domestic and public spaces. The total number of professional service robots sold in 2016 (i.e., non-industrial robots) rose considerably by 24% to 59 706 units, up from 48 018 in 2015, with similarly positive forecasts expected for the period until 2020[1]. When deployed in domestic and public environments, robotic machines are expected to take on roles such as personal home assistants, receptionists, waiters, couriers, and more. It is now feasible, though not without problems, to think of social robots being located in the same physical areas as a person. This societal shift bears the issue of how to make robots that work effectively alongside humans. In other words, how to build robots that possess the capabilities (e.g., perception, reasoning, action) to execute their tasks well, and that in doing that are robust and reactive to uncertainty (e.g., noisy measurements) and unexpected events (e.g., failures), and that collaborate with us adequately by assisting our activities, without being a costly encumbrance in terms of money, time, or patience. Public scenarios require interfaces that are easy to use for the general public, including for special groups like disabled, elderly or technology challenged people. Human users should be able to provide instructions to robots in a natural and effortless way, mainly with verbal language and with nonverbal communication (e.g., with body gestures), but this task has not been attained in general. This thesis contributes to bridging the usability gap that human users face when dealing with robots.

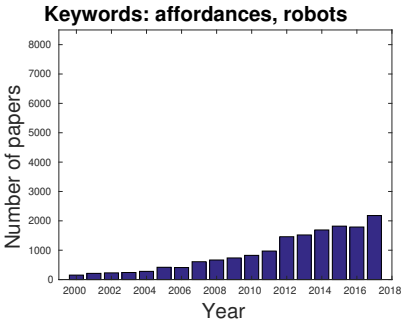
One of the open challenges in designing robots that operate successfully in the *unpredictable* human environment is how to make them able to foresee what actions they can perform onto objects of the world, and what the effects of these actions will be: in other words, how to provide them with the ability to perceive object *affordances* (action possibilities), a concept originally introduced in the field of developmental psychology in the 1960s, and of increasing importance in robotic research (see Fig. 1a). First proposed by J. Gibson [Gib14], an affordance is defined as: "a resource that the environment offers any animal that has the capabilities to perceive and use it". Later, E. Gibson studied the role of affordances and learning in children [Gib03], reflecting on "discovering the information that specifies an affordance". These theories stress how *interacting* with the environment (i.e., acting on it with

[1] Executive Summary World Robotics 2017 Service Robots, `http://www.ifr.org/service-robots/statistics/`

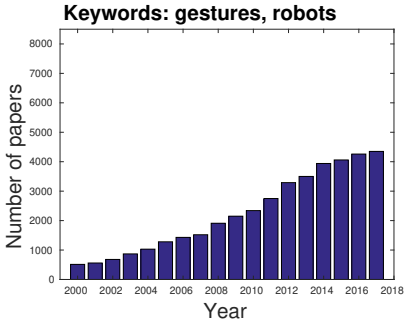
a body) and *perceiving* the environment (i.e., sensing relevant features and changes of the world) are interconnected and related.

Suppose that a robotic agent has to operate in an environment, in particular having to see and use the objects that are available in order to achieve a given goal, such as adding sugar to a coffee (*stirring the coffee*) in the presence of a cup with coffee, of a sugar bowl, and of a spoon (ignoring, for simplicity, the aspect of how the goal was entered into the system by a human user). Classical Artificial Intelligence (AI) and robotic systems [RN09; SK16] will permit the agent to achieve the goal, relying on perceptual sensing algorithms, symbolic planning, and robot manipulator control, *provided that the objects are recognized correctly*, meaning that the "ingredients" needed for the task (e.g., cup, bowl, spoon) have been previously learned by the system at training time, and are detected at testing time. However, what happens when a spoon is not available, or its appearance and shape are considerably different from the spoons that the system was taught? Classical AI may not be able to cope with this scenario, whereas the incorporation of affordances can help filling in the gaps in the following sense. Reasoning on the affordances of the objects gives the benefit of relying on the knowledge about an object's *functional features* or sub-parts, rather than on knowing the object's name or identity. In the coffee example, if no spoon is available, the agent could use another type of cutlery which is long and thin, but if even that is not available, it might use yet another object that would help stir the coffee (e.g., an object with a thin and elongated shape, as in the upper part of Fig. 5). Affordances are mental shortcuts for accessing properties of objects that lead to a goal-directed action, without having to explicitly recognize the object name or type. Therefore, affordances are a means to *generalization* in robotic perception.

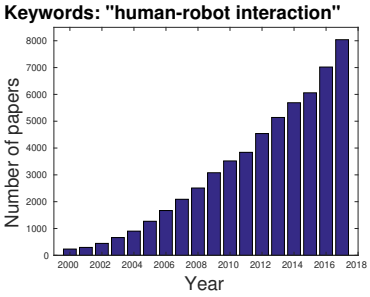
A similar argumentation can be made when we consider instructions provided by a human user to another agent (human or robot). In interactions, too, *objects carry key information* about the action, which is conveyed explicitly or implicitly. Humans can instruct other agents to perform operations on objects by saying their explicit name out loud (using a shared verbal language which is understood by partners), or by referring to their distinctive features in an ambiguous situation (e.g., by saying "the large green box"), or by pointing at the objects themselves (using a shared nonverbal gesture while communicating with an interaction partner). Interestingly, when a person makes a gesture that points in the direction of an object, a system can reason on the affordances of the pointed object, which is useful for *action prediction* (i.e., predicting what the person will do next, or intends to do). Indeed, when a person refers (in whichever way) to an object, they are often really referring to the useful properties of it. All in all, we can summarize these examples by saying that an object can act as a *mediator for interaction*. Affordances are one way to exploit this power.



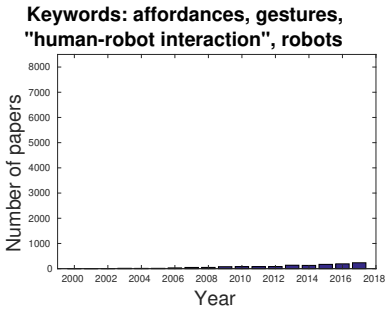
(a) Number of published papers including "affordances" and "robots" in their text.



(b) Number of published papers including "gestures" and "robots" in their text.



(c) Number of published papers including the sentence "human–robot interaction" in their text.



(d) Number of published papers including all of the keywords "affordances", "gestures", "human–robot interaction" and "robots" in their text.

Figure 1: Number of published papers including relevant thesis keywords over time. The searches in Fig. 1a–1c show a growing trend in terms of number of publications for individual topics, however Fig. 1d reveals that not many papers address all the considered topics jointly. Plots computed from Google Scholar data, using the `academic-keyword-occurrence` script by Volker Strobel[2].

Within robotic research, in addition to a growing interest in affordances as mentioned above, there has been a similar trend in topics such as body gestures (see Fig. 1b) and, in general, in human–robot interaction systems and studies (see Fig. 1c). However, there are not many papers yet that address these topics *jointly*, as seen in Fig. 1d: the number of published papers which mention all of the keywords (in the entirety of the article text, not only in the title) ranges between 3 in year 2000 and about 180 in 2016. Even though 180 papers is a reasonable number, it still constitutes a relatively small fraction of the whole body of articles produced by the robotics community yearly.

[2] `http://doi.org/10.5281/zenodo.1218409`

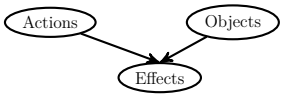
1.1 OBJECTIVES

This thesis revolves around the usefulness of affordances in robots, and the possible *advantages of using robot affordances in conjunction with other modalities, such as human body gestures and language, for supporting effective interactions between humans and robots.* The core goal is to develop computational models to use affordances and other environment elements in order to close the gap between human and robot knowledge. We research how object affordances can provide a *joint reference* between human and robot for the correct interpretation of gestural instructions, and how this can ultimately lead to intuitive robot utilization for the general population. In particular, this work stems from the observation that objects contain important links to physical actions and action understanding when interacting with other agents. From this idea, we develop a theory comprising objects affordances, body gestures and probabilistic inference. We contribute software programs that can be deployed on real robotic systems. We show a number of practical contributions in robot algorithms that incorporate the above concepts.

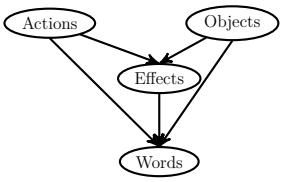
By advancing action recognition capabilities in robots through the combination of gestures and affordance perception, we also endow robots with the ability to recognize human actions during their enactment, i.e., before said actions are completed in their entirety. The advantage is to provide robots the ability to *anticipate* human actions and intentions, given contextual circumstances. We show this anticipatory behavior and capability in human–robot collaborative tasks, relying on the information provided by objects affordances, body gestures and probabilistic inference.

Fig. 2 illustrates the cognitive robotic models used in this thesis schematically. Fig. 2a exemplifies the state of the art in robot affordances: a computational implementation of simple object affordances, defined as the relations between actions, objects and effects, as proposed in the works of Montesano [Mon+08]. Fig. 2b is also prior work from the state of the art, focusing on the joint learning of affordances and language descriptions. Fig. 2c shows our contribution of incorporating gestures and language descriptions to the model, allowing a cognitive robot to reason about physical actions when external agents operate in an environment shared with the robot, to describe the scene verbally, and permitting anticipatory behavior. Fig. 2d depicts our contribution related to tool use, which allow a robot to not only use a single object of the world, but to reason about the possibilities offered by a first grasped object (i.e., using a specific manipulator) onto a second object which is acted upon when the first one is in the agent's hand.

In the following sections, we present the constituent theoretical components upon which this thesis is based. Sec. 1.2 defines the princi-

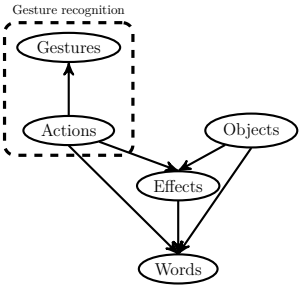


(a) Computational model of robot affordances by Montesano [Mon+08], explained in Ch. 2.

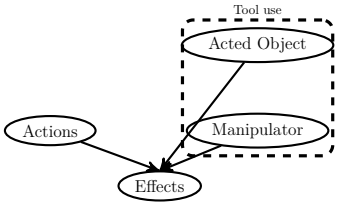


(b) Computational model of affordances with associated verbal descriptions, by Salvi [Sal+12], explained in Ch. 2.

state of the art
our contributions



(c) Computational model of affordances with gestures and language, proposed in Ch. 4.



(d) Computational model of affordances for dealing with multiple objects and tool use, proposed in Ch. 5.

Figure 2: Schematic diagrams of the cognitive robotic models discussed in this thesis. They all include a representation of affordances as relations between actions, objects and effects (Figs. 2a–2d). In addition, some of them include extensions or specific aspects being highlighted (Figs. 2c–2d). Figs. 2a and 2b are prior existing works from the state of the art, whereas Figs. 2c and 2d are contributions of this thesis.

ples that guide the framework of developmental robotics, Sec. 1.3 illustrates the concept of affordances and the advantages that it provides in robotics, and Sec. 1.4 specifies the neuroscience theories which are linked to our research. Finally, Sec. 1.5 lists the main contributions of the thesis, and Sec. 1.6 gives a brief outline of the structure of the next chapters.

1.2 DEVELOPMENTAL ROBOTICS

Developmental robotics, also known as epigenetic robotics or ontogenetic robotics, is a subfield of robotics whose main aims are (i) modeling the development of *increasingly complex cognitive processes* (for example, the understanding of language, or the acquisition of manipulation skills), and also (ii) understanding how such processes emerge through physical and social interaction [Lun+03; CS15]. Developmental robotics takes direct inspiration from the progressive learning phenomena observed in children's cognitive development. It is related to other fields such as Artificial Intelligence (AI), developmental psychology, neuroscience and dynamical systems theory.

In this line of research, robots are used to verify theoretical models of emergence and development of action and cognition. The rationale is the following: if a model is instantiated inside a system embedded in the real world, many things can be learned about its strengths and limitations. Developmental robotics operates on short (ontogenetic) time scales of single individuals, or small groups of individuals. By contrast, evolutionary robotics typically operates on long (phylogenetic) time scales and large populations of several individuals.

The basic idea behind developmental robotics (i.e., that the mechanism of development can be used to understand and to construct cognition) can be traced back to Turing [Tur50]: "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education, one would obtain the adult brain". Another core idea in developmental robotics is *embodiment* (or Embodied Artificial Intelligence), which states that intelligence (e.g., common sense) can only be the result of learned experience of a body living in the real world [PB06].

This thesis follows the developmental robotics perspective: the contributions that we will describe in the next chapters hinge on the paradigm of a robot that learns about its surrounding environment by incremental *self-exploration*, starting from a limited initial knowledge; in addition, this learning process is embodied, in the sense that it is conditioned on a particular, physical robot body (e.g., the arms and hands can reach and manipulate certain objects and locations in the robot workspace, but not all of them).

Figure 3: A door handle affords the ability to be turned and pulled, resulting in the door to be open.

1.3 MOTIVATION FOR USING ROBOT AFFORDANCES

Affordances, introduced in p. 1, correspond to action possibilities offered to agents by elements of the environment. They can support service robotics[3] and human–robot interaction applications for a number of reasons.

First, affordances are *personal*: they depend on the agent (or on the "animal", as in the original definition by J. Gibson [Gib14]). For example, the door handle of Fig. 3 offers the affordance of being manipulated in order to open the door, but the precise motor realization of the act of turning the handle is different for an adult human or for a robot (and also for other types of agents such as children or animals) [Che03; CT07; Jam+16]. We can say that there is one set of affordances for humans and another one for robots. Then, if a robot can understand both types, it can link human actions and robot actions.

Second, affordances are suited for learning and generalization behaviors on cognitive robots that manipulate objects of the world. Modeling all the possible world interactions is unfeasible (we cannot pre-program all the interactions between a robot, its motor action repertoire, and the resulting effects onto the objects of the world), therefore learning from experience is required. However, to collect large amounts of robot sensorimotor data is challenging and costly. Robot affordances are then one possibility to capture meaningful aspects of data, without necessarily requiring large amounts of data, but permitting to learn a model

[3]The International Organization for Standardization (ISO) defines a "service robot" as a robot "that performs useful tasks for humans or equipment excluding industrial automation applications" (ISO 8373).

that can adapt to situations unseen during training. We describe this aspect in Ch. 2.

Third, affordances can be profitably combined with *communication*, both verbal (i.e., language) and nonverbal (i.e., gestures). For example, the ability to foresee the action performed by other human agents onto physical objects is fundamental for successful *anticipation* and collaboration in joint tasks. If a robot can perceive the affordances offered to a human by objects present in a collaborative human–robot scenario, it can monitor the evolution of the task, anticipate the final goal, and intervene in a timely manner. We explore this aspect in Ch. 4.

Fourth, learning how humans operate *tools* is crucial for having a robot operate in complex manipulation tasks that are typical in human-like environments. We note that *our hands are our first tools* in interacting with physical objects of the world; then, from 16 months of age, humans start developing functional tool use [FRO14]. We investigate this transition on a humanoid robot, modeling the transfer from hand affordances (i.e., perception of action possibilities offered by objects using different hand morphologies) to tool affordances (i.e., perception of action possibilities offered by objects using different tools). A robot can learn tool use capabilities in a gradual way, generalizing to different tools that afford different possibilities, similarly to how children progressively learn mutual interactions between different objects. Acquiring this capability permits a robot to perform actions that would otherwise not be possible, for example grasping a faraway object with the help of an elongated tool. We explore these aspects in Ch. 5.

Fifth, robot sensorimotor knowledge (in the form of learned affordances) can be useful for symbolic reasoning in order to form a unified *planning architecture* that allows a robot to carry out a complex manipulation task under challenging conditions and external disturbances (e.g., noisy perception, motor problems, obstruction by other agents). We present a case study about the POETICON++ project, which tackled all of those issues, in Ch. 6, introducing a robust action planning system that combines robot sensorimotor knowledge (in the form of learned affordances) with symbolic reasoning, using a unified probabilistic representation.

1.4 NEUROSCIENCE INSPIRATION

In this work we draw inspiration from neuroscience (the science that deals with the function and structure of the nervous system and brain), in particular from the following concepts: two-streams hypothesis; canonical neurons and mirror neurons. We will now briefly explain these ideas.

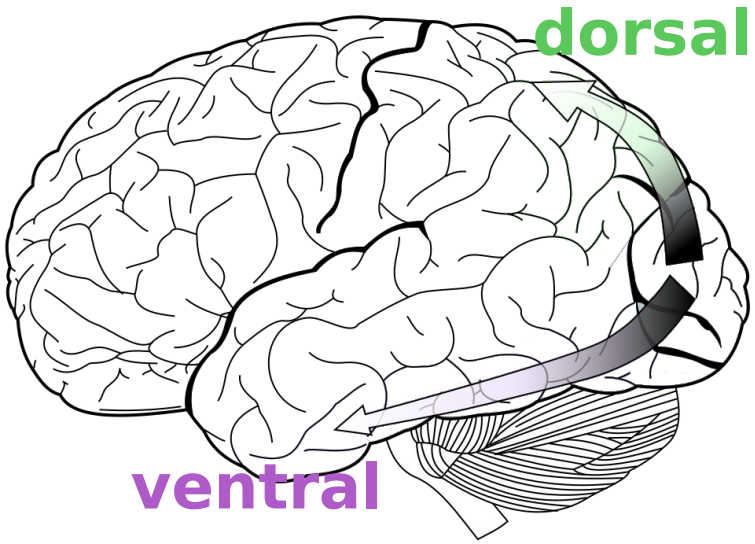


Figure 4: Graphical illustration of the visual processing streams in the human brain according to the two-streams hypothesis. The ventral stream (lower part of figure) is shown in purple, stretching from the visual cortex into the temporal lobe. The dorsal stream (upper part) is shown in green, stretching from the visual cortex into the parietal lobe. Picture elaborated under the CC BY-SA 3.0 license from an image by Wikimedia user Selket, author of the original picture at `https://commons.wikimedia.org/wiki/File:Ventral-dorsal_streams.svg`.

Table 1: Main differences between the ventral stream and the dorsal stream, adapted and simplified from [Nor02].

factor	ventral stream	dorsal stream
function	object recognition	visually-guided behavior (e.g., reaching and grasping)
sensitivity	details (high spatial frequency)	motion (high temporal frequency)
memory	long-term storage	short-term storage
speed	slow	fast
consciousness	high	low
reference frame	object-centered (allocentric)	viewer-centered (egocentric)
visual input	foveal	across all retina

1.4.1 *Two-Streams Hypothesis*

The *two-streams hypothesis* [GM92; CM00] speculates that the primate cerebral cortex processes visual information using two separate pathways:

1. the ventral or "what" stream, responsible for object categorization and recognition;
2. the dorsal or "where" or "how" stream, which guides object-directed actions such as reaching and grasping.

Fig. 4 shows a graphical representation of the two streams in the human brain, whereas Table 1 lists the main differences between the two pathways for the purpose of this thesis. In particular, looking at the *speed* characteristic (i.e., speed of firing after receiving a visual stimulus), it has been observed that the ventral stream is relatively slow [Nor02, p. 84], whereas the dorsal stream is faster [PAD11]. The dorsal pathway constitutes a direct, fast neural link (~250 ms) between vision and motor activation areas in the human brain, bypassing other regions while performing tasks like object reasoning or recognition [Tun+07]. For example, when humans observe a small cup of coffee, they perceive the shape, the handle, and the action possibilities provided by that object (i.e., its affordances) through the fast dorsal stream, whereas the precise object classification provided by the slower ventral pathway does not have the same usefulness, depending on the task.

The reason why this neuroscience theory can be fruitful for cognitive robotics is exemplified by looking at Fig. 5. Each of the two streams has its own perceptual process (with extracted features and outputs), depending on the intended application being considered. In the case of the ventral stream, the application is typically object recognition:

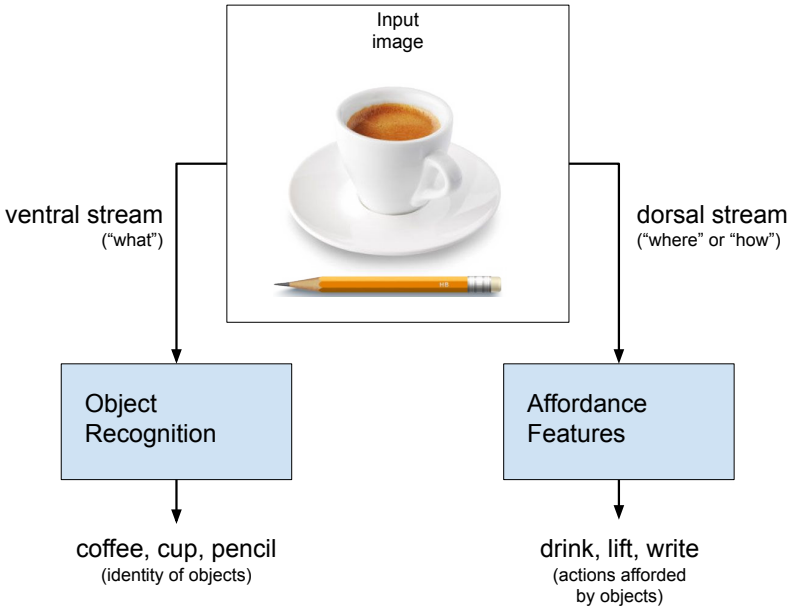


Figure 5: Illustration of the two-streams hypothesis from a computer vision point of view. The ventral and dorsal pathways are highlighted, having different features and scopes. The work developed in this thesis mainly focuses on the dorsal pathway and on learning visual affordances, which does not require to know the identity of objects as in the ventral pathway. However, in Ch. 6, we also show a case study that employs both types of object reasoning in parallel (ventral and dorsal).

that is, identifying the label or name of objects present in the input image, provided that object classes have been previously learned by the system during a learning phase.

On the contrary, in the case of the dorsal stream, we seek visual object features that can be linked to functional properties of the object (i.e., its affordances), without having to recognize the label of the object, that is, without having necessarily seen that object before. Therefore, in the works reported in this thesis we rely on the dorsal stream type of reasoning, because it is a suitable model for *reasoning on objects in a category-independent way* (i.e., not focusing on the object name or category, but rather on its sub-parts and functional features). To that end, we investigate the generalization capabilities provided by this type of reasoning (e.g., when testing objects different from the ones seen during the training phase). This dorsal pipeline is a complement to the ventral one, responsible for object recognition and categorization, which is a topic of its own in computer vision, outside the scope of this thesis. However, in Ch. 6, we will show how it is possible and profitable to employ both types of object reasoning (ventral and dorsal) in a scenario that involves human–robot interaction through spoken verbal requests by the human, action planning and manipulation by the robot.

All in all, we show how learned affordance skills can be leveraged by robots in several contexts and modalities: reasoning on tools, on hand postures, on human gestures, language, and for action planning during manipulation tasks.

1.4.2 *Canonical Neurons and Mirror Neurons*

Certain classes of neurons have been observed to be active in primates (apes and humans) not only during the execution of behaviors, but also during the perception of the objects that are related to these behaviors [RC04].

Canonical neurons respond both during grasping action execution, and when we simply view an *object of a particular shape* [Kan+00, Ch. 19]. In other words, these neurons have sensory properties such that they respond to the presentation of objects on which we can perform an action (e.g., grasping). The grasping of objects needs to be informed by the shape of the object: for instance, we grasp paperclips differently than how we grasp oranges. The sensory input is used to drive appropriate grasping gestures. Canonical neurons are not assumed to be responsible for visual recognition: they just receive relevant input from areas involved in the processing of visual features. We use this idea to link geometrical features, computed from the shape of visually segmented objects, to robot affordances (Ch. 2).

Mirror neurons respond to the motor actions of others. More precisely, according to the *mirror neuron system* theory, there is a neu-

rophysiological mechanism in the brain resulting in certain neurons firing in two different situations: (i) when performing a movement, or (ii) when sensing that a peer is performing a goal-directed movement [Kil+09]. The interpretation is that sensing and execution of movement are two sides of the same coin, and this mirroring mechanism is a fundamental part of action understanding and imitation learning skills [Fog+05; Gaz+07]. We use this idea to recognize manipulative gestures performed by external agents (Ch. 4) and reasoning about the functional properties of the objects involved, thus exploring action prediction and action anticipation capabilities (as in the pointing gesture example of p. 3) integrating affordances with gestures.

People have an *a priori* knowledge about others' body parts, and they use movement cues to understand what actions or gestures are being executed by others. This mechanism can be replicated on a robot, encoding a gesture as the union of (i) body part (appearance cue) and (ii) movement. We can interpret others' actions because what others do belongs to our motor repertoire or experience [RFG01].

To summarize, the contributions described in this thesis are based on a model of the environment surrounding the robot, with the novelty of representing the knowledge of this environment with affordances, beyond the mere recognition of specific objects. Thus, our model looks at the physical objects present in the scene, their affordances, as well as people with their informative body gestures and actions.

1.5 CONTRIBUTIONS

The main contributions of this thesis are:

- a framework for visual robot affordance learning, based on autonomous robot exploration of the world, sensorimotor data collection and statistical models (Bayesian Networks). This system was implemented over the years 2010–2018 as a modular software to be deployed on humanoid robots, focusing on flexibility (e.g., separating perception, learning and motor components, and allowing the user to employ some or all of them as needed) and real-time operations (e.g., supporting robot cameras which capture images at 30 frames per second). As a result, it was adopted as a building block for the further contributions of this thesis (listed in the bullet points below), but it also had an impact externally, being adopted by other researchers: [MC16; Deh+16a; Deh+16b; Deh+17];
- the combination of object affordances with communication (gestures and language). This approach allows a robot to interpret and describe the actions of human agents by reusing the robot's previous experience. This part produced the following publications:

  - Giovanni Saponaro, Lorenzo Jamone, Alexandre Bernardino, and Giampiero Salvi. "Interactive Robot Learning of Gestures, Language and Affordances". In: *Workshop on Grounding Language Understanding*. Satellite of Interspeech. 2017, pp. 83–87. DOI: 10.21437/GLU.2017-17.

  - Giovanni Saponaro, Lorenzo Jamone, Alexandre Bernardino, and Giampiero Salvi. "Beyond the Self: Using Grounded Affordances to Interpret and Describe Others' Actions". In: *IEEE Transactions on Cognitive and Developmental Systems* (2019). DOI: 10.1109/TCDS.2018.2882140.

- a model for tool use affordances in robots. The main contributions were (i) the visual feature extraction component, capable of processing multiple objects simultaneously, extracting information both from the shape of whole objects as well as from their sub-parts; (ii) in the learning component, designing and evaluating various types of computational affordance models and parameters for assorted tasks (e.g., generalization to unseen objects; transfer of learned knowledge from a simulated robot to a real one); (iii) a method for learning the affordances of different robot hand postures, investigating the link from hand affordances (i.e., action possibilities by using the hands) to tool affordances (action possibilities by using tools). This part produced the following publications:

  - Afonso Gonçalves, Giovanni Saponaro, Lorenzo Jamone, and Alexandre Bernardino. "Learning Visual Affordances of Objects and Tools through Autonomous Robot Exploration". In: *IEEE International Conference on Autonomous Robot Systems and Competitions*. 2014, pp. 128–133. DOI: 10.1109/ICARSC.2014.6849774.

  - Afonso Gonçalves, João Abrantes, Giovanni Saponaro, Lorenzo Jamone, and Alexandre Bernardino. "Learning Intermediate Object Affordances: Towards the Development of a Tool Concept". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2014, pp. 482–488. DOI: 10.1109/DEVLRN.2014.6983027.

  - Giovanni Saponaro, Pedro Vicente, Atabak Dehban, Lorenzo Jamone, Alexandre Bernardino, and José Santos-Victor. "Learning at the Ends: From Hand to Tool Affordances in Humanoid Robots". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017, pp. 331–337. DOI: 10.1109/DEVLRN.2017.8329826.

- a case study about the application of affordances and human verbal instructions for robot *planning of manipulation tasks*, de-

veloped within the scope of the POETICON++ research project. This part produced the following publications:

  - Alexandre Antunes, Lorenzo Jamone, Giovanni Saponaro, Alexandre Bernardino, and Rodrigo Ventura. "From Human Instructions to Robot Actions: Formulation of Goals, Affordances and Probabilistic Planning". In: *IEEE International Conference on Robotics and Automation*. 2016, pp. 5449–5454. DOI: 10.1109/ICRA.2016.7487757.
  - Alexandre Antunes, Giovanni Saponaro, Anthony Morse, Lorenzo Jamone, José Santos-Victor, and Angelo Cangelosi. "Learn, Plan, Remember: A Developmental Robot Architecture for Task Solving". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017, pp. 283–289. DOI: 10.1109/DEVLRN.2017.8329819.
  - Giovanni Saponaro, Alexandre Antunes, Rodrigo Ventura, Lorenzo Jamone, and Alexandre Bernardino. "Combining Affordance Perception and Probabilistic Planning for Robust Problem Solving in a Cognitive Robot". In: *Autonomous Robots* (2018). Under review.

- an appendix that describes a novel human gesture recognition model for manipulative hand gestures, inspired by statistical techniques from Automatic Speech Recognition (ASR). It produced the following publication:

  - Giovanni Saponaro, Giampiero Salvi, and Alexandre Bernardino. "Robot Anticipation of Human Intentions through Continuous Gesture Recognition". In: *International Conference on Collaboration Technologies and Systems*. International Workshop on Collaborative Robots and Human–Robot Interaction. 2013, pp. 218–225. DOI: 10.1109/CTS.2013.6567232.

- finally, an appendix related to robot communication (rather than robot perception and action, the core topics of the thesis): the perceived social attitude attributed by non-technical users when observing certain head and body gestures performed by a robot. This study paves the way for an active learning system capable of optimizing robot motion parameters with the aim of improving the communicative expressiveness. It produced the following publication:

  - Giovanni Saponaro and Alexandre Bernardino. "Generation of Meaningful Robot Expressions with Active Learning". In: *ACM/IEEE International Conference on Human–Robot Interaction*. Late Breaking Report. 2011, pp. 243–244. DOI: 10.1145/1957656.1957752.

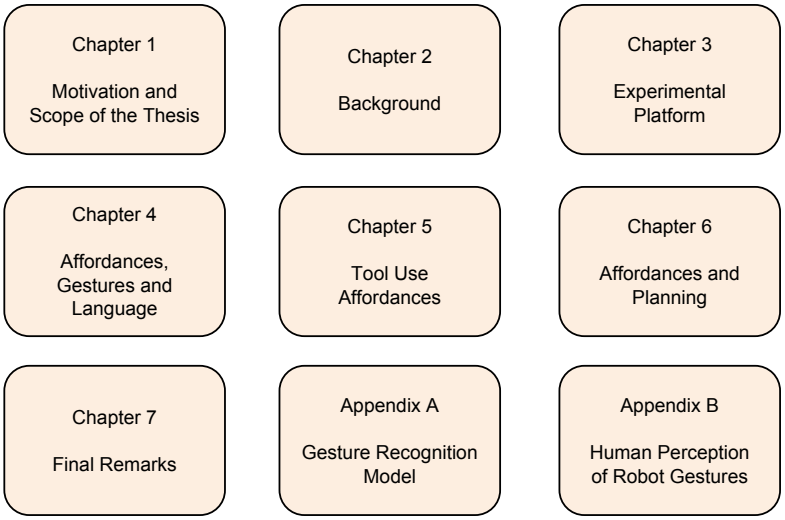


Figure 6: Structure of the thesis.

1.6 OUTLINE OF THE THESIS

The thesis is structured as sketched in Fig. 6.

Ch. 2 illustrates the main machine learning notation and concepts (including Bayesian Networks) used in the thesis, and the previous works in the literature which are related to our broad scope (other chapters also contain specific related work sections).

Ch. 3 describes the experimental platform: the iCub humanoid robot, the experimental scenario under study, and our modular software framework for robot affordance learning, focused on autonomous robot exploration of the world and visual processing, used as a building block in the rest of the thesis.

Ch. 4 presents a computational model that combines affordances with language, broadly speaking. It does that by incorporating nonverbal language, in the form of *human gestures*; and also verbal language, by generating verbal descriptions of manipulative scenes.

Ch. 5 shows an affordance model that deals with multiple objects, giving rise to *tool use*, including the link from hand affordances (i.e., action possibilities by using the hands) to tool affordances (action possibilities by using tools).

Ch. 6 illustrates a case study about the application of affordances and human verbal instruction interpretation for robot *planning of manipulation tasks*, developed within the scope of the POETICON++ research project.

Ch. 7 draws the conclusions and lists the avenues for future work.

Appendix A presents the details of the human gesture recognizer for manipulative hand gestures: this recognizer was inspired by statistical techniques from Automatic Speech Recognition (ASR), and it was employed as one of the components of Ch. 4,

Finally, as far as the robot communication aspect is concerned (besides robot perception and action, the core topics of the thesis), Appendix B deals with the perceived social attitude attributed by non-technical users when observing certain body gestures performed by a robot.

2

BACKGROUND

In this chapter, we provide the theoretical groundwork for the thesis. In Sec. 2.1 we define some necessary machine learning concepts and models, how to train these models and how to make use of them. Then, in Sec. 2.2 we discuss the main works in the literature which are related to the broad scope of the thesis (note that, in the other chapters, we will also provide chapter-specific summaries of the related works about a particular topic, for example tool use in robotics).

2.1 THEORETICAL CONCEPTS

In this section, we introduce the probabilistic models and machinery used for developing affordance learning in the rest of the thesis. We adopt the notation from [Bis07].

2.1.1 *Probability Theory*

A random variable X is a variable whose possible values are numerical outcomes of a random phenomenon. In general, these outcomes can be discrete or continuous, but *we focus on random variables with discrete values*. We write $p(X = x_i)$ (supposing discrete values indexed by $i = 1, \ldots, M$) to denote the probability that X takes the value x_i. Given two random variables X and Y, the notation $p(X = x_i, Y = y_j)$ indicates the *joint probability* of $X = x_i$ and $Y = y_j$ ($j = 1, \ldots, L$), expressing the probability that each of X and Y falls in any particular value specified for that variable. In the case of two random variables, this joint probability distribution is also called a bivariate distribution. The concept can be generalized to any number of random variables: in that case, it is called a multivariate distribution.

The joint probability distribution can be used to determine two other types of distributions:

- the *marginal probability* distribution, which gives the probabilities for any one of the variables with no reference to any specific ranges of values for the other variables; and
- the *conditional probability* distribution, which expresses the probabilities for any subset of the variables, *conditioned on* particular values of the remaining variables.

So far, we have used the notation $p(X = x_i)$ to distinguish the random variable X from its possible value x_i. Now, we introduce a notation that is more compact and readable: $p(X)$ denotes a *distribution* over

the random variable X. With this, we can write the two *fundamental rules of probability theory*, which are (i) the sum rule

$$p(X) = \sum_{Y} p(X, Y) \tag{1}$$

and (ii) the product rule

$$p(X, Y) = p(Y \mid X)p(X), \tag{2}$$

where $p(X, Y)$ is the joint probability of X and Y, $p(Y \mid X)$ is the conditional probability of Y given X, and $p(X)$ is the marginal probability of X.

From (2), using the symmetry property $p(X, Y) = p(Y, X)$, we obtain the *Bayes' rule* (or Bayes' theorem), which is a relationship between conditional probabilities:

$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}, \tag{3}$$

where $p(Y \mid X)$ is called the *posterior* probability of the hypothesis Y given the evidence X, $p(X \mid Y)$ is the *likelihood* of the evidence X if the hypothesis Y is true, $p(Y)$ is the *prior* probability of the hypothesis Y, and $p(X)$ is the probability that the evidence X itself is true.

Bayes' rule is the basis of *Bayesian inference* or *reasoning*: a method of statistical inference in which we use the rule to update the probability of a hypothesis, as more information becomes available. The two key elements of (3) are the prior $p(Y)$ and the likelihood $p(X \mid Y)$. The prior can be interpreted as the probability that we assign to a hypothesis before we gather any new information. The likelihood can be interpreted as the probability of some particular piece of data being collected if the hypothesis is correct.

2.1.2 *Graphical Models*

Even though probabilistic events of the world can be modeled purely with algebra by using the two fundamental probability rules of (1) and (2), in many applications it is useful to capture richer events by resorting to *graphical models* [Bis07, Ch. 8]. Their advantages are:

- they provide a simple way to visualize the structure of a probabilistic model and can be used to design and motivate new models;
- information about the properties of the model, including conditional independence properties, can be obtained by inspecting the graph;
- complex computations, required to perform inference and learning, can be conveniently expressed in terms of graphical manipulations (which maintain the underlying mathematical expressions and properties implicitly).

In addition, in the next chapters we will see how a particular type of graphical models, *Bayesian Networks* [Pea88; Jen96] (also called directed graphical models or belief networks), exhibits further advantages for modeling the specific problem of robot affordance learning. Bayesian Networks (BNs):

- allow us to take into account the *uncertainty* of the world [RN09, Ch. 14, Probabilistic Reasoning];
- are suited to capture the notion of *causality* [Bis07, p. 366];
- provide a unified framework for learning and using affordances [Mon+08];
- have been introduced back in the mid-1980s [Pea88], so they have been widely studied. In practical terms for researchers, that means that a number of mature, documented software packages and examples implementing BNs is readily available: for instance in the form of MATLAB toolboxes[1], Python packages[2] or R packages[3]. This makes the usage of BNs convenient for prototyping.

A graph comprises *nodes* (or vertices) connected by *edges* (or arcs, or links). In a probabilistic graphical model, each node represents a random variable (or group of random variables), whereas the edges represent probabilistic relationships between these variables. The graph captures the way in which the *joint distribution* over all of the random variables can be decomposed into a product of factors, each depending only on a subset of the variables.

In the case of BNs, the edges of the graphs have a directionality, indicated by arrows. BNs offer the *possibility of expressing causal relationships* between random variables. We will clarify this aspect momentarily.

In formal terms, a BN is a graphical model representing dependencies between random variables as a Directed Acyclic Graph (DAG). The network is defined by a pair $B = (G, \theta)$, where G is the DAG structure whose nodes represent random variables, and θ is the set of parameters of the network. Each node represents a random variable $Y_i, i = 1, \ldots, n$, whereas the edges (or lack of them) between two nodes Y_i and Y_j represents *conditional independence* of the corresponding variables.

The Conditional Probability Distribution (CPD) of each variable Y_i in the network, denoted as $p(Y_i \mid Y_{\text{parents}(Y_i)}, \theta_i)$, depends on (i) the par-

[1] Bayes Net Toolbox (`https://github.com/bayesnet/bnt`), Probabilistic Modeling Toolkit (`https://github.com/probml/pmtk3`).

[2] Pomegranate (`https://github.com/jmschrei/pomegranate`), Python Library for Probabilistic Graphical Models (`https://github.com/pgmpy/pgmpy/`).

[3] bnlearn (`http://www.bnlearn.com/`).

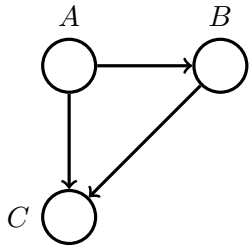


Figure 7: A directed graphical model representing the joint Probability Density Function (PDF) over three variables A, B and C according to the decomposition of the right-hand side of (6). Adapted from [Bis07].

ents node of Y_i, denoted as parents(Y_i), and (ii) a set of parameters θ_i. The joint distribution of the BN decomposes as:

$$p(Y_1, \ldots, Y_n \mid \theta) = \prod_{i=1}^{n} p\left(Y_i \;\middle|\; Y_{\text{parents}(Y_i)}, \theta_i\right), \tag{4}$$

where θ represents all the parameters in the different CPDs.

Above, we mentioned that BNs offer the possibility of representing causal relationships: for example, an edge $Y_i \to Y_j$ can represent the information that "Y_i causes Y_j" [Pea88]. We shall now clarify that possibility. If we (experimenters) know that there is a causal relation in a phenomenon of the world, we can represent such information in a BN by attributing a certain arrow direction to an edge. However, this is only an indication to us. It reminds us that we possess extra information, in addition to the one codified by the BN model. A BN, *per se*, does not describe causal relationships, nor can it learn them. In other words, causal relationships and directions of arrows are decided arbitrarily by the experimenters, depending on the phenomenon being modeled (in the case of this thesis, this is expressed in Sec. 3.2). This arbitrariness is related to the factorization being chosen, of which we give some examples below.

To illustrate BNs, let us consider a joint distribution $p(A, B, C)$ over three discrete variables. By applying the product rule of probability (2), we can write the joint distribution in the form

$$p(A, B, C) = p(C \mid A, B)p(A, B). \tag{5}$$

By applying the product rule again to the right-hand side of (5), we obtain

$$p(A, B, C) = p(C \mid A, B)p(B \mid A)p(A). \tag{6}$$

The above decomposition holds for any choice of the joint distribution. We now represent the right-hand side of (6) in terms of a *graphical model* as follows. First, we introduce a node for each of the random variables A, B and C, and we associate each node with the corresponding *CPD* on the right-hand side of (6). Second, for each CPD we add

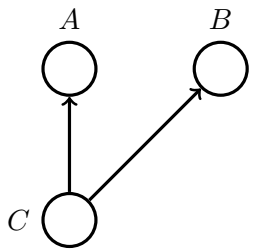


Figure 8: A directed graphical model representing the joint Probability Density Function (PDF) over three variables A, B and C according to the decomposition of the right-hand side of (7). Adapted from [Bis07].

directed edges (arrows) to the graph from the nodes corresponding to the variables on which the distribution is conditioned. Therefore, for the factorization $p(C \mid A, B)$, there will be edges from nodes A and B to node C, whereas for the factorization $p(A)$ there will be no incoming edges.

Note that the order for the factorization was arbitrary: other factorizations represent the same evidence identically. However, if we know *a priori* the causal relationships of the domain, we can choose the parents so that they better reflect our beliefs about the causality in the domain.

The resulting graph is shown in Fig. 7. If there is an edge going from a node A to a node B, we say that node A is a *parent* of node B, and conversely we say that B is the *child* of node A. We do not make any formal distinction between a node and the variable to which it corresponds, but we simply use the same symbol to refer to both (interchangeably).

We now give a few examples of conditional independence and its properties.

As a *first example* of graphical models to illustrate the concept of *conditional independence*, let us consider the graph in Fig. 8. In this case, the joint probability $p(A, B, C)$ is

$$p(A, B, C) = p(A \mid C)p(B \mid C)p(C). \tag{7}$$

Let us now apply the rules of probability, in particular we will *marginalize* (i.e., sum out over irrelevant variables), which can be done with regard to any variable. If none of the three variables is observed (see Sec. 2.1.4), then we can marginalize both sides of (7), for instance with respect to C, obtaining

$$p(A, B) = \sum_{C} p(A \mid C)p(B \mid C)p(C).$$

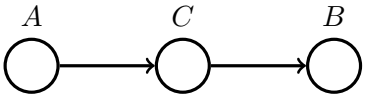


Figure 9: A directed graphical model representing the joint Probability Density Function (PDF) over three variables A, B and C according to the decomposition of the right-hand side of (9). Adapted from [Bis07].

If, instead, we condition (7) on the variable C, we can write the CPD of A and B given C (conditional independence property) as

$$\begin{aligned} p(A, B \mid C) &= \frac{p(A, B, C)}{p(C)} \\ &= p(A \mid C)p(B \mid C). \end{aligned} \tag{8}$$

We can think of a graphical interpretation of (8) by looking at the *path* from node A to B via C in Fig. 8. We say that C has a *tail-to-tail* connection with respect to this path, because the node is connected to the tails of the two arrows, and the presence of the path connecting A and B causes these nodes to be dependent. When we condition on C, the conditioned node "blocks" the path from A to B, as a consequence A and B become (conditionally) independent.

As a *second example*, we can consider the graph of Fig. 9. Its corresponding joint distribution is

$$p(A, B, C) = p(A)p(C \mid A)p(B \mid C). \tag{9}$$

If none of the variables are observed, we can marginalize over C, obtaining

$$\begin{aligned} p(A, B) &= p(A) \sum_{C} p(C \mid A)p(B \mid C) \\ &= p(A)p(B \mid A). \end{aligned}$$

If we condition on C, using Bayes' rule (3) and (9), we obtain the conditional independence

$$\begin{aligned} p(A, B \mid C) &= \frac{p(A, B, C)}{p(C)} \\ &= \frac{p(A)p(C \mid A)p(B \mid C)}{p(C)} \\ &= p(A \mid C)p(B \mid C). \end{aligned}$$

We say that C is *head-to-tail* with respect to the path from A to B. This path connects nodes A and B and makes them dependent. If we now observe C, then this observation "blocks" the path from A to B and we obtain the conditional independence.

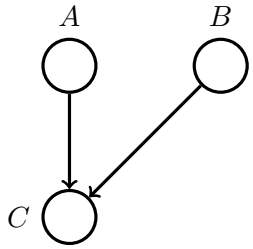


Figure 10: A directed graphical model representing the joint Probability Density Function (PDF) over three variables A, B and C according to the decomposition of the right-hand side of (10). Adapted from [Bis07].

As a *third example*, let us consider the graph of Fig. 10. Its corresponding joint distribution is

$$p(A, B, C) = p(A)p(B)p(C \mid A, B). \tag{10}$$

If none of the variables are observed, we can marginalize both sides of (10) over C, obtaining

$$\begin{aligned} p(A, B) &= p(A)p(B)\cancel{\sum_C p(C \mid A, B)} \\ &= p(A)p(B). \end{aligned} \tag{11}$$

If we condition on C, using Bayes' rule (3) and (10), we obtain the conditional independence

$$\begin{aligned} p(A, B \mid C) &= \frac{p(A, B, C)}{p(C)} \\ &= \frac{p(A)p(B)p(C \mid A, B)}{p(C)}. \end{aligned}$$

We say that C is *head-to-head* with respect to the path from A to B, because it connects to the heads of the two arrows. When C is not observed, it "blocks" that path, and the variables A and B are independent, as expressed by (11), in contrast to the two previous examples. However, conditioning on C "unblocks" the path, rendering the variables A and B dependent.

Summarizing, a tail-to-tail node or a head-to-tail node leaves a path unblocked unless it is observed, in which case it blocks the path. Instead, a head-to-head node blocks a path if it is unobserved, but once the node (or one of its descendants[4]) is observed, the path becomes unblocked.

Having examined the above instances, we shall now introduce the notion of *equivalence classes* of graph structures: two DAGs G and G'

[4]Node Y is a *descendant* of node X if there is a path from X to Y in which all steps of the path follow the directions of the arrows [Bis07, p. 376].

are equivalent if, for every BN $B = (G, \theta)$, there exists another network $B' = (G', \theta')$ such that both define the same probability distribution.

Structure Learning techniques, which we will describe in Sec. 2.1.3, are able to distinguish among equivalence classes of graph structures.

This is linked with the concept of *correlations* in the following sense: equivalence classes contain different correlations between the nodes of the network.

In order to be able to infer the correct correlation, i.e., to disambiguate between graph structures in the same equivalence class, it is necessary to use *interventional variables*, i.e., variables which are fixed to a specific value. We will use interventional variables in robot experiments throughout the thesis, giving an example when we describe the experimental robot setup in Ch. 3. The fact that robots make decisions to intervene in the world is what makes it possible to learn correlations.

2.1.3 *Learning the Structure of Bayesian Networks*

In light of the principles of developmental robotics, which is one of the motivations of this thesis (see Sec. 1.2), it is interesting to mention Structure Learning. Recall that, in developmental robotics, an embodied agent builds its cognition step by step, typically by incremental self-exploration of the surrounding environment, starting from a limited initial knowledge, then progressing towards the discovery of patterns and facts about the world, as time and experience advance. In this sense, Structure Learning can be loosely interpreted as the discovery of correlations in the environment.

Learning the structure of the network, G, is a model selection problem, where the search space contains all possible structures of DAGs, given the number of variables in the domain [Pea88].

This can be formalized as estimating the distribution over all possible network structures $G \in \mathcal{G}$ given the data. Using Bayes' rule (3), we can express this distribution as the product of the marginal likelihood and the prior over graph structures,

$$p(G \mid D) = \eta\, p(D \mid G) p(G), \tag{12}$$

where $\eta = 1/p(D)$ is a normalization constant. The prior $p(G)$ allows to incorporate previous knowledge on possible structures.

Because the number of DAGs is super-exponential in the number of nodes [Rob77][5], it is unfeasible to enumerate all possible network

[5] See for example the Bayes Net Toolbox documentation (`http://bayesnet.github.io/bnt/docs/usage.html`), where the number of DAGs as a function of the number of nodes, $G(n)$, is given by the recurrence equation (super-exponential in n)

$$G(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} G(n-k).$$

structures and assign them a score, even for a low number of nodes. This justifies the usage of heuristics to find a (local) maximum in the structure space, approximating the full distribution. Several methods have been proposed to approximate the distribution $p(G \mid D)$, such as: Markov Chain Monte Carlo (MCMC) [MYA95], K2 [CH92; BLL11], Bayesian Dirichlet likelihood-equivalence (BDe) [SW09].

The MCMC algorithm [MYA95] applied to BN Structure Learning generates a set of samples of possible network structures with relative frequencies that correspond to the Bayesian posterior distribution $p(G \mid D)$. These samples can then be used to estimate the posterior probabilities of particular features of interest, marginalizing over the various structures. Typically, implementations of this algorithm employ METROPOLIS–HASTINGS sampling [GC03].

The K2 algorithm [CH92; BLL11] searches for the structure that maximizes the joint probability of structure and data, $p(G, \theta)$. For this, it assumes a known ordering on the domain variables and that all possible structures are equally likely. It starts from the lowest-order node and makes its way sequentially to the highest. At each node, it first assumes that it has no parents, then it uses a greedy-search method over the K2 score [CH92] of the lower-order nodes to incrementally add them as its parents. With BDe [SW09], the structure of the networks is maximized by using greedy search and simulated annealing. In Ch. 5, we will examine BN Structure Learning algorithms used in robot tool use affordance experiments.

2.1.4 *Learning the Parameters of Bayesian Networks*

The structure of a BN can either be provided by a human expert, or it can be learned with the methods described in the previous section. In any case, given the structure of a BN, the parameters θ_i of each node can be estimated (learned) with a Bayesian approach [HGC95]. Then, the estimated parameters can also be updated online, permitting the incorporation of further information provided by new trials and experiments.

If a BN has a known structure and it is fully observable (i.e., all the variables represented by nodes are observed), the goal of parameter learning is to find the values of the BN parameters (in each CPD) that maximize the (log)-likelihood of the training data. In this case, we can use maximum-likelihood estimation. Given a training dataset $\Sigma = \{x_1, \ldots, x_m\}$, where each $x_l = (x_{l1}, \ldots, x_{ln})^\top, 1 < l < m$, is an n-dimensional data point corresponding to one realization (value) of the random variable X_i, and the parameter set $\theta = (\theta_1, \ldots, \theta_n)$, where θ_i is the vector of parameters for the CPD of variable X_i (represented

by one node in the graph), the log-likelihood of the training dataset is a sum of terms, one for each node:

$$\log L(\theta \mid \Sigma) = \sum_l \sum_i \log p(x_{li} \mid \text{parents}(X_i), \theta_i). \tag{13}$$

On the other hand, if the BN is only partially observable (i.e., some nodes are hidden or data is missing), parameter learning is (in general) computationally intractable. However, we can use the Expectation–Maximization (EM) algorithm to find a locally optimal maximum-likelihood estimate of the parameters. If the conditional distributions and the parameter priors are conjugate, the CPDs and marginal likelihood can be computed in closed form, thus being efficient in terms of learning and inference algorithms.

2.1.5 *Making Inference on Bayesian Networks*

Having a BN, we can compute $p(X_{\text{inf}} \mid X_{\text{obs}})$, where X_{obs} is the set of observed variables, and X_{inf} is the set of variables on which we wish to perform an inference. This computation is also called a query (i.e., we query a network and as a result we obtain a response). Usually, this is done by first converting the BN into a tree-like structure[6], and then applying the Junction Tree algorithm [LS88; HD96; Bis07] in order to compute the queried distribution of interest. The advantage of the Junction Tree algorithm is that of avoiding to work directly with the joint distribution of the variables considered, relying instead on factorization properties.

Importantly, for performing inference it is not necessary to know all the values of all the variables. This entails that a query can combine any combination of the nodes (e.g., any combination that uses object features, actions and effects, as we will see in the affordances applications in Sec. 2.2) either as observed variables or as the desired (inferred) output.

Based on this probabilistic machinery, we can now use an affordance knowledge BN to answer questions such as "which is the best action to achieve an effect?" or "which effect will be obtained by exerting this action on this object?", simply by computing the appropriate distributions. For instance, predicting the effects of an observed action a_i given the observed features o_j can be performed from the distribution $p(E \mid A = a_i, O = o_j)$.

The advantage of using BNs is that their expressive power allows the marginalization over any set of variables given any other set of variables. For instance, referring to the diagram of Fig. 2a which depicts the computational model of affordances by Montesano [Mon+08], one can extract different types of information (i.e., perform different types of queries) from a previously-trained network, such as:

[6] Converting a BN graph into a tree is an operation that involves several computational steps. For a detailed explanation see, for example, [Bis07, p. 416].

EFFECT PREDICTION Given the motor action A executed by the robot and its target object O, compute the distribution of the resulting physical effects: $p(E \mid A, O)$;

PLANNING Given the target object O and the (desired) physical effect E, compute the appropriate motor action:

$$A^* = \arg\max_A p(A \mid O, E);$$

OBJECT PROPERTIES Given the target object O, compute the distribution of its possible physical effects: $p(E \mid O)$;

ACTION PROPERTIES Given the motor action A, compute the distribution of possible resulting physical effects: $p(E \mid A)$.

2.2 PREVIOUS WORKS

In this section, we outline some previous works in the literature which are related to the broad scope of the thesis. Explaining these works is useful to understand the contributions of the next chapters, where we will also provide chapter-specific summaries of the related works about particular topics.

What the works described below have in common is that they tackle the challenges associated with having autonomous robots operate in human-centered, unstructured environments. To do that, they propose (i) to equip robots with the capability of building a *model of the environment* surrounding them from autonomous exploration; (ii) to incorporate (in such a model of the environment) elements such as physical *objects* present in a scene, their affordances, and possibly the information expressed by human agents by performing physical *actions* or body gestures; (ii) to use such a model for making sense or finding meaning (in other words, to do *reasoning*) about the environment.

This possibility of robots reasoning about their environment has several applications: prediction of the future, imitation of another agent, planning of complex actions that require multiple sub-actions, provision of feedback to humans when doing shared human–robot collaboration tasks.

We now proceed in citing the previous works, categorized according to their main focus or topic.

2.2.1 *Reasoning about Objects*

In the previous chapter we have mentioned some advantages obtained by incorporating object affordances in cognitive robotic systems. The concept of affordances is applicable to autonomous robots and has influenced many robotic studies and models, specifically because (i) affordances depend on perceptual and motor capabilities of the agent (e.g.,

whether the robot is mobile or not, how tall it is, whether it has arms, actuators, etc.); (ii) affordances suggest action possibilities to the agent from direct perception, thus providing a means to predict the consequences of physical actions (e.g., accomplish a given goal in a novel situation, as in the coffee example of p. 2).

We can summarize the advantages as follows:

1. affordances can be *learned* by a robot that explores different actions exerted on the environment (e.g., on the objects present in the environment) autonomously, or semi-autonomously;

2. after learning, the acquired knowledge can be used for *reasoning* (e.g., to perform inference about object and action properties);

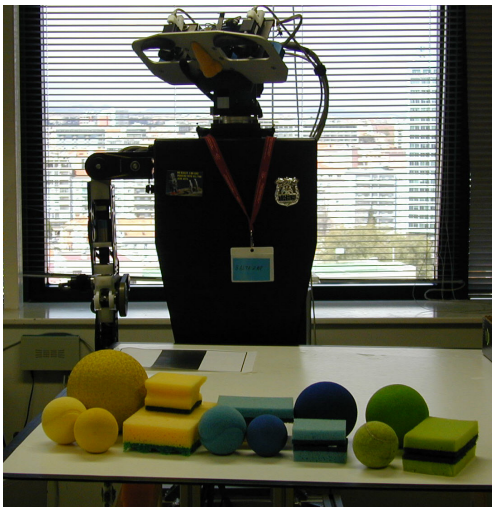
3. affordances are *robust* in the sense that they can use incomplete data or limited data.

The above aspects are relevant because modeling all the possible world interactions of a robot is unfeasible (see Sec. 1.3), thus learning from experience is required. In turn, this poses the challenge of collecting a large amount of experiments or training data, which can be partially mitigated by learning affordance models that perform adequately with only limited data.

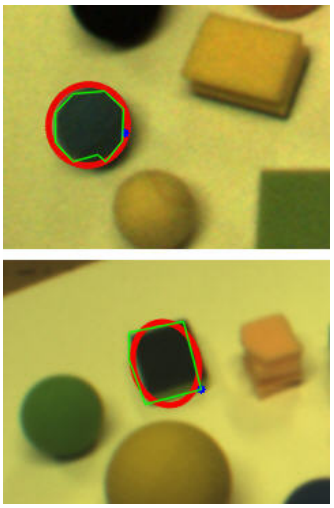
The idea of using object affordances for supporting robot capabilities such as scene understanding, reasoning and learning, has been proposed by several authors since the mid-2000s [Fit+03; LS05; MT08] with different aims and motivations. In perceiving human activities, object affordances can be used to infer the action executed by a user just by observing the resulting effect [KNY02; Mon+08]. This knowledge can then be used to make a robot provide feedback, or to imitate the human action with skill transfer [KNY02; LS05; LMM07; LS07], or to actively aid the human towards realizing a shared collaborative plan [Lal+13; JKS13]. Thill [Thi+13] published a review of computational models of affordances inspired by the mirror neuron system. More recently, a survey about the role of affordances in psychology, neuroscience, and robotics was published [Jam+16], followed by a comprehensive taxonomy of approximately 150 computational models of affordances in the robotics literature [Zec+17].

In this section, we focus on the work by Montesano [Mon+08; Mon+10], which is the starting point behind the contributions presented in this thesis.

That work is influential in the cognitive robotics community, because it shows a computational model of affordances that is able to account for multiple possible affordances present in a robot's environment in a principled and probabilistic way, as opposed to assuming the existence of only one pre-defined affordance (e.g., liftability). In other words, this model is capable of learning multiple affordances present in the environment, or multiple possibilities offered by the objects perceived



(a) Humanoid robot in its workspace with a table and some objects.



(b) Objects being perceived and visually segmented by vision algorithms.

Figure 11: Experimental setup of [Mon+08]. In this work, a robot learns object affordances by autonomous exploration of colorful toys on a table; affordances are modeled as relationships between actions, objects and effects.

by the agent. Computationally, it achieves this by using a Bayesian Network (BN) with a structure that encodes relations between motor actions, object features and resulting effects, as depicted in Fig. 2a.

In a self-exploration manner (see Sec. 1.2), the Baltazar humanoid robot [Lop+04] tries out different motor actions onto different physical objects and records the observed effects, as shown in Fig. 11. Then, it learns the relations between the random variables involved (i.e., the variables pertaining to actions, object features and effects). The actions are pre-defined tapping motions performed with the end effector, from four different directions. The object features are (discretized) quantities related to size, shape, and color of objects extracted from vision. The effects are the (discretized) physical displacements of the objects being moved on a tabletop, and the (discretized) durations of the contacts acquired with tactile sensors. Data is discretized by using k-means clustering [Llo82] in order to train the BN efficiently.

By repeating the exploration procedure several times, the robot acquires a set of N samples

$$D = \{y^{1:N}\}, \tag{14}$$

where the lower-case letter y represents the possible realizations (i.e., values) of the random variable indicated by the upper-case letter Y (see Sec. 2.1.1). Then, the set of nodes in a network, Y, includes all

of its variables, i.e., the ones representing robot actions (A), object features (O) and resulting effects (E), as follows:

$$Y = \{A, O_1, \ldots, O_{n_O}, E_1, \ldots, E_{n_E}\}. \tag{15}$$

For the sake of this summary, let us assume for simplicity that the structure of the BN is known, i.e., that we know the dependencies between the variables in Y.

Given the (discrete) representation of actions, object features and effects, the authors use a *multinomial distribution* and its corresponding conjugate, the Dirichlet distribution, to model the Conditional Probability Distributions (CPDs) $p\left(Y_i \;\middle|\; Y_{\text{parents}(Y_i)}, \theta_i\right)$ and the corresponding parameter priors $p(\theta_i)$, respectively. Let $\mathcal{Y}_i$ and $\mathcal{Y}_{\text{parents}(Y_i)}$ indicate the range of values of random variables Y_i and the range of values of the parent nodes of Y_i, respectively. Assuming independence between the samples in D, the marginal likelihood for the random variable Y_i and its parents given D ([HGC95]) is:

$$\begin{aligned} p\left(y_i^{1:N} \;\middle|\; y_{\text{parents}(Y_i)}^{1:N}\right) &= \int \prod_{n=1}^{n} p\left(y_i^N \;\middle|\; y_{\text{parents}(Y_i)}^n, \theta_i\right) p(\theta_i)\, d\theta_i \\ &= \prod_{j=1}^{|\mathcal{Y}_i|} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{|\mathcal{Y}_{\text{parents}(Y_i)}|} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \end{aligned}$$

where N_{ijk} counts the number of samples in which $Y_i = j$ and $Y_{\text{parents}(Y_i)} = k$, $N_{ij} = \sum_k N_{ijk}$ and Γ represents the gamma function. The *pseudo-counts* α_{ijk} denote the Dirichlet hyper-parameters of the prior distribution of θ_i and $\alpha_{ij} = \sum_k \alpha_{ijk}$. The marginal likelihood of the data is simply the product of the marginal likelihood of each node,

$$p(D \mid G) = p(Y^{1:N} \mid G) = \prod_i p\left(y_i^{1:N} \;\middle|\; y_{\text{parents}(Y_i)}^{1:N}\right), \tag{16}$$

where we have made explicit the dependency on the graph structure, G.

One of the characteristics of Montesano's computational model of affordances is that it relies on discrete quantities being computed (by a clustering algorithm) and passed as input to the BN, rather than on the raw continuous-valued variables themselves. In an extention work, Osório relaxes this assumption [Osó+10] in order to use the continuous values directly, by employing Gaussian Mixture Models (GMMs) to represent the perceived visual features. Results from a simulated environment suggest that continuous values can help BNs when data is noisy and when plenty of training data is available. However, the practical applicability of this approach on real robots is problematic because of its computational cost: the proposed solution uses the Expectation–Maximization (EM) algorithm, whose execution time is much slower than the one of Montesano's discrete BN nodes.

Also, Montesano's model examines the possibilities afforded by *one object* at a time to the agent (e.g., a ball affords high rollability). In

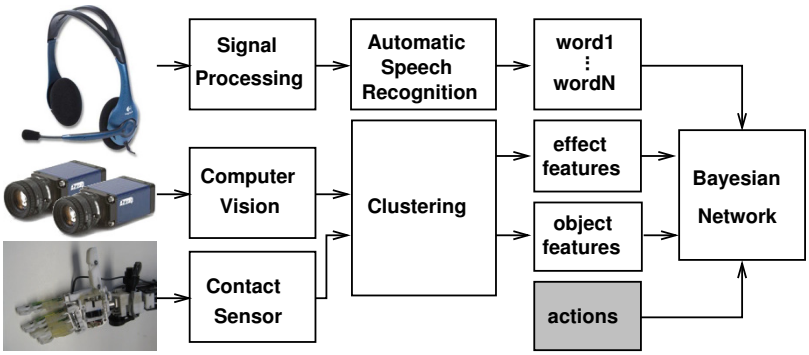


Figure 12: Experimental setup of [Sal+12]. In this work, which extends [Mon+08], a robot learns associations between *spoken words* and object affordances (where affordances are modeled as the relationships between actions, object features, effect features).

Ch. 5 we will extend the model in the context of *tool use* affordances, meaning that, with exploration, the robot will learn to reason about two objects at a time, to be used as a tool and as an affected object, respectively (e.g., a hammer and a nail).

2.2.2 *Affordances and Language*

We have mentioned that robot affordances can help understanding a user's intention by recognizing the action performed by them. Additionally, some works have used the concept of *sensor fusion* to build *multimodal affordances*, for example incorporating human language in addition to objects, where different modalities complement each other.

One of the first papers in this category is the one by Moratz [MT08], where linguistic instructions from a human to a robot are grounded in the affordances of objects present in a scene. This system employs a robot object recognition system composed with a laser range finder, making use of an affordance-informed visual algorithm (relating object shapes to object functionalities in a way pre-defined by the experimenter). As a result, the system can be used to instruct a robot verbally, so that words relating to the affordances are mapped to the objects allowing the robot to choose the objects to use. Notably, this setup requires a pre-defined set of known objects and of the rules associating affordances to objects. Also, this approach does not exploit the information contained in nonverbal language, such as human gestures and movements.

A system with less stringent assumptions is the one by Salvi [Sal+12], whose experimental setup is shown in Fig. 12. In this work, building upon Montesano's model (see Sec. 2.2.1), object affordances are used to associate human words to the actual action, object and effect that they refer to. Computationally, BNs are used, serving to learn

Figure 13: Sequence of frames of a cutting action, reproduced from [PA12].

words–meanings associations. This knowledge is then used together with spoken instructions for removing ambiguities during interactions with humans, for example permitting to command the robot to perform tasks. We will expand upon the system by Salvi in Ch. 4.

Another work relating affordances and language is [COK15], which models the co-occurrence of actions, object information and language with a *concept web* based on Markov Random Fields (MRFs). During operation, if partial information is available (e.g., only the visual object information or the corresponding words), the corresponding affordance concepts previously learned are also activated.

2.2.3 *Reasoning about Human Actions*

Up to now, we have mentioned a number of works about robots that explore their environment, they operate on it (e.g., using their limbs), and they build a cognitive model of the environment that takes into account the physical objects and the afforded actions. Even though some of the works considered language, which is a human trait, the human dimension was not prominent in a physical or visual sense, meaning that the cognitive model used by the robot did not have any explicit representation of human users (e.g., their location, their state, their physical action). However, a growing line of research *does* tackle this aspect, incorporating advances from other disciplines (e.g., human activity recognition, machine learning and computer vision [AR11]) onto robots.

Thus, we now list some works about autonomous robots possessing cognitive reasoning algorithms about environment objects *and the humans* surrounding them. This list is not exhaustive, but it is useful to understand the contributions of the next chapters.

Pastra and Aloimonos [PA12] propose a "minimalist grammar of action" for robot cognition, linking the two aspects of language and action together. That work is motivated by the biological evidence that both language and action are organized in a hierarchical, compositional way, and that the neural locus for composing their mechanisms is shared in Broca's area [Pul05]. For example, Fig. 13 shows a human person cutting an eggplant. In order to do that, the person uses some prior knowledge and performs a sequence of low-level motor actions (e.g., reaching for a knife tool, positioning the knife over the vegetable, exerting a

Figure 14: Tennis-playing robot, from [Wan+13].

vertical force to cut the vegetable, etc.), resulting in a high-level action (e.g., cutting the vegetable). In short, the proposed "grammar" is a formal, tree-like specification of actions with a biological human base. This specification allows the development of generative computational models for action in the motor and visual space, by deploying a software component of a semantic memory (i.e., the general human knowledge accumulated with experience) called the PRAXICON [Pas08; MP16]. Indeed, this integration has been performed on a humanoid robot during the POETICON++ project[7], and it will be described in detail in Ch. 6.

In [Wan+13], a group from the Technical University of Darmstadt shows an example of a robotic system capable of recognizing and *anticipating* a human's movements. This system, shown in Fig. 14, is capable of playing table tennis against a human opponent, using vision, control and machine learning. It uses Gaussian Processes [Bis07, p. 303], finding a latent state representation of noisy and high-dimensional observations of human movement, at the same time capturing the dynamics of the motor task being considered. Online approximate inference permits to anticipate the target position of the tennis ball (i.e., the table region where the ball will fall) when the opponent performs the actual strike. The predicted intention is then used to select the optimal robot hitting type (e.g., forehand, middle, backhand strike). This system requires specialized hardware, such as Gigabit Ethernet camera sensors with 200 frames per second.

The interest in vision techniques aimed at understanding human activity from videos has also grown (e.g., using YouTube or other large video datasets). In [Wu+15], a group by Cornell University propose a method to decompose complex events into simpler actions with video segmentation, and then learn the sub-action dynamics using an unsupervised graphical model, based on Conditional Random Fields (CRFs)

[7] `http://www.poeticon.eu/`

Figure 15: The Watch-n-Patch assistive robot system, reproduced from [Wu+16]. After spotting an unusual or incomplete action, the robot signals the information to the human user with a laser pointer.

over Kinect v2 data. In [Wu+16], they then demonstrate how Watch-n-Patch[8], an assistive robot with such a previously-trained system on board, can be useful not only to monitor daily human activities, but also to actively remind users of steps and pieces that they might forget in their typical activity sequences. For example, they do that by using a laser pointer to indicate a "forgotten" object (e.g., a milk carton) that was not handled appropriately after usage (e.g., it was not put back in the fridge).

Koppula [KS16] consider the problems of detecting and anticipating human activities by combining complex full-body human trajectories, robot trajectories and object affordances knowledge in a graphical model based on CRFs. That work shows that this kind of model can improve detection and anticipation over human action datasets. The object affordances part in that model consists of a prior grounding specified by the programmer, assigning categories like "drinkable", "pourable", "reachable" to action–object and object–object relations, where the object features are Scale-Invariant Feature Transform (SIFT) [Low99].

The above ideas have also been explored in psychology, for example by Sciutti [Sci+15], where the authors propose a model to make humanoid robots anticipate human partners' intentions, based on actively engaging humans in face-to-face interation and measuring the subtle kinematic movement signals that emerge.

In [CFT16], a group from Georgia Tech analyzes the impact of providing human guidance to a robot while it explores the environment and learns affordances (see Fig. 16), as opposed to having the robot learn them autonomously, like the approaches that we described so far. In a controlled scenario with four household objects that a humanoid robot manipulates with different action parameters, they conclude that a mixed approach (i.e., partially human-guided, partially using self-exploration biased by information previously provided from human teachers) is effective for learning the affordances, requiring fewer interactions than other modalities. A strong limitation of this approach is that it considers affordances as binary values (i.e., does an object offer a specific affordance or not?) rather than probabilistically. As previ-

[8] `http://watchnpatch.cs.cornell.edu/`

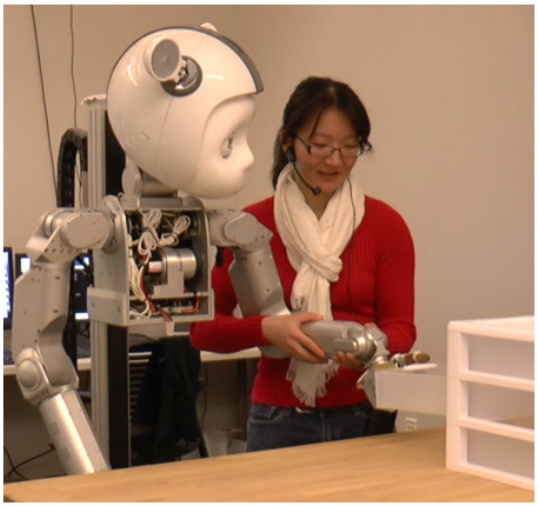


Figure 16: A human guides a robot while it tries motor actions onto world objects to learn their affordances, from [CFT16].

ously mentioned, a probabilistic representation is key in modeling the inherent noise in uncontrolled, human environments.

In this chapter, we have provided the fundamental information needed to understand the contributions of the thesis in the next chapters. Specifically, we have gone through some theory (Bayesian Networks) and we have listed relevant works related to the broad scope of the thesis.

3 EXPERIMENTAL PLATFORM

In this chapter, we provide the practical groundwork for the experiments described in the thesis.

We start by describing the iCub humanoid robot in Sec. 3.1. Then, in Sec. 3.2 we illustrate the experimental scenario adopted in the experiments. Finally, Sec. 3.3 presents our modular system for robot affordance learning, based on autonomous robot exploration of the world and visual processing. The whole system will be used as a building block for the next chapters.

3.1 THE ICUB HUMANOID ROBOT

In this section, we illustrate the humanoid *robot* that we will use to run the experiments of this thesis.

The robot that we use is the *iCub* child-like robot [Met+10], shown in Fig. 17. Its shape is similar to that of a 5-year-old child, with a height of 1 meter, a weight of 27 kilograms, and a head with fully articulated eyes [Bei07]. The structure of the iCub is sophisticated: it has a high number of Degrees of Freedom (DoF)[1], the majority of which are located at the arms and hands, in order to make the robot perform object grasping, dexterous manipulation as well as articulatory gestures. The iCub also has tactile sensors and microphones: these robot sensors are not directly used in this thesis, however they are included and used in some previous studies ([Mon+08; Sal+12]) related to this thesis. It is an *open-source platform*, having been adopted by more than 30 research groups and universities worldwide [Nat+19].

The open-source iCub software is, as such, the work of a large community[2]: 2 million Lines of Code (LoC), hundreds of contributors[3]. Parts of the available software can be used in other applications and platforms (i.e., without the iCub), for example the visual processing algorithms. iCub software modules, such as the ones developed for this thesis, rely on the Yet Another Robot Platform (YARP) middleware [MFN06; Fit+14]. YARP is similar to Robot Operating System (ROS)[4], following the same *middleware* concepts about managing distributed computations across a cluster of heterogeneous computers,

[1] At the time of writing this thesis, the iCub robot is the second humanoid robot with the highest number of DoF (53 DoF), being surpassed only by the ARMAR robot by Karlsruhe Institute of Technology (63 DoF) [Asf+19].

[2] `http://www.icub.org/`, `https://github.com/robotology`

[3] `https://www.openhub.net/p/robotology`

[4] At the time of writing this thesis, ROS [Qui+09] is the *de facto* standard middleware in robotic research.

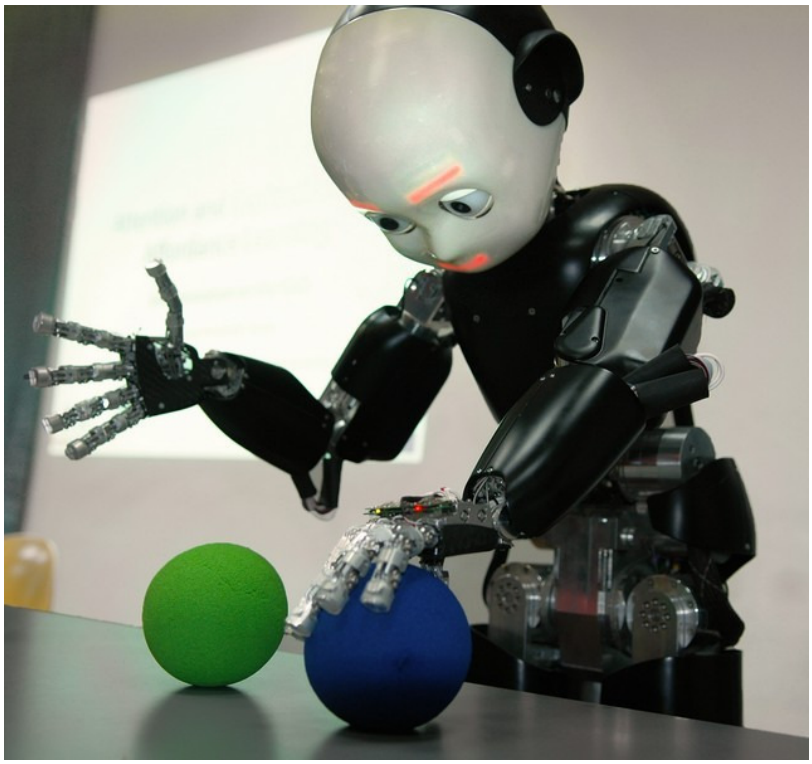


Figure 17: The iCub humanoid robot (picture by Lorenzo Natale).

hardware abstractions, low-level device drivers, message passing between processes, and implementation of commonly used functionality (e.g., geometry, linear algebra, vision algorithms).

The iCub robot was developed for studying cognition and learning. The main idea behind this platform is that it is born with simple skills, and then it can become intelligent over time by interacting with the enviroment, where the term intelligence encompasses manipulation, social skills, and interactive skills. As a result, it can gain a certain degree of autonomy. The usefulness of the iCub is in its completeness: it encompasses movement (high number of motors) and the possibility to measure the external environment (sensors), which in turn permit to study aspects of higher intelligence (algorithms).

3.2 EXPERIMENTAL SCENARIO

In our scenario, the iCub robot is positioned next to a *playground table*, meaning a table which can have one or more objects on top of it. Examples of these objects are colorful toys, sponges and elongated tools. This scenario is designed so that the robot interacts with the objects, it acquires the model of the environment from sensorimotor data, and it can then use the acquired knowledge for reasoning about the environment. Fig. 18 shows an example of our scenario.

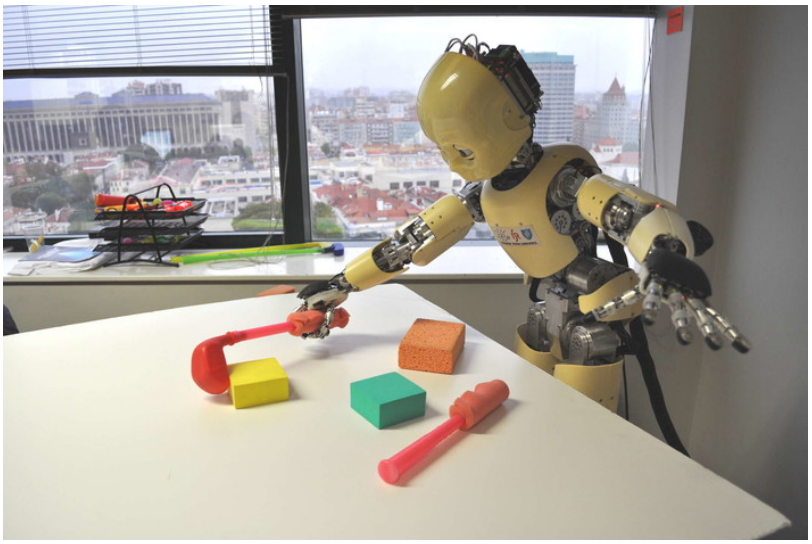
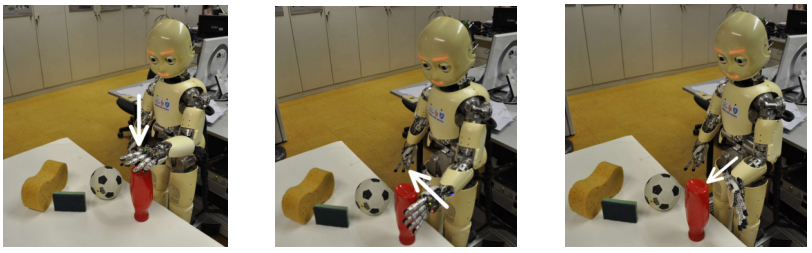


Figure 18: The iCub robot in a playground table scenario. Objects and tools are present in the scenario, so that the robot can interact with them, it can acquire a model of the environment derived from sensorimotor data, and it can use such a model for reasoning about the environment.



(a) Grasp. (b) Tap. (c) Push.

Figure 19: Examples of manipulative motor actions performed by the iCub robot onto environment objects.

In following Montesano's approach (see Sec. 2.2.1), the variables that our model considers are related to: action, object, and effect.

Note that, when a robot interacts with its environment autonomously in a self-exploration fashion, several variables can be actively chosen by the robot, for example the motor action and the target object. In our case, these variables are usually interventional (see p. 26), being set to their specific value during each experiment by the experimenter or by the robot.

Regarding the motor *actions* that can be executed by the iCub during experiments, in this thesis we consider manipulative hand gestures. These are movements performed by the arm–hand chain of the robot, capable of touching objects in the environment from different directions. Fig. 19 shows some examples of these motor actions, highlighted in white.

In this thesis, the low-level control routines to realize the motor behaviors on the iCub robot (both simulated and real) are based on works and software modules previously made available by other researchers in the iCub community [Pat+10; Ron+16].

Regarding the *effects* in the environment that are measured using robot perception, we consider physical displacements of objects on the table, being moved by the effector of robot from the time when the robot performs the action, until a pre-defined fixed duration (number of frames) afterwards. The effector can be the agent's bare hand or the tip of a grasped tool, as we will describe in Ch. 5. The physical displacement of an object is computed as the difference between the final and initial coordinates of that object on a table. Furthermore, we consider two displacement effects: along the lateral and longitudinal direction, respectively. By clustering the continuous sensory values, we define five discrete levels for these effects: Very Positive (VP), Low Positive (LP), No Movement (NM), Low Negative (LN) and Very Negative (VN). These levels correspond to an object moving

- VP: significantly to the left (or front) from the robot's perspective;
- LP: slightly to the left (or front);
- NM: little or no movement;
- LN: slightly to the right (or back);
- VN: significantly to the right (or back).

Recall from p. 22 that Bayesian Networks (BNs) offer the possibility of representing causal relationships, based on knowledge (specified by the experimenter arbitrarily) of the world domain being modeled. In the experimental part of this thesis, the robot chooses an action A, it executes it on an object (with certain features) O, and it "causes" the effect E as a consequence. Since A and O are given as priors, and since effects happen later in time, we attribute the meaning that A and O are causes, whereas E are consequences. Still, from the point of view of BNs, A, O and E are variables with no temporal information. In this sense, as clarified in Sec. 2.1.2, the model does not learn causal relationships.

3.3 SOFTWARE ARCHITECTURE

In this section, we introduce our software framework that permits robots to sense, learn, and use the information contained in surrounding objects as well as their affordances. It is a modular software framework for experiments in visual robot affordances, directed at the robotics, psychophysics and neuroscience communities. Although the system

was developed for the iCub platform in particular, thanks to its modular nature, parts of the system can be used for other robots. We make this framework publicly available[5]. Below, we show how this system can be used for (i) the perception of relevant visual features of a robot's surrounding objects; (ii) reasoning about the affordances of those objects and, as a result, (iii) supporting the new capabilities and behaviors described in the next chapters, for example tool use and action planning.

The basic intuition behind our architecture is that a robot learns links between object shape features and their physical properties: for instance, the notion that spherical objects roll faster than cubic ones when pushed laterally with an effector (e.g., a robot hand or a tool held in the robot's hand).

Because learning is based on a *probabilistic model*, the approach is able to deal with uncertainty, redundancy and irrelevant information. We say that affordances are learned, because we let the robot discover them from autonomous experience in its environment, and then use the learned model in various profitable ways, e.g., prediction, tool use learning, planning a sequence of actions to achieve complex goals such as stacking objects.

As mentioned in Ch. 2, Zech published a systematic taxonomy of robot affordance models [Zec+17]. According to their criteria (defined in their taxonomy), in terms of *perception* the works in this thesis classify as using an agent perspective, meso-level features, first order, stable temporality (i.e., related to static object properties that do not change over time, such as the shape of rigid objects). In terms of *development*: acquisition by exploration, prediction by inference, generalization exploitation by action selection and language, and offline learning.

3.3.1 *Visual Pipeline*

The objective of this section is to illustrate a real-time (30 fps), versatile and easy-to-use visual processing software that can be used in different scenarios such as robot perception, higher-level cognitive reasoning (e.g., reasoning about the possibilities afforded by perceived objects) and related studies.

Starting from the beginning of such a perception and reasoning pipeline, let us first focus on visual features. Computing features of interest of objects present in images is a frequent task in cognitive robotics. For example, for the iCub humanoid robot this fundamental capacity was required during the research projects RobotCub[6] and POETICON++[7], where the robot had to see, characterize and grasp objects located on a playground table scenario, as described in Sec. 3.2. The exact require-

[5] `https://github.com/gsaponaro/robot-affordances`

[6] `http://www.robotcub.org/`

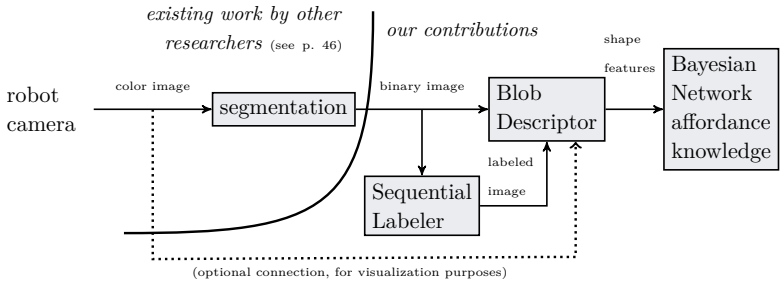


Figure 20: Our pipeline for computing visual affordance features. Boxes indicate software modules, arrows indicate data flow connections, dotted arrows indicate optional data flow connections. See text for details, and Fig. 21 for an example computation.

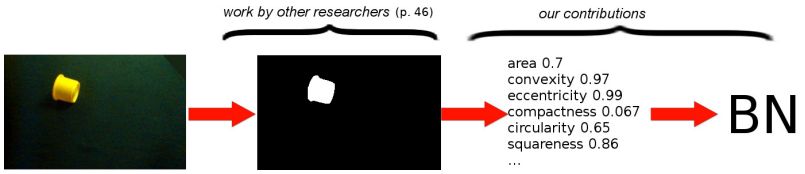


Figure 21: Example computation of extracting salient shape features for affordances. From left to right: robot camera color image, binary segmentation image, shape features, Bayesian Network affordance knowledge. The set of extracted shape features characterizes the object. This set is then used to train the robot affordance knowledge model, and to perform inference queries on an affordance database. See also Fig. 20 for the diagram of software modules.

Table 2: Shape descriptors.

descriptor	definition
area	number of blob pixels, normalized w.r.t. a constant
convexity	ratio between convex hull perimeter and object perimeter
eccentricity	ratio between minor and major axes of best-fit ellipse
compactness	ratio between object area and squared external contour perimeter
circularity	ratio between object area and area of minimum-enclosing circle
squareness	ratio between object area and area of minimum-enclosing rectangle
convexity defects	number of "holes" along blob contour
image moments	weighted averages of the blob pixels' intensities

ments and subtleties vary slightly among different projects, setups and demonstrations.

Recall from Sec. 1.4.1 that the two-streams hypothesis of neuroscience speculates that visual perception occurs across two separate pathways: ventral and dorsal. The former is mainly related to object recognition and categorization, the latter guides object-directed actions (e.g., reaching and grasping). We now illustrate the main components of our robot affordance pipeline, which is inspired by the dorsal pathway, in the sense that it does not require to know or recognize the category of objects, instead it reasons on their low-level shape features, and it permits the agent to act fast.

Fig. 20 shows the organization of our pipeline, where boxes indicate conceptual (and software) modules, arrows indicate data flow connections, and dotted arrows indicate optional data flow connections. Fig. 21 shows an example computation. At the end of the pipeline, shape features are computed. They serve as the data (about world objects) for the Bayesian Network (BN) computational implementation of the affordance knowledge. In particular, using clustering of the continuous sensory values, we define three discrete levels for each shape feature: Low (L), Medium (M), and High (H).

The modules are the following:

SEGMENTATION A visual module that takes as input a color image obtained from a robot camera, and outputs a binary image containing white pixels on the location of the objects, black pixels elsewhere (i.e., background). Because the overall architecture is modular, it is possible to plug and play different segmentation

implementations and algorithms as this module, provided that they have the required input and output format. The implementation that we adopt for the experiments in this thesis (written by other researchers[7]) uses Local Binary Patterns (LBPs) [OPM02] for analyzing the texture of objects on a table in front of the robot, based on comparing the intensity of each pixel with its neighbors and representing this relationship with a histogram of binary numbers.

SEQUENTIAL LABELER A visual module that takes as input a binary segmentation image, and extracts as output the connected components contained therein (i.e., contiguous subsets of pixels). We call the output image *labeled*, meaning that its pixels correspond to identifiers of the segmented objects: pixels are set to zeros over the background, to ones over the first segmented object, to twos over the second segmented object, etc.

BLOB DESCRIPTOR A visual module that, from the binary and labeled images, computes a vector of descriptors for each segmented object present in the scene. This module performs measurements on the connected components (obtained by the two modules above), in order to extract some of their salient characteristics. The features that we extract are pre-categorical shape descriptors [ZL04] computed as geometric relationships between perimeter, area, convex hull and approximated shapes of the segmented silhouettes of the objects in front of the robot. We list these shape descriptors in Table 2.

The different types of images being processed and passed along between the modules are:

COLOR IMAGE as acquired from the robot camera driver;

BINARY SEGMENTATION IMAGE, which separates the objects of interest from the background. This image thus contains the contours of the object shapes, which we call *blobs*;

LABELED IMAGE, containing the uniquely-numbered connected components corresponding to the visual blobs: the pixels of the i^{th} blob are numbered i, the background pixels are numbered zero.

3.3.2 *Impact*

In addition to being applied for this thesis and other works authored by Giovanni Saponaro, our visual affordances system has also been used by external researchers outside our scope, namely:

[7] `lbpExtract` software module, written by Vadim Tikhanoff and available at `https://github.com/robotology/segmentation`

- in a developmental psychology work that links the visual appearance of objects with language learning on an iCub robot [MC16], and
- in studies about robot tool affordances which rely on the *deep learning* paradigm instead of Bayesian Networks [Deh+16a; Deh+16b; Deh+17].

4

AFFORDANCES, GESTURES AND LANGUAGE

In this chapter, we present a computational model (see Fig. 22) that combines object affordances with communication, and we show the benefits of such an approach in cognitive robotic systems.

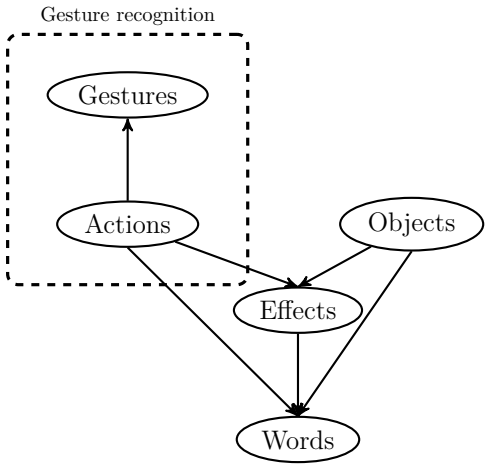


Figure 22: Computational model of affordances with gestures and language.

We consider two aspects of communication: nonverbal (i.e., body gestures) and verbal (i.e., language). It is worth exploring both of these modalities in robots, as means for providing them the skills to engage in sociality and collaboration with humans. In other words, communication is useful for becoming social[1]. By incorporating communication aspects into a cognitive robotic system, we permit the leap from egocentric behavior (where the robot explores its surrounding world) to a social one (where the robot perceives the actions of other agents and links them to its own actions).

Regarding nonverbal communication, we focus on human *gestures* perceived with vision sensors. We developed a gesture recognizer for manipulative hand gestures. This recognizer receives a sequence of camera images depicting a person making manipulative gestures (these images contain the *gesture feature* inputs), and it produces a probability distribution over the gesture being recognized as output[2].

We embed the gesture recognizer into a computational model of object affordances, permitting to extend previous works ([Sal+12], see also

[1] A social robot is "[a robot that is] able to communicate and interact with us, understand and even relate to us, in a personal way. [It] should be able to understand us and itself in social terms" [Bre02].

[2] Implementation details about the gesture recognition model will be given in Appendix A.

Sec. 2.2.2). With our combination of affordances and gestures we show that, after having acquired knowledge of its surrounding environment from autonomous exploration, a humanoid robot can generalize this knowledge to the case when it observes another agent (human partner) performing the same motor actions previously executed by the robot during training. This is the shift from reasoning purely about actions performed by the robot itself (ego-centric phase) to reasoning about actions performed by external human users (social phase).

We also incorporate *verbal language* capabilities into the model, motivated by the observation that human–human cooperation is greatly facilitated and influenced by human language [MC00], therefore language description skills can benefit human–robot cooperation. In addition, having the verbal language component in our computational model allows us to visualize the results produced by the robot from a different angle.

Throughout this chapter, we use the following *terminology*, in accordance to a review by Aggarwal on human activity recognition [AR11]. Human activities can be categorized into different levels with increasing level of complexity. *Gestures* are elementary movements of a person's body part, and are the atomic components describing the meaningful motion of a person. *Actions* are single-person activities that may be composed of multiple gestures organized temporally, such as walking or waving. *Interactions* are activities that involve two or more persons and/or objects.

We make the code and data from this chapter publicly available[3] in the interest of reproducibility.

This chapter is the subject of the following publications:

- Giovanni Saponaro, Lorenzo Jamone, Alexandre Bernardino, and Giampiero Salvi. "Interactive Robot Learning of Gestures, Language and Affordances". In: *Workshop on Grounding Language Understanding*. Satellite of Interspeech. 2017, pp. 83–87. DOI: 10.21437/GLU.2017-17.

- Giovanni Saponaro, Lorenzo Jamone, Alexandre Bernardino, and Giampiero Salvi. "Beyond the Self: Using Grounded Affordances to Interpret and Describe Others' Actions". In: *IEEE Transactions on Cognitive and Developmental Systems* (2019). DOI: 10.1109/TCDS.2018.2882140.

The outline of this chapter is as follows. Sec. 4.1 gives motivations for building models that jointly consider object affordances and communication (nonverbal and verbal). Sec. 4.2 lists related works from the robotic literature implementing this fusion. Sec. 4.3 presents our proposed approach for combining object affordances, nonverbal communication (gestures), and verbal language. In Sec. 4.4 we report the

[3] https://github.com/gsaponaro/tcds-gestures: code from [Sap+19].

experimental results, and finally in Sec. 4.5 we draw our conclusions and possible future extensions.

4.1 MOTIVATION

Communication is defined as "a process by which information is exchanged between individuals through a common system of symbols, signs, or behavior"[4]. In this section, we motivate why combining object affordances with communication (gestures and language) can be beneficial in cognitive robotic systems, as already hinted in the beginning of this chapter and in Sec. 1.3.

The common system of symbols existing between individuals during communication can be encoded by nonverbal aspects (e.g., body gestures) as well as verbal ones (i.e., natural human language). Both nonverbal gestures and verbal words have specific motivations to be incorporated in robot perception algorithms and robot cognitive capabilities. By relying on a gesture recognizer, we augment the computational affordance model of Sec. 2.2.1 with gestures, permitting the shift from reasoning about actions performed by the robot itself (ego-centric phase) to reasoning about actions performed by external users (social phase). In addition, we incorporate language into the model, allowing to estimate the probability of words given other observed variables. This kind of reasoning over language is useful for human interpretability, because it allows to generate verbal descriptions of experimental data. It also shows how our model can exhibit semantic language properties: the choice of relevant words to describe a scene, the choice of synonyms, and of congruent/incongruent conjunctions.

We now give some motivations for incorporating gestures and words, respectively, in cognitive robotic systems.

Gestures expose the role of physical movement in communication and interaction[5]. Humans learn to use gestures during their first year of age, even before they learn to speak [TCL07]. Psychology has studied how humans interact with body gestures for many activities and purposes [McN96; MC99], including: greeting, leaving, showing agreement or disagreement, threatening, emphasizing a spoken sentence, physically pointing at something or someone. In this chapter, we employ a gesture recognition model capable of recognizing the manipulative gesture made by a person probabilistically[2].

[4] `https://www.merriam-webster.com/dictionary/communication`

[5] Human gestures can be *static* or *dynamic*. During static gestures, body joints do not move: examples are pointing, or displaying a number with the fingers. Instead, in dynamic gestures, body joints move: for example during waving and clapping. In the case of sign languages, a gesture can have both static and dynamic elements. In this chapter *we draw our attention to dynamic gestures*, which, due to their rapidly evolving nature, are particularly relevant in the manipulative scenarios that we consider, introduced in Sec. 3.2.

Figure 23: Proof of concept of a robot (left) recognizing a human struggling while opening a bottle (right): the robot intervenes, providing help. Picture elaborated from `http://flic.kr/p/b8bbYZ` with permission from the original owner.

Verbal language is another fundamental aspect of human communication that is useful to model in machines, particularly for the collaborative aspect. A child acquires the skill of coordinating with peers or adult caregivers in shared problem-solving activities and social games (therefore, to *collaborate*) around the second year of life [BRZ06]: this is achieved not only by mere behavioral coordination, but also by employing communicative strategies [MS10] and by continuously observing partners' actions [RM04].

Even though social robots are becoming common in domestic and public environments, human–robot teams still lag behind human–human teams in terms of effectiveness. For robots, interpreting the actions of others and learning to describe them verbally (for effective cooperation) is challenging. One reason is that we cannot possibly model all the imaginable verbal cues that can take place during human–robot interaction, due to the richness of language and the high variability of the real world outside of structured research laboratories and factories. A viable alternative is to have robots that *learn* world elements and properties of language [Iwa07], and the ability to link these verbal elements with other skills, such as other perceptual modalities (e.g., vision of objects and other agents) and manipulation abilities (e.g., grasping objects and placing them in order to achieve a goal) [Ste03].

A key motivation for using the above-mentioned sources of information (object affordances, words, gestures) in robot algorithms, is to support *activity recognition* of human agents. For example, this is useful for revising the belief that a certain action occurred, given the observed effects of the human action onto physical objects (correction of action estimation). In addition, a robot can anticipate the effects when the action has only been partially observed (early action recognition). Equipping robots with these prediction capabilities allows them

to anticipate effects before action completion, thus enabling interactions between human and robot to be uninterrupted and natural.

In turn, action recognition and prediction abilities serve: (i) to predict what is going to happen, (ii) to understand the *motivation* beyond the others' action (mental simulation), and (iii) to provide feedback or commentary by an automated (possibly robotic) system. Fig. 23 sketches an example of these uses. Inherent in this motivation is the leap from an ego-centric phase to a social one, permitting agents to reason about the actions of others, and to describe them verbally. In this chapter, we illustrate our implementation of this process.

4.2 RELATED WORK

This section describes works that are related to the scope of this chapter, that is, the combination of object affordances with communication (nonverbal and verbal) in cognitive robotic systems.

First, in Sec. 4.2.1 we describe the model by Salvi in greater detail than in Ch. 2. That work is the building block that this chapter extends. Then, in Sec. 4.2.2 we go through other works in the literature that use robot affordances and communication.

4.2.1 *Affordances and Language*

In Sec. 2.2.2, we have introduced the Affordance–Words model by Salvi [Sal+12].

Recall that Salvi proposes a joint model to learn robot affordances *together with word meanings*. It uses a Bayesian probabilistic framework to allow a robot to ground the basic world behavior and verbal descriptions associated to it, as shown in Fig. 11 on p. 31 and Fig. 12 on p. 33. The data used for learning such a model is obtained from robot manipulation experiments. Each experiment is associated with a number of alternative verbal descriptions uttered by two human speakers according to a pre-defined grammar, for a total of 1270 recordings.

Note, however, that in [Sal+12] no grammar was used during the learning phase: the speech recognizer used as a frontend to the spoken descriptions is based on a loop of words with no grammar, and the Affordance–Words model is based on a bag-of-words assumption, where only the presence or absence of each word in the description is considered.

The data in Salvi's work is acquired from a robot's *ego-centric perspective*, meaning that the robot learns a model by interacting with the environment by self-exploration (see also Sec. 1.2 and Sec. 3.2), and then it reasons about its own actions.

In this Affordance–Words model, the world behavior is defined by random variables, following the probabilistic machinery introduced in Ch. 2. Table 3 presents a list of variables and their possible values.

Table 3: Symbolic variables of the Affordance–Words Bayesian Network (from [Sal+12]), with the corresponding discrete values obtained from clustering during robot exploration of the environment. We call *word variables* the booleans of the last row, whereas we call *affordance variables* all the other symbols. See also Fig. 24.

symbol	name: description	values
a	Action: motor action	grasp, tap, touch
f_1	Color: object color	blue, yellow, green1, green2
f_2	Size: object size	small, medium, big
f_3	Shape: object shape	sphere, box
e_1	ObjVel: object velocity	slow, medium, fast
e_2	HandVel: robot hand velocity	slow, fast
e_3	ObjHandVel: relative object–hand velocity	slow, medium, fast
e_4	Contact: object hand contact	short, long
w_1–w_{49}	presence of each word in the verbal description	true, false

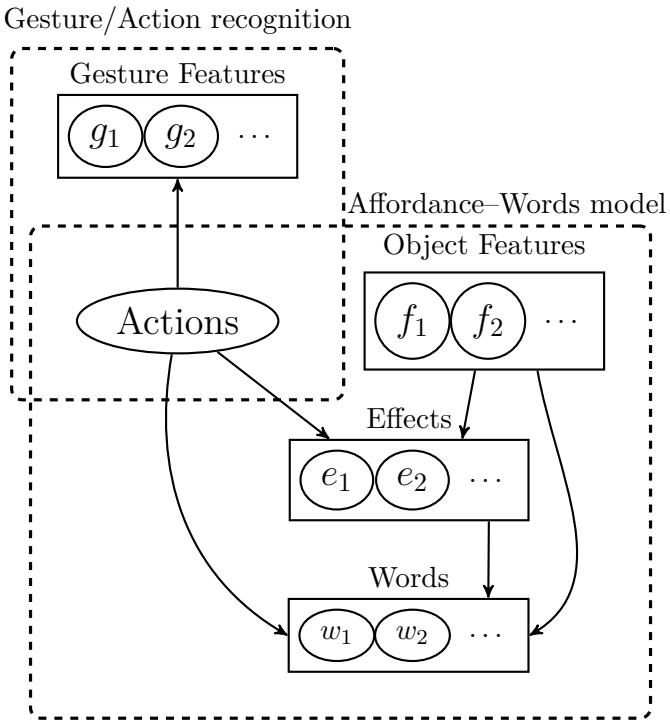


Figure 24: Abstract representation of the probabilistic dependencies in our model which integrates affordances, gestures and language. See also Table 3.

All variables are discrete or are discretized from continuous sensory variables through clustering in a preliminary learning phase.

Note that the name of the possible values have been assigned by the researchers arbitrarily to the clusters, for the sake of making the results more human-interpretable. However, the robot has no prior knowledge about the meaning of these clusters nor about their order, in case they correspond to ordered quantities.

The variables can be divided according to their use: affordance variables and word variables. Affordance variables are actions variables $A = \{a\}$, object feature variables $F = \{f_1, f_2, \dots\}$, and effect variables $E = \{e_1, e_2, \dots\}$. Word variables are $W = \{w_1, w_2, \dots\}$.

To simplify the notation, let us call

$$\begin{aligned} X &= \{A, F, E, W\} \\ &= \{a, f_1, f_2, \dots, e_1, e_2, \dots, w_1, w_2, \dots\} \end{aligned}$$

the set of affordance and word variables. Consequently, the relationships between words and concepts are expressed by the joint probability distribution $p(X) = p(A, F, E, W)$ of actions, object features, effects, and words in the spoken utterance.

This joint probability distribution, illustrated by the dashed box labeled Affordance–Words model of Fig. 24, is estimated by the robot in an ego-centric way through interaction with the environment. The

dependency structure and the model parameters are estimated by the robot in an ego-centric way through interaction with the environment. As a consequence, *during learning, the robot knows* a priori *what action it is performing with certainty, and the variable A assumes a deterministic value*. During inference, the probability distribution of the variable A can be inferred from evidence on the other variables. For example, if the robot is asked to make a spherical object roll, it will be able to select the action tap as most likely to obtain the desired effect, based on previous experience.

There is no one-to-one correspondence between affordance nodes and words in [Sal+12]. Each word is connected with many affordance nodes, that constitute the word's significant (for example, the word "ball" is not only connected to the shape object feature, but also to action and effect).

The lack of correspondence between affordance nodes and words was partly emerging from the natural variability that is inherent in the way humans describe situations in spoken words. It was also a design choice, because in that work the authors wanted to prove that the model was not merely able to recover simple word–meaning associations, but was able to cope with more natural spoken utterances. Consequently, in the spoken descriptions: (i) there are many synonyms for the same concept: for instance, cubic objects are called "box", "square" or "cube". Also, actions and effects are described using different tenses ("is grasping", "grasped", "has (just) grasped"); (ii) different affordance variable values may have the same associated verbal description, e.g., two color clusters corresponding to different shades of green are both referred to as "green"; (iii) finally, many affordance variable values have no direct description: for example, the object velocity and object–hand velocity (slow, medium, fast), or the object–hand contact (short, long) are never described directly, and need to be inferred from the situation.

The Affordance–Words model does not account for the concepts of parts of speech, verb tenses or *temporal aspects* explicitly. For example, the words "is", "grasping", "has", "grasped", "just", and so on, are initially completely equivalent to the model, which has no prior information about what verbs, adjectives or nouns are, nor about similarity between words. It is only through the association with the other robot observations that the model realizes that "grasping" has the same meaning as "grasped"[6]. The following three phrases, which were used interchangeably in the experiments by [Sal+12], are mapped to exactly the same meaning, after learning: (i) "is grasping", (ii) "has grasped", (iii) "grasped". Note that the model *per se* would be fully capable to distinguish between those phrases, provided that they were used in different situations, which however was not the case in the experimental data.

[6]The model of [Sal+12] has no concept of past, present and future, and cannot distinguish between tenses.

The above assumption of knowing the action with certainty during learning, is relaxed in the proposed approach presented further down in this chapter, in Sec. 4.3, by extending the model to the observation of external (human) agents. In doing this extension, we introduce a *social perspective* where the robot reasons about other agents.

4.2.2 *Other Works*

A few works have studied the potential coupling between learning robot affordances and *language grounding* (where grounding refers to linking the symbolic nature of language with the sensorimotor experience of a robot). The union of robot affordances with language grounding gives new skills to cognitive robots, such as: creation of categorical concepts from multimodal association obtained by grasping and observing objects, while listening to partial verbal descriptions [NNI09; Ara+12], learning the association of spoken words with sensorimotor experience [MC16], linking language with sensorimotor representations [Str+16], or carrying out complex tasks (which require *planning* of a sequence of actions) expressed in natural language instructions to a robot. The planning aspect will be the topic of Ch. 6.

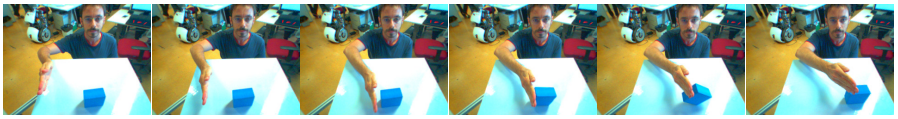
In other works, both object-directed action recognition in external agents [KGS13] and the incorporation of language in human–robot systems [Har90; Mat+14] have received ample attention, for example using the concept of *intuitive physics* [Lak+17; Gao+18] to be able to predict outcomes from real or simulated interactions with objects.

DeepMind and Google published a method [San+17] to perform relational reasoning on images, i.e., a system that learns to reflect about entities and their mutual relations, with the ability of providing answers to questions such as "Are there any rubber things that have the same size as the yellow metallic cylinder?". That work is very powerful from the point of view of cognitive systems, vision and language. Our approach is different because (i) we focus on *robotic* cognitive systems, including manipulation and the uncertainties inherent to robot vision and control, and (ii) we follow the developmental paradigm and the embodiment hypothesis (see Sec. 1.2), meaning that, leveraging the fact that a human and a humanoid produce actions with similar effects, we relate words with the robot's *sensorimotor* experience, rather than sensory only (purely images-to-text).

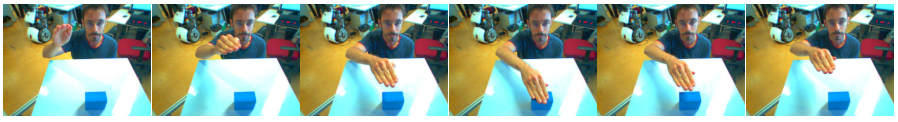
4.3 PROPOSED APPROACH

In this section, we explain our approach for combining object affordances with communication (nonverbal and verbal) in cognitive robotic systems. This combination builds upon the intuition that a robot can use its previously-acquired knowledge of the world (e.g., motor actions, objects properties, physical effects, verbal descriptions) to those situa-

(a) Grasp: moving the hand towards an object vertically, then grasping and lifting it.



(b) Tap: moving the hand towards an object laterally then touching it, causing a motion effect.



(c) Touch: moving the hand towards an object vertically, touching it (without grasping), then retracting the hand.

Figure 25: Examples of human manipulative actions from the point of view of the robot.

tions where it observes a human agent performing familiar actions in a shared human–robot scenario.

4.3.1 *Staged Developmental Process*

Our method is a staged developmental process from a self-centered, individualistic learning, to socially aware learning. This transition happens gradually in subsequent phases.

In the first phase, the system engages in manipulation activities with objects in its environment (following Montesano's approach, as described in Sec. 2.2.1). The robot learns object affordances by associating object properties, actions and the corresponding effects.

In a second phase, the robot interacts with a human who uses spoken language to describe the robot's activities (following Salvi's approach, as described in Sec. 4.2.1). Here, the robot interprets the meaning of the words, grounding them in the action–perception experience acquired so far. Although this phase can already be considered *social* for the presence of a human *narrator*, it is still self-centered, because the robot is still learning how to interpret its own actions.

In the last phase, which is our contribution, the system turns to observing human actions of a similar nature as the ones explored in the first phases. We consider three *manipulative gestures*, displayed in Fig. 25, corresponding to physical actions performed by an agent onto objects on a table, in a similar fashion as in Sec. 3.2. The robot reuses the experience acquired in the first phases to interpret the new observations between its own actions and the actions performed by the human. In this phase, human movements are interpreted using the

experience acquired so far, and they are incorporated into the model using a gesture recognizer[2].

4.3.2 *Combining Affordances and Communication*

Starting from the Affordance–Words computational model by Salvi [Sal+12], we propose a way to fuse two sources of information (about the self and about others) in a fully probabilistic manner. This addition allows to perform fine-grained types of inferences and reasoning, by doing predictions over affordances and words when observing another agent with uncertainty.

In extending Salvi's model, we relax the assumption (described in Sec. 4.2.1) that the action is known during the learning phase. That assumption is acceptable when the robot learns through self-exploration and interaction with the environment, but must be relaxed if the robot needs to generalize the acquired knowledge through the observation of another (human) agent. We estimate the action performed by a human user during a human–robot collaborative task, by employing a human gesture recognition algorithm[2]. This provides two advantages. First, we can infer the executed action during training. Second, at testing time we can merge the action information obtained from gesture recognition with the information about affordances.

To permit the transfer from robot self-centered knowledge to human knowledge to work, we assume that the *same actions*, performed on objects with the *same properties*, cause the *same effects* and are described by the *same words*. In other terms, all of the variables under consideration, listed in Tab. 3, are the link between robot and human.

In our formulation and in our implementation, we will hinge on the existence of the discrete action variable, the value of which is known to the robot in the ego-centric phase of learning, but must be inferred when observing human actions.

The *gesture recognition model* (that will be fully detailed in Appendix A) is based on a statistical algorithm called Hidden Markov Model (HMM): therefore, we denote the probabilities obtained by the gesture recognizer as $p_{\text{HMM}}(\cdot)$. The input of this model is a sequence of T gesture feature vectors (the sequence going from image frame 1 to image frame T), which we define as G_1^T. Thus, $p_{\text{HMM}}(A \mid G_1^T)$ denotes the probability distribution over the actions recognized by this model, given gesture features from 1 to T. For example, for a certain input we can obtain that $p_{\text{HMM}}(A \mid G_1^T)$ corresponds to the following action probabilities summing up to one: grasp 0.8, tap 0.15, touch 0.05.

We define the Affordance–Words model as $p_{\text{BN}}(A, F, E, W)$. Our goal is to combine the information from $p_{\text{BN}}(A, F, E, W)$ and $p_{\text{HMM}}(A \mid G_1^T)$ into a single probabilistic model $p_{\text{comb}}(A, F, E, W \mid G_1^T)$, that is, the joint probability of all the affordance and word variables, given that we observe a certain action performed by the human.

The two models can be combined by having the gesture recognizer (Gesture HMMs) provide a posterior distribution to the Bayesian Network (BN). The posterior distribution represents a probabilistic or soft decision [PPD06], as opposed to a deterministic hard decision (which would consider only the top result with full confidence, and would be in fact a less general case, that we described in [Sap+17a]).

Recall from Sec. 4.2.1 that we call $X = \{A, F, E, W\}$ the set of affordance and word variables $\{a, f_1, f_2, \dots, e_1, e_2, \dots, w_1, w_2, \dots\}$. During inference, we have a (possibly empty) set of observed variables $X_{\text{obs}} \subseteq X$, and a set of variables $X_{\text{inf}} \subseteq X$ on which we wish to perform the inference. In order for the inference to be non-trivial, it must be $X_{\text{obs}} \cap X_{\text{inf}} = \varnothing$, that is, we should not observe any inference variable. According to the BN alone, without the gesture recognizer, the inference will compute the probability distribution of the inference variables X_{inf} given the observed variables X_{obs} by marginalizing (see p. 23) over all the other latent variables $X_{\text{lat}} = X \setminus (X_{\text{obs}} \cup X_{\text{inf}})$, where $\setminus$ is the set difference operation:

$$p_{\text{BN}}(X_{\text{inf}} \mid X_{\text{obs}}) = \sum_{X_{\text{lat}}} p_{\text{BN}}(X_{\text{inf}}, X_{\text{lat}} \mid X_{\text{obs}}).$$

If we want to combine the evidence brought by the BN with the evidence brought by the gesture recognizer, there are two cases that can occur:

1. the variable action is included among the inference variables: $A \in X_{\text{inf}}$, or

2. the variable action is not included among the inference variables: $A \in X_{\text{lat}}$.

Here, we are excluding the case where we observe the action directly ($A \in X_{\text{obs}}$) for two reasons. First, this would correspond to the robot performing the action by itself, whereas we are interested in interpreting other people's actions, which is a necessary skill to engage in social collaboration with humans. Second, this would make the evidence on the gesture features G_1^T irrelevant, because in the model of Fig. 24, there is a tail-to-tail connection (see p. 24) from G_1^T to the rest of the variables through the action variable, which means that, given the action, all dependencies to the gesture features are dropped.

The two cases 1., 2. enumerated above can be addressed separately when we do inference. In the first case, we call X'_{inf} the set of inference

variables excluding the action A, that is, $X_{\text{inf}} = \{X'_{\text{inf}}, A\}$. We can write:

$$
\begin{aligned}
&p_{\text{comb}}(X_{\text{inf}} \mid X_{\text{obs}}, G_1^T) = p_{\text{comb}}(A, X'_{\text{inf}} \mid X_{\text{obs}}, G_1^T) = \\
&= \sum_{X_{\text{lat}}} p_{\text{comb}}(A, X'_{\text{inf}}, X_{\text{lat}} \mid X_{\text{obs}}, G_1^T) = \\
&= \sum_{X_{\text{lat}}} \Big[p_{\text{BN}}(A, X'_{\text{inf}}, X_{\text{lat}} \mid X_{\text{obs}}, G_1^T) \\
&\qquad\qquad p_{\text{HMM}}(A, X'_{\text{inf}}, X_{\text{lat}} \mid X_{\text{obs}}, G_1^T) \Big] = \\
&= \left[\sum_{X_{\text{lat}}} p_{\text{BN}}(A, X'_{\text{inf}}, X_{\text{lat}} \mid X_{\text{obs}}) \right] p_{\text{HMM}}(A \mid G_1^T) = \\
&= p_{\text{BN}}(X_{\text{inf}} \mid X_{\text{obs}}) p_{\text{HMM}}(A \mid G_1^T). \qquad (17)
\end{aligned}
$$

This means that we can evaluate the two models independently, then multiply the distribution that we obtain from the BN (over all the possible value of the inference variables) by the gesture posterior for the corresponding value of the action.

In the second case, where the action is among the latent variables, we define, similarly, $X_{\text{lat}} = \{A, X'_{\text{lat}}\}$, and we have:

$$
\begin{aligned}
&p_{\text{comb}}(X_{\text{inf}} \mid X_{\text{obs}}, G_1^T) = \\
&= \sum_{\{A, X'_{\text{lat}}\}} p_{\text{comb}}(X_{\text{inf}}, A, X'_{\text{lat}} \mid X_{\text{obs}}, G_1^T) = \\
&= \sum_{\{A, X'_{\text{lat}}\}} \Big[p_{\text{BN}}(X_{\text{inf}}, A, X'_{\text{lat}} \mid X_{\text{obs}}, G_1^T) \\
&\qquad\qquad p_{\text{HMM}}(X_{\text{inf}}, A, X'_{\text{lat}} \mid X_{\text{obs}}, G_1^T) \Big] = \\
&= \sum_{\{A, X'_{\text{lat}}\}} \Big[p_{\text{BN}}(X_{\text{inf}}, A, X'_{\text{lat}} \mid X_{\text{obs}}) p_{\text{HMM}}(A \mid G_1^T) \Big] = \\
&= \sum_{A} \left[p_{\text{HMM}}(A \mid G_1^T) \sum_{X'_{\text{lat}}} p_{\text{BN}}(X_{\text{inf}}, A, X'_{\text{lat}} \mid X_{\text{obs}}) \right] = \\
&= \sum_{A} \Big[p_{\text{HMM}}(A \mid G_1^T) p_{\text{BN}}(X_{\text{inf}}, A \mid X_{\text{obs}}) \Big]. \qquad (18)
\end{aligned}
$$

This time, we first need to use the BN to do inference on the variables X_{inf} and A, and then we marginalize out (see p. 23) the action variable A after having multiplied the probabilities by the gesture posterior.

4.3.3 *Verbal Descriptions*

In this section, we describe the *verbal language description* capabilities of the combined model described in Sec. 4.3.2. These capabilities are made possible by reasoning on the co-occurring verbal descrition of the experiments, linking affordance variables to word variables.

Verbal descriptions allow the robot to:

- use language in order to determine the mapping between human and own actions, and learn the corresponding perceptual models;
- use the affordance variables to infer the above mapping even in the absence of verbal descriptions;
- once the perceptual models for human actions are acquired, use the combined model (BN and gestures) to do inference on any variable given some evidence.

Such a system makes a robot able to describe the actions of human agents with human language, given some input evidence about the words being uttered and about the visual signals that are detected in the scene.

We use the following notation in order to distinguish between the values of the affordance nodes (all but the last row in Table 3) and the words (last row in the table). Words and sentences are always enclosed in quotation marks. For example, "sphere" refers to the spoken word, whereas sphere refers to the value of the Shape variable corresponding to the specific cluster. Similarly, "grasp" corresponds to a spoken word, whereas grasp correponds to a value of the action variable.

In Sec. 4.3.3.1 we specify the grammar, then in Sec. 4.3.3.2 we outline how we generate and score the verbal descriptions generated from it.

4.3.3.1 *Grammar Definition*

The model described above defines a probability distribution over words, given evidence from the scene. Therefore, it can be used to inspect the understanding by the robot of the current situation. However, interpreting those probability distributions can be hard. For this reason, we have augmented the model with a Context-Free Grammar (CFG)[7] that allows us to generate human-readable descriptions from the evidence encoded by the model.

Here, we provide the *grammar definition*.

As a note, recall from Sec. 4.2.1 that in [Sal+12], therefore also in this chapter, no grammar was used during learning: the model is based on a bag-of-words assumption, where only the presence or absence of each word in the description is considered (see Sec. 4.3.3.2). In other words, the CFG will be useful *for interpreting* the results that involve semantic language properties in a human-readable manner (see Sec. 4.4.2), but those results come from our developmental model, not from the grammar itself.

The pre-defined Context-Free Grammar uses the following notation. The symbol `.|.` represents alternative items, while the symbol `[.]` optional items. Non-terminal symbols are given between `<.>`, while

[7] A CFG is a set of recursive rewriting rules (also called productions) used to generate patterns of strings [Sip12].

words (terminal symbols) are given in plain text and font: thus, the full set of words is given by all the plain text words below.

⟨*sentence*⟩ ::= ⟨*agent*⟩ ⟨*action*⟩ ⟨*object*⟩ ⟨*conjunction*⟩ ⟨*object*⟩ ⟨*effect*⟩

⟨*agent*⟩ ::= the robot | he | baltazar

⟨*action*⟩ ::= ⟨*touch*⟩ | ⟨*poke*⟩ | ⟨*tap*⟩ | ⟨*push*⟩ | ⟨*grasp*⟩ | ⟨*pick*⟩

⟨*touch*⟩ ::= touches | [has] [just] touched | is touching

⟨*poke*⟩ ::= pokes | [has] [just] poked | is poking

⟨*tap*⟩ ::= taps | [has] [just] tapped | is tapping

⟨*push*⟩ ::= pushes | [has] [just] pushed | is pushing

⟨*grasp*⟩ ::= grasps | [has] [just] grasped | is grasping

⟨*pick*⟩ ::= picks | [has] [just] picked | is picking

⟨*object*⟩ ::= the [⟨*size*⟩] [⟨*color*⟩] ⟨*shape*⟩

⟨*size*⟩ ::= big | small

⟨*color*⟩ ::= green | yellow | blue

⟨*shape*⟩ ::= sphere | ball | cube | box | square

⟨*conjunction*⟩ ::= and | but

⟨*effect*⟩ ::= ⟨*inertmove*⟩ | ⟨*slideroll*⟩ | ⟨*fallrise*⟩

⟨*inertmove*⟩ ::= is inert | is still | moves | is moving

⟨*slideroll*⟩ ::= slides | is sliding | rolls | is rolling

⟨*fallrise*⟩ ::= rises | is rising | falls | is falling

4.3.3.2 *Generation and Scoring*

In order to illustrate the language capabilities of the model, rather than displaying the probability distribution of the words inferred by the model, we use the CFG described in Sec. 4.3.3.1 to generate written descriptions of the robot observations, on the basis of those probabilities.

With our approach, by merging the Affordance–Words model and the gesture recognition model, we allow the robot to *reinterpret* the concepts that it has learned in the self-centered phase, but we do not add any new words to the model. Consequently, the descriptions that the model generates when observing humans use the same words to describe the agent (see also Sec. 4.4.2).

The textual descriptions are generated as follows. Given some evidence X_{obs} that we provide to the model (not including any W variables) and some human observation features G_1^t extracted from frames 1 to t, we extract the generated word probabilities $p(w_i \mid X_{\text{obs}}, G_1^t)$. We generate N sentences randomly from the CFG using the `HSGen` tool from HTK [You+06]. Then, the sentences are re-scored according to the log-likelihood of each word in the sentence, normalized by the length of the sentence:

$$\text{score}(s_j \mid X_{\text{obs}}, G_1^t) = \frac{1}{L_j} \sum_{k=1}^{L_j} \log p(w_{jk} \mid X_{\text{obs}}, G_1^t), \tag{19}$$

where s_j is the jth sentence, L_j is the number of words in the sentence s_j, and w_{jk} is the kth word in the sentence s_j. Finally, an N-best list of possible descriptions is produced by sorting the scores.

4.4 EXPERIMENTAL RESULTS

In this section, we provide experimental results obtained with our combined model of affordances and communication. First, in Sec. 4.4.1 we focus on the results made possible by incorporating gestures into the Affordance–Words model by Salvi, permitting the social leap towards the observation of other agents. Then, in Sec. 4.4.2 we report the verbal descriptions results in the form of human-interpretable sentences.

4.4.1 *Combining Affordances and Communication*

Because our combined model is based on Bayesian Networks (see Sec. 2.1.2), it can make inferences over any set of its variables X_{inf}, given any other set of observed variables X_{obs}.

In particular, the model can do reasoning on the elements that constitute our computational concept of affordances. Referring to Fig. 24, these are action, object features, and effect elements, as well as words. We present the following types of results:

- inferences over affordance variables (i.e., over all the entries of Table 3 except the last row therein) in Sec. 4.4.1.1, 4.4.1.2, 4.4.1.4;
- predictions of word probabilities (i.e., predictions of the last row entry of Table 3) in Sec. 4.4.1.3;
- verbal descriptions generated from the word probabilities of the previous point, according to a Context-Free Grammar (CFG) (see footnote [7] on p. 62). These descriptions are useful for clear human interpretation. They serve as a way to observe the emergence (from the model) of certain language phenomena: Sec. 4.4.2.1, 4.4.2.2, 4.4.2.3.

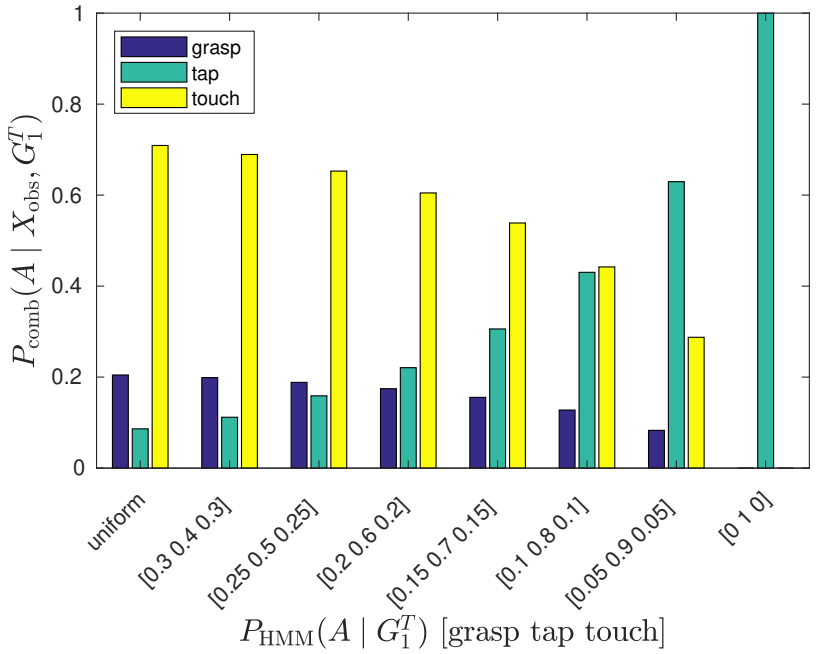


Figure 26: Affordances and gestures combined model: inference over action given the evidence $X_{\text{obs}} = \{\text{Size} = \text{small}, \text{Shape} = \text{sphere}, \text{ObjVel} = \text{slow}\}$, combined with different probabilistic soft evidence about the action.

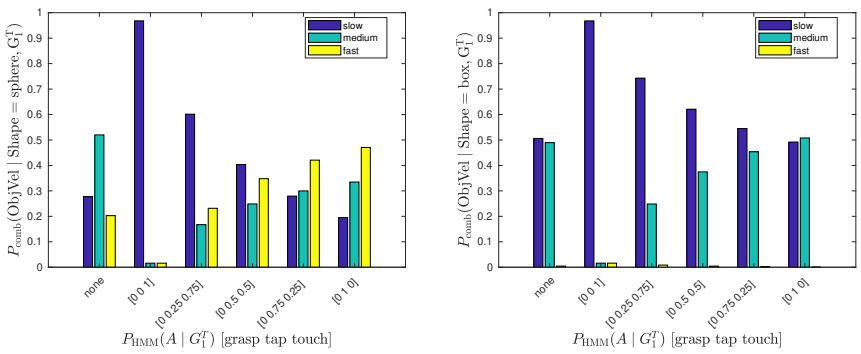
4.4.1.1 *Inference over Action*

In this experiment, we test the ability of our approach to recognize actions. Both the Affordance–Words model and the gesture recognizer can each perform inference of the action variable individually: the former by using the variables of Tab. 3, the latter by using human gesture features. We show how our combined model performs the inference over action in a joint way. This includes dealing with information with different degrees of confidence, or conflicting information.

Let us consider the evidence $X_{\text{obs}} = \{$ Size=small, Shape=sphere, ObjVel=slow $\}$. This corresponds to an experiment that involves a small ball which, after the manipulative action, exhibits a low velocity. Fig. 26 displays the inference over the action variable by our model.

Based on the evidence, the affordance model alone gives the highest probability $p_{\text{BN}}(A \mid X_{\text{obs}})$ to the action *touch*, which usually (in training) does not result in any movement of the object. However, in this particular situation, let us further assume that the action performed by the human was an (unsuccessful) *tap*, that is, a tap that does not result in any movement for the object.

In the figure, we show the effect of augmenting the inference with information from the gesture recognizer, that is, computing (17) (in the case where the action variable is included among the inference variables). We analyze the effect of varying the degree of confidence of the gesture classifier. We start from a uniform posterior $p_{\text{HMM}}(A \mid G_1^T)$,



(a) Predictions with a sphere object. (b) Predictions with a box object.

Figure 27: Affordances and gestures combined model: inference over the object velocity effect of different objects, when given probabilistic soft evidence about the action.

corresponding to a poor classifier, and gradually increase the probability of the correct action until it reaches 1. In this particular example, in order to win the belief of the affordance model, the gesture recognizer needs to be very confident ($p_{\text{HMM}}(A = \text{tap} \mid G_1^T) > 0.81$).

4.4.1.2 *Inference over Effects*

We now show how our approach does inference over variables other than the action one. This corresponds to computing (18) (in the case where the action variable is not among the inference variables, but it is among the latent variables).

We will run this test by using different degrees of probabilistic confidence about the action, and analyzing the outcome in terms of velocity prediction. This experiment exposes that *all* the variables of Tab. 3 jointly link robot and human, not only the action variable, for the reasons expressed in Sec. 4.3.2.

Fig. 27 shows the considered inference in two cases: when the prior information indicates that the shape is spherical (see Fig. 27a), and when it is cubic (see Fig. 27b).

The leftmost distribution in both figures shows the prediction of object velocity from the Affordance–Words model alone, without any additional information. When the shape is spherical, the model is not sure about the velocity, whereas if the shape is cubic, the model does not expect high velocities. If we add clear evidence on the action *touch* from the gesture recognizer, suddenly the combined model predicts slow velocities in both cases, as expected. However, if the action recognition evidence is gradually changed from *touch* to *tap*, the predictions of the model depend on the shape of the object. Higher velocities are expected for spherical objects that can roll, compared to cubic objects.

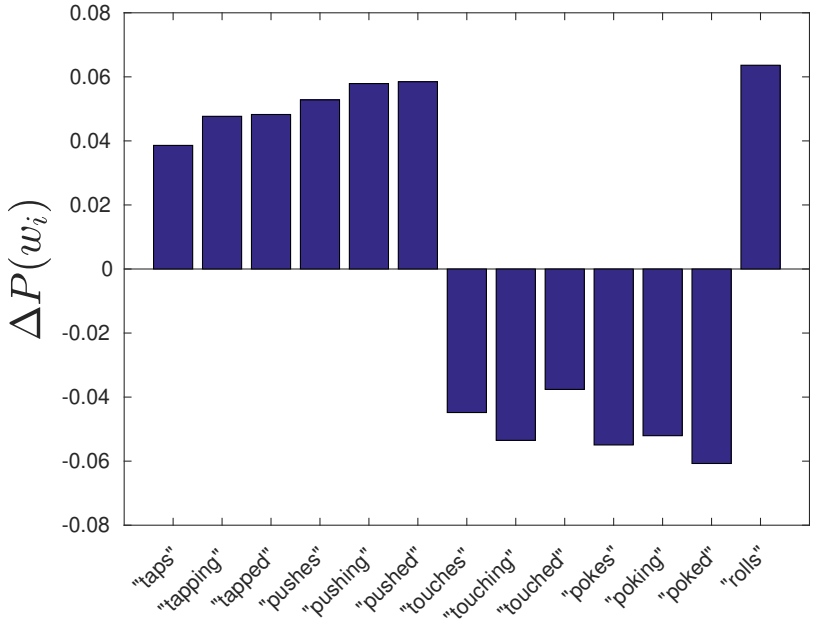


Figure 28: Affordances and verbal language: variation of word occurrence probabilities $\Delta p(w_i) = p_{\text{comb}}(w_i \mid X_{\text{obs}}, \text{Action=tap}) - p_{\text{BN}}(w_i \mid X_{\text{obs}})$, where $X_{\text{obs}} = \{\text{Size=big, Shape=sphere, ObjVel=fast}\}$. This variation corresponds to the difference of word probability when we add the tap action evidence (obtained from gesture recognition) to the initial evidence about object features and effects. We have omitted words for which no significant variation was observed.

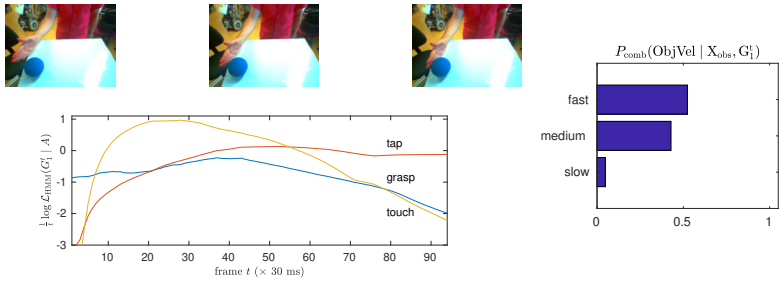
4.4.1.3 *Prediction of Word Probabilities*

Our model permits to make predictions over the word variables associated to affordance evidence (see Table 3, last row). In Fig. 28 we show the variation in word occurrence probabilities between two cases:

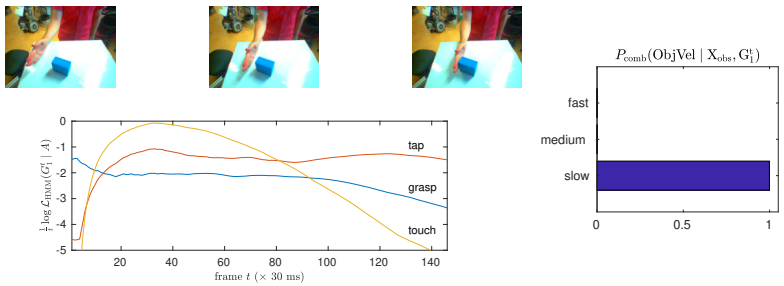
1. when the robot's prior knowledge evidence consists of information about object features and effects only: {Size=big, Shape=sphere, ObjVel=fast};

2. when the evidence corresponds to the one of the previous point, with the addition of the *tap* action observed from the gesture recognizer (deterministic hard evidence).

This result is interesting for two reasons. First, the probabilities of words related to tapping and pushing increase when a tapping action evidence from the gesture recognizer is introduced; conversely, the probabilities of other action words (touching and poking) decreases. Second, the probability of the word "rolling" (which is an effect of an action onto an object) also increases when the tap action evidence is entered.



(a) Action performed on small sphere. Description: "the robot pushed the ball and the ball moves".



(b) Action performed on big box. Description: "the robot is pushing the big square but the box is inert".

Figure 29: Affordances and gestures combined model: object velocity effect anticipation before impact. The evidence from the gesture recognizer (left) is fed into the Affordance–Words model before the end of the execution. The combined model predicts the effect (right) and describes it in words (verbal language).

4.4.1.4 *Effect Anticipation*

Since the gesture recognition method interprets sequences of human motions, we can test this predictive ability of our combined model when we observe an incomplete action. Fig. 29 shows an example of this where we reason about the expected object velocity caused by a tap action.

In particular, Fig. 29a shows the action performed on a spherical object, whereas Fig. 29b on a cubic one. Within each of the two figures, the graphs on the left side show the time evolution of the evidence $p_{\text{HMM}}(A \mid G_1^t)$ from the gesture recognizer. In order to make the variations emerge more clearly, instead of the posterior, we show $\frac{1}{t} \log \mathcal{L}_{\text{HMM}}(G_1^t \mid A)$: the log-likelihood normalized by the length of the sequence.

Note how, in both cases, the correct action is recognized by the model given enough evidence, although the observation sequence is not complete. The right side of the plot shows the prediction of the object velocity, given the incomplete observation of the action and the object properties. The model correctly predicts that the sphere will probably

Table 4: Affordances and verbal language: 10-best list of sentences generated from the evidence X_{obs} = {Color=yellow, Size=big, Shape=sphere, ObjVel=fast}.

sentence	score
"the robot pushed the ball and the ball moves"	−0.54322
"the robot tapped the sphere and the sphere moves"	−0.5605
"he is pushing the sphere and the sphere moves"	−0.57731
"the robot is tapping the yellow ball and the big yellow sphere is moving"	−0.57932
"he pushed the yellow ball and the sphere is rolling"	−0.58853
"the robot is poking the ball and the sphere is rolling"	−0.58998
"he is pushing the ball and the yellow ball moves"	−0.59728
"he pushes the sphere and the ball is moving"	−0.60528
"he is tapping the yellow ball and the ball is moving"	−0.60675
"the robot pokes the sphere and the ball is rolling"	−0.60694

move but the box is unlikely do so. Finally, the captions in the figure also show the human-interpretable verbal description (see Sec. 4.3.3) generated by feeding the probability distribution of the words estimated by the model, given incomplete evidence, into the CFG.

4.4.2 *Verbal Descriptions*

We now present results about the verbal descriptions generated by the model with the CFG. They allow us to observe the emergence of non-trivial language phenomena (they emerge from our developmental model, not from the grammar itself, which is provided only for the purpose of interpreting the probability distributions over the words).

By generating and scoring verbal descriptions about what the robot observes (see Sec. 4.3.3), we can provide evidence to the model and interpret the verbal results.

From Sec. 4.3.3.2 recall that, with our method, we do not add new words to the model when we observe the human performing actions. Rather, the human-readable descriptions that we generate are based on the same words that were present in the self-centered learning phase (see Sec. 4.3.1). In that phase, the verbal descriptions described the agent of the observed actions as either "the robot", "he", or "Baltazar" (the name of the robot used in [Sal+12]). Consequently, the Affordance–Words model learned by the robot includes those words as the subject of the action.

4.4.2.1 *Choice of Synonyms*

As an example, by providing the evidence X_{obs} ={Color=yellow, Size=big, Shape=sphere, ObjVel=fast} to the model, we obtain the sentences reported in Table 4. The higher the score, the more likely the sentence.

In many of the sentences in the table, we note that (i) the correct verb related to the tap action is generated (in the initial evidence, no

sentence	score
"the robot is picking the sphere **and** the sphere is moving"	−0.59328
"the robot grasps the sphere **and** the ball is moving"	−0.59507
"the robot is picking the sphere **and** the sphere is rising"	−0.60882
"the robot grasped the sphere **and** the sphere is rising"	−0.61842
"the robot picked the ball **and** the ball is rising"	−0.64052
"baltazar grasps the sphere **and** the sphere is moving"	−0.66182
"the robot has grasped the ball **and** the ball is rising"	−0.66398
"the robot picked the ball **and** the green ball is moving"	−0.67134
"baltazar grasped the sphere **and** the ball is moving"	−0.67283
"baltazar is grasping the ball **and** the sphere is rising"	−0.6787

(a) Evidence: X_{obs} ={ Action=grasp, ObjVel=medium }.

sentence	score
"the robot is picking the cube **but** the square is still"	−0.52575
"the robot is grasping the sphere **but** the box is inert"	−0.55
"the robot is grasping the square **but** the sphere is still"	−0.55388
"the robot grasped the square **but** the cube is inert"	−0.55608
"baltazar is grasping the square **but** the square is inert"	−0.5571
"the robot is grasping the cube **but** the ball is inert"	−0.56011
"the robot picks the box **but** the square is inert"	−0.56397
"baltazar is picking the square **but** the square is still"	−0.56402
"he is grasping the square **but** the cube is inert"	−0.56815
"the robot grasps the square **but** the sphere is inert"	−0.57417

(b) Evidence: X_{obs} ={ Action=grasp, ObjVel=slow }.

Figure 30: Affordances and verbal language: 10-best list of sentences generated given two different sets of evidence. In (a) the model interprets the object movement as indicating a succesful grasp and uses the conjunction "and". In (b) the slow movement is interpreted as no movement at all and, therefore, as an unsuccessful grasp: for that reason, the conjunction "but" is used.

action information was present, only object features and effects information were), and (ii) the object term "ball" or synonyms thereof (e.g., "sphere") are used coherently, both in the first part of the sentence describing the action and in the second part describing the effect.

This result shows that different synonyms may be used by the model in the same sentence. This is a consequence of the random generation of sentences, described in Sec. 4.3.3.2, and because synonyms are often assigned similar (but not necessarily equal) probabilities by the model, given the same evidence.

4.4.2.2 *Choice of Conjunction*

The manipulation experiments that we consider in this chapter have the following structure, similar to the one described in Sec. 3.2: an agent (human or robot) performs a physical action onto an object with certain properties, and this object will show a certain physical effect as a result. For example, a touch action on an object yields no physical movement, but a tap does (especially if the object is spherical). In the language description associated to an experiment, it makes sense to analyze the *conjunction* chosen by the model given specific evidence. In particular, it would be desirable to separate two kinds of behaviors: one in which the action and effect are coherent (expected conjunction: "and"), and the other one in which they are contradictory ("but").

Fig. 30 shows an example of the behavior described above. We give the same action value *grasp* to the model as evidence, but two different values for the final object velocity. When the object velocity is medium (Fig. 30a), the model interprets this as a successful grasp, and it uses the conjunction "and" to separate the description of the

(a) "the robot is grasping the box and the green box is moving".

(b) "the robot is poking the green square and the cube is inert".

(c) "the robot picked the ball and the green ball is moving".

(d) "baltazar is poking the green sphere and the sphere is still".

Figure 31: Affordances and verbal language: examples of descriptions generated by the model.

action from the description of the effect. When the object velocity is slow (in the clustering procedure, the velocity was most often zero in those cases), the model predicts that this is an unsuccessful grasp and it uses the conjunction "but", instead.

4.4.2.3 *Description of Object Features*

In Fig. 31, we show examples of verbal descriptions generated by the model given different values of observed evidence:

- X_{obs} = {Action=grasp, Color=green1, Shape=box} (Fig. 31a);
- X_{obs} = {Action=touch, Color=green1, Shape=box} (Fig. 31b);
- X_{obs} = {Action=grasp, Color=green2, Shape=sphere} (Fig. 31c);
- X_{obs} = {Action=touch, Color=green2, Shape=sphere} (Fig. 31d).

Note that the box object in the two first examples has a dark shade of green (value of Color affordance variable of Table 3 clustered as: green1), whereas the spherical one in the two last examples has a lighter shade (Color value: green2). However, the verbal descriptions reported in Fig. 31 all use the adjective "green". This behavior emerges from fact that the robot develops its perceptual symbols (clusters) in an early phase, and only subsequently associates them with the human vocabulary. We believe that this phenomenon is practical and potentially

useful (i.e., the possibility that a low-level fine-grained robot representation can be abstracted into a high-level language description, which bundles the two shades of green under the same word).

4.5 CONCLUSIONS AND FUTURE WORK

This chapter has illustrated a computational model that combines object affordances, human gestures and verbal language. We presented such a combined model, allowing a robot to interpret and describe the actions of external agents, by reusing the knowledge previously acquired in an ego-centric manner.

We have shown that for cognitive robots it is possible, and indeed fruitful, to combine knowledge acquired from interacting with elements of the environment (affordances) with the probabilistic observation of another agent's actions (gestures) as well as verbal language elements. In this sense, our model supports the growing field of *human–robot collaboration* [BWB08; Dra+15], whose goal is to enable effective teamwork between humans and robots.

In a developmental setting, the robot learns the link between words and object affordances by exploring its environment. Then, it classifies the manipulative gestures performed by another agent. Finally, by fusing the information from the affordances model and a gesture recognizer, the robot can reason over affordances and words when observing the other agent. This can also be leveraged to do early action recognition (see Sec. 4.4.1.4).

In terms of language, although the complete model only estimates probabilities of single words given the evidence, we showed that feeding these probabilities into a pre-defined grammar produces human-interpretable sentences that correctly describe the situation. We also highlighted some interesting language-related properties of the combined model, such as: the choice of relevant words to describe a scene, the choice of synonyms, and of congruent/incongruent conjunctions,

Our demonstrations are based on a restricted scenario (see Sec. 4.2.1), i.e., one human and one robot manipulating simple objects on a shared table, a pre-defined number of motor actions and effects, and a vocabulary of approximately 50 words to describe the experiments verbally. However, one of the main strengths of our study is that it spans different fields such as robot learning, language grounding, and object affordances. We also work with real robotic data, as opposed to learning images-to-text mappings (as in many works in computer vision) or using robot simulations (as in many works in robotics).

In terms of *scalability*, note that our Bayesian Network (BN) model can learn both the dependency structure and the parameters of the model from observations. The method that estimates the dependency structure, in particular, is sensitive to biases in the data. Consequently, in order to avoid misconceptions, the robot needs to explore any possi-

ble situation that may occur. For example, if the robot only observes blue spheres rolling and the objects with any other shape are never blue, it might infer that it is the color that makes the object roll, rather than its shape. In order to scale the method to a larger number of concepts, it would be necessary to scale the amount of data considerably, similarly to what is done in many deep learning approaches. In models of developmental robotics, where this is neither practically feasible, nor desirable, we would need to devise methods that can generalize more efficiently from very few observations.

As future work, it would be useful to investigate how the model can extract syntactic information from the observed data autonomously, thus relaxing the bag-of-words assumption in the current model. Another line of research would be to study how the model can guide the discovery of new acoustic patterns (e.g., [FS17; VS14; VS12]), and how to incorporate the newly discovered symbols into our Affordance–Words model. This would release our current assumption of a pre-defined set of words.

5

TOOL USE AFFORDANCES

In this chapter, we present a computational model of affordances capable of dealing with multiple objects (see Fig. 32), giving rise to a *tool use* behavior. This skill is useful to operate in complex manipulation tasks typical of human-like environments.

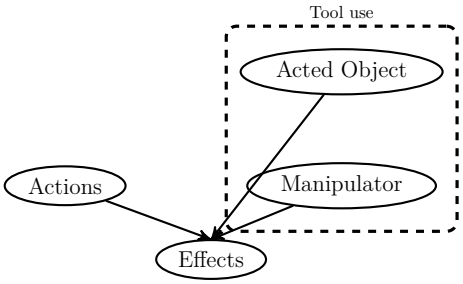


Figure 32: Computational model of affordances for dealing with multiple objects and tool use. In this chapter, the manipulator corresponds to the held object (i.e., tool) or to the robot hand.

The concept of tool use has been studied extensively in cognition research of humans and other animals [Bec80; SH08]. Beck defines it as "[...] the external employment of an unattached environmental object to alter more efficiently the form, position, or condition of another object, another organism, or the user itself when the user holds or carries the tool during or just prior to use and is responsible for the proper and effective orientation of the tool" [Bec80, p. 10].

We accomplish a tool use behavior with our computational model through reasoning, learning and a developmental approach.

In terms of reasoning, we have the robot interpret the possibilities offered by *multiple entities* in a joint way. In particular, the robot interprets the possibilities permitted by a *manipulator* onto an *acted object*. By manipulator, we refer to a first, grasped object which acts as a tool (also known as held or intermediate object), being located in the agent's hand; we also use the term manipulator to indicate the bare hands of the robot with a particular aperture of the fingers, when we study the transition from hand to tool affordances. By acted (or primary) object, we refer to the target object in the scene over which the action is exerted. A key aspect of our system is that it analyzes the relationship between *sub-parts* of objects: not only looking at the level of their entirety, but also at their constituent sub-parts, such as the handle part or the effector/tip part of objects.

For *learning* tool affordances, we evaluate different computational models, in particular different BN structures and parameters, for evaluating the capability of predicting effects from previously unseen data (generalization), and the possibility of transfering predictions from robot simulation to the real world.

Finally, we propose a method for learning the affordances of different robotic hand postures, investigating the *developmental link* from hand affordances (i.e., action possibilities by using the hands) to tool affordances (action possibilities by using tools).

This chapter is the subject of the following publications:

- Afonso Gonçalves, Giovanni Saponaro, Lorenzo Jamone, and Alexandre Bernardino. "Learning Visual Affordances of Objects and Tools through Autonomous Robot Exploration". In: *IEEE International Conference on Autonomous Robot Systems and Competitions*. 2014, pp. 128–133. DOI: 10.1109/ICARSC.2014.6849774.

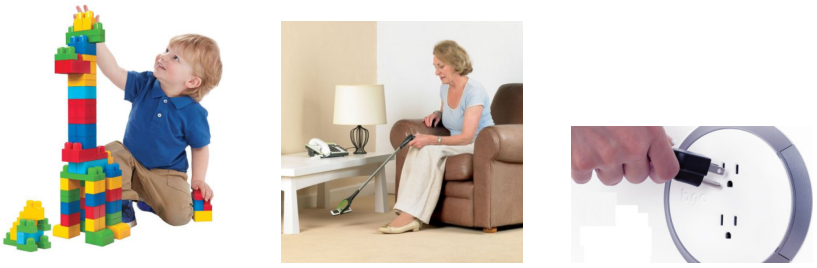
- Afonso Gonçalves, João Abrantes, Giovanni Saponaro, Lorenzo Jamone, and Alexandre Bernardino. "Learning Intermediate Object Affordances: Towards the Development of a Tool Concept". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2014, pp. 482–488. DOI: 10.1109/DEVLRN.2014.6983027.

- Giovanni Saponaro, Pedro Vicente, Atabak Dehban, Lorenzo Jamone, Alexandre Bernardino, and José Santos-Victor. "Learning at the Ends: From Hand to Tool Affordances in Humanoid Robots". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017, pp. 331–337. DOI: 10.1109/DEVLRN.2017.8329826.

The outline of this chapter is as follows. Sec. 5.1 gives motivations for developing a tool use behavior in robots; Sec. 5.2 lists related work from the literature; Sec. 5.3 presents our proposed approach. In Sec. 5.4 we report the experimental results, and finally in Sec. 5.5 we draw our conclusions and possible future extensions.

5.1 MOTIVATION

Many important human behaviors require putting (multiple) objects in such a way that they are in physical contact with each other. A fundamental cognitive ability to master such skill is to understand the relationships between the physical properties of the objects' surfaces that enter into contact, i.e., *inter–object* affordances or mutual affordances among objects. For instance, to pile objects we must put into contact their flat surfaces to assure stability (see Fig. 33a); to bring objects that are out of reach from our arms closer to us, we pull them with



(a) Making a pile. (b) Reaching for faraway objects. (c) Fitting.

Figure 33: Examples of human behaviors that involve multiple objects in relationship with each other.

elongated objects (see Fig. 33b); to fit objects together we match concave parts on one object to corresponding convex parts on the other (see Fig. 33c).

In humans, the first tools are one's own hands. Learning the affordances of the hands (i.e., what actions one can do with them, what effects one can obtain) is a long developmental process that begins in infancy [Gib94; Jam10]. At 6 months of age, infants already manipulate objects in a differentiated manner depending on the object's properties [BB93]. The process continues during childhood through exploration of different actions and different objects [Ros77; Ros09]. In essence, children achieve inter–object and functional tool use reasoning abilities over several stages [Loc00; SD10; LG13; FRO14]. The knowledge previously acquired by babies during manual exploration of objects is likely to play a role in tool use. Definitely, one of these roles is that the increased hand dexterity acquired during development allows the child to correctly grasp, manipulate and orient a tool; however, another role may be that the child "sees" in the shapes of some tools relevant characteristics that remind the child of previously used shapes of the own hands (although no experimental evidence of this perceptual skill has been provided in the developmental psychology literature, as far as we know).

Typically, a manipulative robot operates on external objects by using its own hands (or similar end-effectors), but in some cases the use of tools may be desirable. For instance, if the robot has to use certain objects which are not reachable (due to geometric workspace constraints), tools may be a convenient way to extend the length of robot limbs, thus permitting the robot to reach for far objects. The advantage of modeling inter–object affordances (i.e., affordances among multiple objects) is that this permits to infer (i) affordances of acted objects, (ii) affordances of manipulator (held) objects, and (iii) affordances of the interaction between held and acted objects. Our model can be used to predict effects given both objects and the performed action (i.e., effect prediction), or choose the best manipulator object or tool to achieve a

goal (i.e., tool selection), or choose the best action given the available objects and a desired effect (i.e., action selection, which is particularly useful for planning complex actions made up of many simple steps, as we will see in Ch. 6). In general, the evaluation tasks that we devise aim to test the capability of predicting effects from previously unseen data (generalization), and the possibility of transfering predictions from robot simulation to the real world.

Inspired by the above observations in developmental psychology, and motivated by a need of autonomous robotic systems, we investigate the following aspects related to tool use on robots:

- we design a *reasoning model* of visual inter–object affordances, namely a model that deals with the relationships between (i) pairs of objects, (ii) sub-parts of said objects;
- we devise a method to *learn* inter–object affordances, by designing and evaluating variations of the above model, both specified *a priori* or automatically obtained from experimental data via Structure Learning (see Sec. 2.1.3);
- we make a robot learn the affordances of different hand postures and, having done that, we investigate the *developmental link* from hand affordances (i.e., action possibilities by using the hands) to tool affordances (action possibilities by using tools).

About the model, we specialize our system for visual robot affordances (introduced in Sec. 3.3) so that it is able to process multiple simultaneous objects, their mutual relationships, and the relationships between object sub-parts (e.g., handle part and effector part).

In terms of learning, we compare different Bayesian Network (BN) structures and parameters, and we evaluate them for the tasks of (i) predicting effects from previously unseen data (generalization), and (ii) the possibility of transfering predictions from robot simulation to the real world.

About the link from hands to tools, we explore how a learned representation of hand affordances can be generalized to estimate the affordances of tools which were never seen before by the robot.

5.2 RELATED WORK

This section overviews related work about hand and tool affordances in the contexts of psychology and robotics.

5.2.1 *Psychology*

In developmental psychology, it is still debated whether the skill of tool use emerges progressively through familiarization with experience, or it appears through sudden insight at a certain age.

The skill of tool use has been observed in greater apes for almost a century [Köh17]. In humans and in more recent times, Fagard reports a longitudinal study on five infants aged 12 to 20 months, where they have to use a rake-like tool to reach toys that are out of reach [FRO14]. Their results indicate that it is only between 16 and 20 months that the infants *suddenly* start to intentionally try to bring the toy closer with the tool. According to this research, the sudden success at about 18 months might correspond to the coming together of a variety of capacities, such as the development of means–end behavior (i.e., they notice and recall cause and effect actions and reactions).

In terms of the connection from hand affordances to tool affordances, several researchers have investigated the role of hand actions during human intelligence development for learning to deal with the uncertainty of the real world (e.g., toddler visual attention [Yu+09]) and tool use. Piaget documents an observation where his daughter makes an analogy between a doll's foot hooking her dress, and her own finger bent like a hook [Pia62]. Tool use competence in humans emerges from *explorative actions*, such as those performed with the child's bare hands in the first year [Smi+14].

Lockman [Loc00] suggests that the actions employed by toddlers on a daily basis initially incorporate many of the (previously learned) motor patterns that infants employ with their hands and arms for exploring and learning their everyday objects. Szokolszky [SD10] stresses how tool use is dependent and continuous with other action routines, such as reaching, grasping, focusing on an object or on a person, and eating with the hand.

In [LG13], Lobo highlights the following points about the relationship between early self-exploration behaviors and developing object exploration behaviors: (i) in the first months of life, infants are already actively engaging in exploratory behaviors to inform themselves about the affordances of their own bodies, objects, and the intersection of the two; (ii) the emergence of reaching is an important step forward towards advanced object exploration and advanced self-exploration; (iii) the behaviors that infants adopt to explore their own bodies and surfaces during the first months of life may form the set of behaviors from which they later *choose*, as they begin to interact with objects.

With these considerations in mind, one of the things that we pursue in this chapter is a robotic model that transfers hand knowledge to tool knowledge. In the experimental part, we verify the applicability of a *tool selection* problem when the robot is presented with unknown tools, given its previous exploratory hand knowledge.

5.2.2 *Robotics*

In robotics, many computational models have been proposed to express affordances and tool use [WHS05; Sto05; SS07; Sto08; Tik+13;

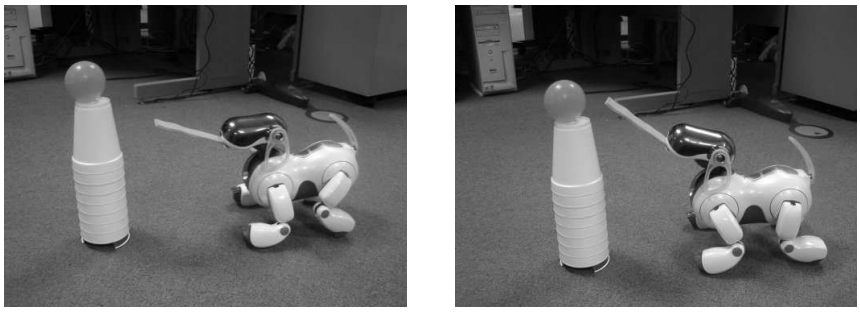


Figure 34: Sequence of frames of a robot using a stick tool, reproduced from [WHS05].

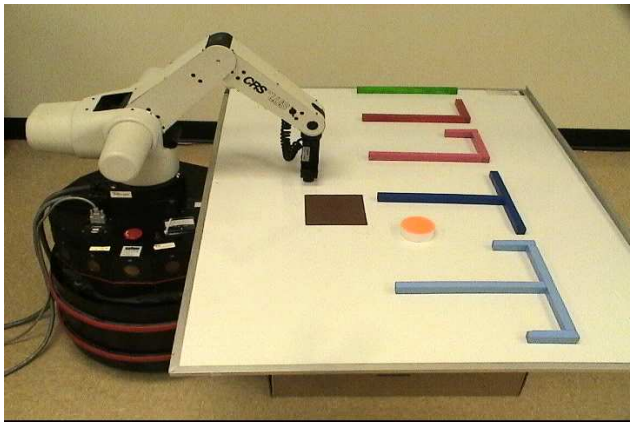


Figure 35: A robot arm with different tools and an orange target object, reproduced from [Sto08].

MTN18; JI13; AG17; Mol+18]. The objective of these works is to implement complex problem solving abilities in autonomous robots [Jam+16]. What they have in common is that they give a robot model the possibility of dealing with multiple objects, in other words, of reasoning about inter–object affordances.

One of the first examples of a robot computational model designed to acquire tool use capabilities is [WHS05]. In this work, a Sony Aibo dog-like robot is equipped with an artificial neural network to learn appropriate postures for grasping a stick tool and thus for reaching a faraway ball on a tower, as shown in Fig. 34. Implicitly, it is relying on an internal representation of its body (a body schema) with the attached tool: we will elaborate on this concept for visually processing images of robot hands in Sec. 5.3.3.

The works by Stoytchev and Sinapov [Sto05; SS07; Sto08] propose to learn tool affordances as tool–behavior pairs which yield a desired effect on a robot manipulator, shown in Fig. 35. Using a number of possible tools (designed to be similar to the ones used by [Köh17] in experiments with chimpanzees), the robot arm explores different possible behaviors and observes their effects on the environment. We should note that

these models learn the affordances of specific tools (i.e., considered as individual entities), however no association between the distinctive features of a tool and its affordances is made. Therefore, the generalization capabilities of these models are limited to dealing with smaller and larger versions of known tools.

Tikhanoff [Tik+13] focuses on an iCub robot learning the specific tool affordance of pulling. This is done by learning a relationship between angles of the robot action being exerted, and distance traveled by objects on the table, after a series of pull actions. Although useful for robot operations, this knowledge is specific for the tool that is experienced at training time, and it cannot be easily generalized to novel, previously unseen, tools. This limitation is relaxed by Mar [MTN18]: visual features are extracted from the functional part of the tool (also accounting for the way in which the tool is grasped), and they are related to the effects observed after the robot action. This allows to robustly adjust the motion parameters depending on how the tool is grasped by the robot. However, the target object's shape is not taken into consideration and, as such, the influence of the object on the measured effects is not studied. In addition, that system starts with the tool in the robot's hand: therefore, it does not address tool selection.

In [JI13], a Bayesian Network (BN) is used to model tool affordances as probabilistic dependencies between actions, tools and effects. To address the problem of predicting the effects of unknown tools (i.e., the generalization ability of the model), the authors of that work propose a tool representation based on the functional features of the tool (geometrical features, e.g., corners, bars, etc.), arguing that those features can remain distinctive and invariant across different tools used for performing similar tasks. However, it is not clear how those features are computed or estimated, if they can be directly obtained through robot vision and if they can be applied to different classes of tools. Also, the functional features in that system have to be annotated by hand, contrary to other works such as [MTN18].

It is worth noting that in [Sto05; SS07; Sto08; Tik+13; MTN18; JI13] the properties of the acted objects are not explicitly considered in the model; only the general affordances of tools are learned, regardless of the objects that the tools act upon. Instead, in our model we relate the properties of the acted objects with the properties of the tools.

Abelha and Guerin [AG17] propose a system that, given a specified task and some available candidate tools in a scene, learns to predict the individual tool affordances (the results are in the form of pixelwise scores, as well as the regions for grasping and using tools). Prior task knowledge is learned from simulating actions with 116 object CAD models available from the web. One strength of this system is that, in addition to predicting how well a tool part affords a task, it also provides geometric manipulation cues (indicating the region for grasping the tool and the region for using it onto the target object), thus explor-

ing the idea that a tool potentially possesses many ways in which it can be used for a given task. However, this work is done in simulation only: not only is it not evaluated on a real robot, but even these simulations are not embodied in any specific robot. Because of this limitation and of the differences between all the end effectors that exist in real robots, the applicability of such a work on real robots remains to be seen.

Moldovan and colleagues consider a multi-object scenario [Mol+18] in which the relational affordances between objects pairs are exploited to *plan* a sequence of actions to achieve a desired goal, using probabilistic reasoning (we will elaborate on the usefulness of affordances for planning in Ch. 6). The pairwise interactions are described in terms of the objects' relative distance, orientation and contact. However, the authors of that work do not investigate how these interactions are affected by different geometrical properties of the objects.

In Sec. 5.2.1 we analyzed the possible link from hand affordances to tool affordances. To the best of our knowledge, ours is the first contribution, in the robot affordances field, which explicitly looks at the visuomotor possibilities offered by different hand morphologies and postures (e.g., hands with straight fingers, bent fingers, or arched fingers, see Fig. 42). We exploit this information to acquire, through self-exploration, a model that is able to generalize to novel situations for a robotic agent, including making the crucial developmental leap from hand use to tool use, as observed in babies by psychology studies.

5.3 PROPOSED APPROACH

Fig. 32 shows a diagram of our computational model of affordances for dealing with multiple objects and, thus, permitting tool use behavior. Our model is an extension of the model by Montesano (see [Mon+08] and Sec. 2.2.1), which had the limitation of the robotic agent dealing with one object only. By contrast, our extension permits the agent to consider a pair of entities: a manipulator (i.e., the held object in the robot's hand, or the bare hand itself) and an acted object upon which actions are executed.

Below, we describe our proposed approach as follows. Sec. 5.3.1 illustrates the reasoning model of inter–object affordances. In Sec. 5.3.2 we show how to learn affordances of multiple objects and tools, including the transfer of knowledge from simulation to a real robot. Finally, in Sec. 5.3.3 we explore the developmental link from hand affordances to tool affordances.

Notably, in Sec. 5.3.2 ([Gon+14a]) the concept of toolness is not specified in the model, but it emerges from experiments. Instead, in Sec. 5.3.3 ([Sap+17b]) the concept of toolness is a starting hypothesis, and the focus is on the developmental transition from hand to tool affordances.

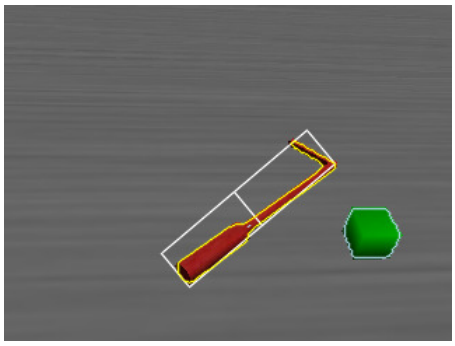


Figure 36: Two environment objects being visually processed in simulation.
Left: a possible manipulator (held object or tool), whose segmented silhouette is divided in two parts (top and bottom) along its main axis.
Right: possible acted object. This one is also divided in two parts by the model, however we do not show the halves graphically when the whole object's compactness descriptor is above an empirically-defined threshold (which only affects the display, not the experiments).
From each object and each object part we compute a set of shape descriptors, used to build the tool affordances knowledge. Compare with Fig. 21 on p. 44 for the one-object case.

5.3.1 *Computational Model*

The computational formulation of affordances by Montesano [Mon+08] models affordances as action–object–effect triplets (see Sec. 2.2.1). Due to this formulation, only certain robot scenarios can be considered: those where the action is applied to a *single object* using the robot hands and the effects are observed. In this section, we extend that formulation by explicitly modeling both the manipulator (e.g., the held object) and the acted object (i.e., the target of the action) with corresponding variables[1], thus reasoning about *inter–object* affordances. We do this by modifying the visual affordance and reasoning framework of Sec. 3.3.

We now illustrate our model to capture meaningful relationships between actions, effects, manipulators and acted objects.

Both the manipulator and the acted object are represented by the pre-categorical shape descriptors (described in Sec. 3.3.1) of visually segmented objects ("blobs") seen by the robot. The shape descriptors

[1]For now, we consider the manipulator node of Fig. 32 to refer to the held object (i.e., tool). Later on in the chapter, we will consider the special case when this manipulator is the robot's bare hand.

employed here are a subset of the ones reported in Table 2 on p. 45: convexity, eccentricity, compactness, circularity, squareness.

The most distinctive aspect of our model with respect to the state of the art is that we consider elongated objects split in two halves along its main axis, as shown in the example of Fig. 36.

The intuition for reasoning about sub-parts (halves) of tools is that the affordance offered by a tool often resides in only the most salient and functional part of the tool perceived by the agent [Loc00], not in the entirety of it. A hammer tool affords the action of hitting a nail so that it enters a wall, and this capability resides in the characteristics of the tip of the hammer (e.g., the shape and the material of the tip).

For reasoning on tool affordances with robot perception algorithms, the graphical splitting of a tool along its main axis is a simple yet helpful way to capture affordances of manipulator (held) objects, for which only the effector part (tip part, or non-grasped part) physically interacts with the acted object. Note that, when the robot sees a possible manipulator object lying on the table, in our model any of the two halves could be potentially used as effector: *we do not pre-program which of the halves is the handle and which is the effector*, but we let the robot discover that by *autonomous exploration*, following the developmental robotics perspective described in Sec. 1.2.

For manipulators (held objects), the Bayesian Network variables that we consider are the visual descriptors of one of the object halves: during learning, the half that is not grasped is considered; during inference (e.g., effect prediction, tool selection, action selection), each of the halves can be considered, if the held object has not been grasped yet. For acted objects, the whole blob visual descriptors are used (i.e., the blob is not split in two halves). As in p. 45, each shape descriptor has a value that can range within three empirically-defined discrete levels.

In terms of motor *actions* performed by the agent with a manipulator (held) object in its hand onto an acted object on the table, we consider the following pre-defined directional movements: left lateral tap, right lateral tap, pull closer, push away.

The discrete identifiers of these four actions are the values of the action node in the affordance Bayesian Networks (BNs).

Finally, as for the resulting *effects*, we do as in p. 42 in terms of modeling the two directions of displacement (lateral and longitudinal) of the acted object on a tabletop, each direction having a value that can range within five empirically-defined discrete levels of possible displacement magnitude. Fig. 37 shows an illustration, with the two effects marked E_x and E_y.

Our model allows to predict the motion effects induced by the motor action being exerted by the robot and the shape descriptors of the two objects involved: $p(\text{EffectX}, \text{EffectY} \mid A, T, O)$, where EffectX is the lateral motion of the acted object on the table, EffectY is its lon-

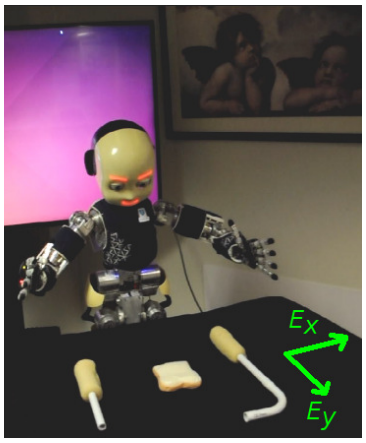


Figure 37: The iCub robot in a tool use scenario, with green overlay arrows showing the effects (i.e., displacements of objects).

gitudinal motion, A is an identifier of the robot action (e.g., pushing, pulling), T is the vector of shape descriptors of the manipulator (tool) held in the robot's hand, and O is the vector of shape descriptors of the acted object on the table.

5.3.2 *Learning*

In this section, we show how to learn from data our computational model for tool use affordances described in Sec. 5.3.1. We compare different Bayesian Networks that implement our model, to determine the most suitable one. The comparison is in terms of memory complexity, prediction ability and generalization capability. In all the Bayesian Network structures that we discuss, we use discrete variables and Maximum A Posteriori (MAP) probability estimates to learn the Conditional Probability Distribution (CPD) table parameters.

To begin with, we capture these relationships *in simulation*, which has the advantage of letting us run hundreds of robot manipulation experiments without the cost associated to real robot experiments. Later, we will see how it is possible to transfer tool use affordances knowledge from a simulated robot to a real robot, making predictions in the real world, and take optimal decisions with respect to a desired outcome during manipulation.

The first, baseline structure for our comparisons is a manually defined *fully connected* network, shown in Fig. 38. This is the most general structure, in which all the acted object, manipulator (held object) and action nodes are connected to the effect nodes.

The fully connected network suffers from a number of limitations: low performance, overfitting, large number of parameters. Basically, this network structure suffers from the curse of dimensionality[2]. Each effect

[2]The curse of dimensionality [Bis07, p. 33] is a difficulty arising when we add more features to a pattern recognition model (i.e., when we increase the dimension-

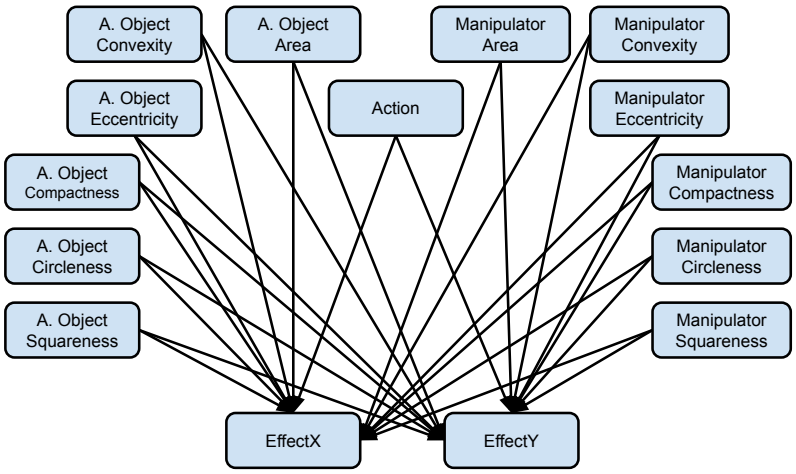


Figure 38: Fully connected Bayesian Network structure to encode inter–object affordances. A. Object means Acted Object, Manipulator refers to the held object. This structure was specified manually.

node has a high number of parents. In our case 6 for the manipulator (held object), plus 6 for the acted object, plus 1 for the action, in total 13: this results in big CPD tables, which makes the network hard to train and unfit to generalize the trained observations to unseen situations.

The second structure is a dimensionality-reduced one. To reduce the dimensionality of the feature space, we apply Principal Component Analysis (PCA)[3] to the features seen in our training data, as shown in the upper part of Fig. 39 in white. We use 80% of our experimental data for training and 20% for testing, where the original feature space has 12 dimensions: 6 features for the manipulator (held object) and 6 for the acted object, considered jointly.

PCA provides the 12 eigenvectors and eigenvalues computed from the data matrix, of which, however, we only need the 2 Principal Components with the highest significance (i.e., the 2 with the highest eigenvalue), to explain over 99% of the data variance. This shows that acted object and manipulator features are highly correlated. Therefore, we create two nodes, each corresponding to a Principal Component, and these, along with the action node, are now the parents of the effect nodes of a *reduced* Bayesian Network, displayed in the lower part of Fig. 39 in blue. The values of these nodes are the coefficients of each eigenvector given the observable features. These coefficients are then

ality of its feature space), which in turn requires to collect more data. The amount of data that we need to collect to avoid overfitting grows exponentially as we add more dimensions.

[3]PCA is a technique for dimensionality reduction and feature extraction. It attempts to find a (linear) sub-space of lower dimensionality than the original input space, where the new features have the largest variance [Bis07, p. 561].

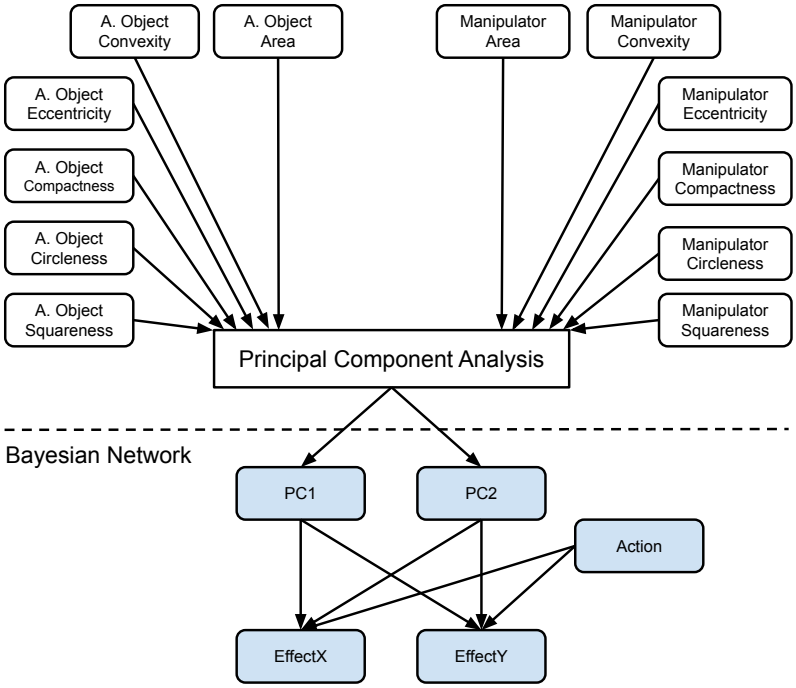
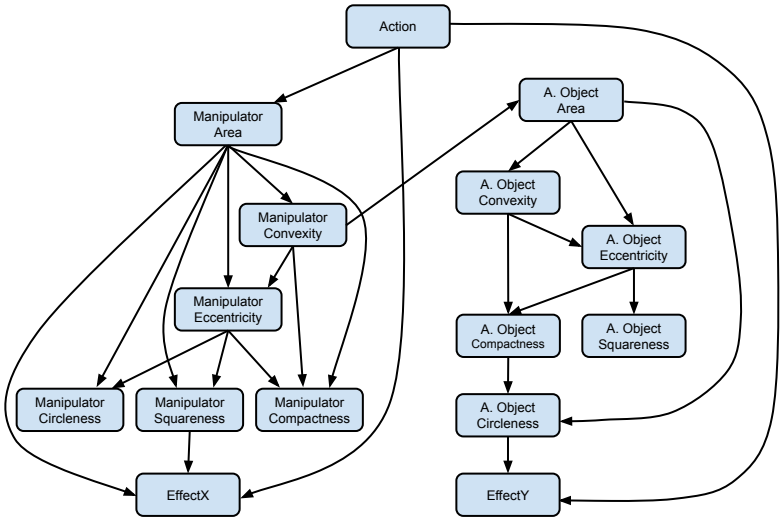


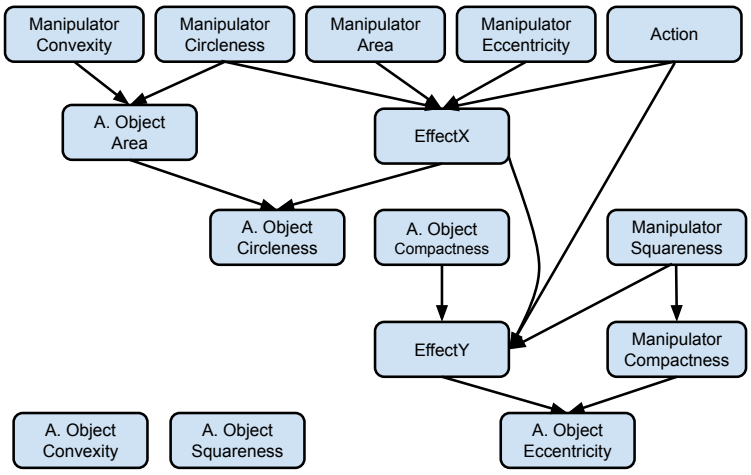
Figure 39: Dimensionality-reduced Bayesian Network structure to encode inter–object affordances, adapted from [Gon+14a]. A. Object means Acted Object, Manipulator refers to the held object. This structure was specified manually, after having empirically tried different PCA hyper-parameters, see Table 5. The PCA dimensionality reduction is computed on the continuous vectors of the visual features.
In this case, there is one PCA block for the whole original visual feature space with 12 dimensions (6 features for the manipulator or held object, and 6 for the acted object, considered jointly). This is because the concept of toolness was not specified in this work, but was emerging from experiments.

Table 5: Hyper-parameters used to train the Bayesian Network of Fig. 39 for predicting the distribution of the effects.

parameter	value (and comment)
number of PCA blocks	1 (manipulator and acted object features considered jointly)
number of components in the PCA	2
number of discretization values (bins) of each PCA component	3
number of discretization values (bins) of each Effect node	5
intervals (in meters) of the Effect bins	$]-\infty, -0.06]$, $]-0.06, -0.025]$, $]-0.025, 0.025]$, $]0.025, 0.06]$, $]0.06, \infty[$



(a) K2 Structure Learning network.



(b) BDe Structure Learning network.

Figure 40: Structure Learning Bayesian Network structures to encode inter–object affordances. A. Object means Acted Object, Manipulator refers to the held object. These structures were obtained from experimental data.

Table 6: Complexity of affordances Bayesian Networks, computed as the sum of the elements in the CPDs of all nodes.

Baseline	PCA	Structure Learning BDe	Structure Learning K2
21 257 680	**168**	1594	535

discretized, based on the training data, into two bins (half the data to each bin). The PCA dimensionality reduction is computed on the continuous vectors of the visual features. This structure was specified manually, after having empirically tried different PCA hyper-parameters. We tried to discretize each node into more bins, but the performance of the network when predicting effects of unseen data got significantly worse.

In Sec. 2.1.3 we have introduced Bayesian Network Structure Learning, that is, the problem of learning the structure of the Directed Acyclic Graph (DAG) from data, and two common heuristic-based approaches for this problem: K2 and BDe. We employ these two approaches and compare the performance of the resulting networks (Figs. 40a and 40b, respectively) with those of the fully connected network and of the PCA one.

We use 80% of our experimental data for training and 20% for testing. All the nodes except for EffectX and EffectY are entered as interventional variables, defined on p. 26, which means that we force a node to take a specific value, thereby effectively severing its incoming arcs [Mur12].

The effect prediction inference performed on the networks is of the type $p(\text{Effect} \mid \text{parents}(\text{Effect}))$, which, considering for example the topology of the network from Fig. 39, amounts to this marginalization over our two effect nodes (horizontal and vertical displacement of the object):

$$p(\text{EffectX}, \text{EffectY} \mid M, O, A), \tag{20}$$

where M is the vector of features of the manipulator, O is the vector of features of the acted object, A is the motor action identifier.

The measure of *complexity* in Table 6 is computed as the number of elements in the largest CPD of a network. Complexity depends only on the discretization and on the network structure, independently of data and learning.

5.3.3 *Hand to Tool Transition*

We now show how the model presented in the previous sections can be adapted for learning the affordances of different hand postures (e.g., hands with straight fingers, bent fingers, or arched fingers, see Fig. 42).

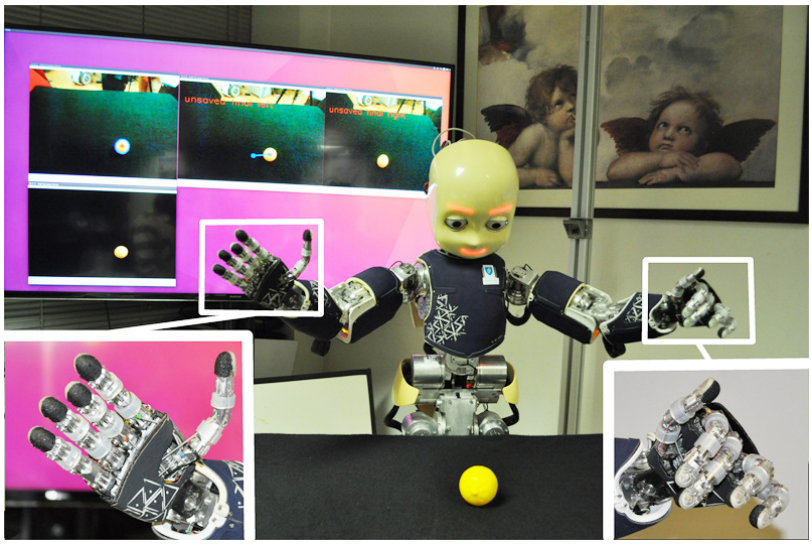


Figure 41: The iCub humanoid robot performing motor actions with different hand postures onto a physical object. In the background screen, we show the visual routines that monitor the evolution of the environment.

This approach is useful for implementing the developmental leap from hand use to tool use on a humanoid robot.

One important motivation to study this problem is the *cost of data acquisition*. While a robot can collect many sensorimotor experiences using its own hands, this cannot happen for all possible human-made tools. Therefore, we investigate the developmental transition from hand to tool affordances: what sensorimotor skills that a robot has acquired with its bare hands can be employed for tool use?

In particular, we adapt the model described earlier in this chapter to relate the (i) visual features of the agent's *own hands* (i.e., in this case the manipulator node of Fig. 32 refers to the hand, not to the held object), (ii) visual features of an acted object located on a surface, (iii) a motor action, and (iv) the resulting effects of the action onto the object, in the sense of the physical displacement compared to the initial position. We use three different robot hand postures, shown in Fig. 42.

The setup is similar to the one described in Sec. 5.3.1, with the following differences:

- motor control: the four directional actions performed with the robot's bare hands are: tapping an object from the left side (with the palm of the hand), tapping an object from the right side (with the back of the hand), pushing an object away from the agent, and pulling the object towards the agent;

- robot actions: the location of the acted object (i.e., the location where the robot performs an action) can be anywhere on the ta-

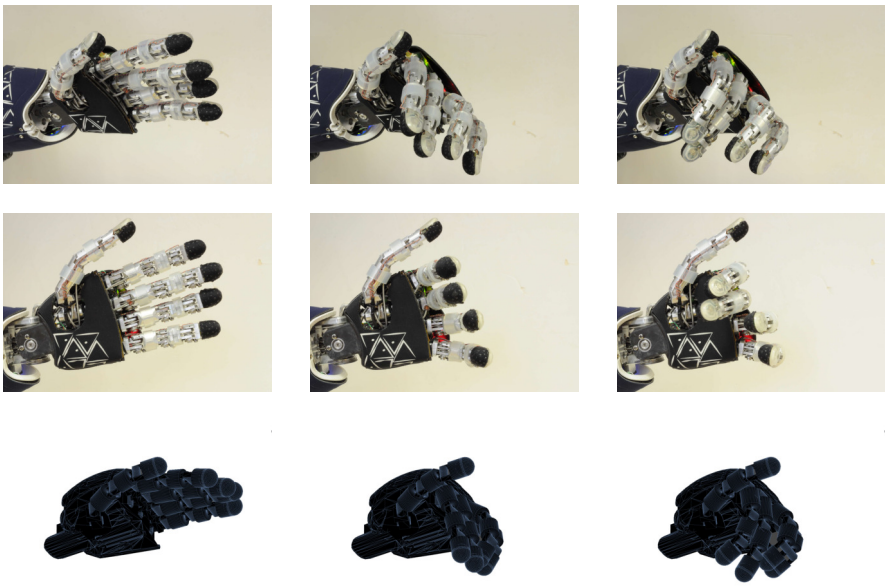


Figure 42: The three robot hand postures adopted to study the hand to tool transition. Left column: straight hand; center column: bent hand; right column: arched hand. The first two rows are real robot postures seen from different viewpoints; the last row shows the simulated body schema CAD model of the top viewpoint. From the latter simulated view, we obtain the segmented silhouette contour of the hand and its shape features.

ble, provided that it is within the reachable space of the robot end-effector, and that it satisfies geometric safety limits to avoid self-collisions. We determine this location with the visual segmentation routines described in Sec. 3.3.1;

- visual features: we incorporate a richer set of 13 features instead of 5 as was the case in Sec. 5.3.1. In addition to convexity, eccentricity, compactness, circularity, and squareness, we also compute features useful to characterize hand shapes: number of convexity defects (i.e., number of cavities along the contour, for example the "holes" between fingers in a hand image), and seven central normalized moments. See also Sec. 2. The raw visual features of manipulators and objects are real-valued and normalized between 0 and 1.

In our endeavor, we wish to compute the visual shape features of the robot's bare hands (for relating them with the other variables in the model), in the various hand postures, but this poses a technical challenge. The image of a robot hand is difficult to segment from the background, and additionaly its contour is not easy to extract, given the different colors of the metal and plastic parts (see for example the top-left hand image in Fig. 42). We bypass this problem by resorting to an internal model of the robot's hand, based on the ideas of *body awareness*.

From a developmental psychology perspective, body awareness appears to be an incremental learning process that starts in early infancy [Hof04] or probably even prenatally [Jos00]. Such awareness is supported by a neural representation of the body that is constantly updated with multimodal sensorimotor information acquired during motor experience and that can be used to infer the limbs' position in space and guide motor behaviors: a *body schema* [BA97].

We use an internal model simulator from [VJB16]. From a technical perspective, using the simulated robot rather than the real one to obtain the hand posture visual shape, serves to filter out noise from the image processing pipeline. Although it is not always true that we can generalize from simulation to the real robots, in this case, we adopt a graphically and geometrically precise appearance model of the robotic hand (based on the CAD model), therefore we can use the internal model simulation without losing generality or compromising the overall idea, as shown when visually comparing the real and simulated hands of Fig. 42.

In terms of hand affordance *learning*, the structure of the BN is the PCA one as described in the previous sections, this time putting in relationships robot manipulator (which can be the hand or the tool), acted object, motor action, and resulting effects. Fig. 43 shows the structure of the BN that we train with robot self-exploration hand affordance data, using the hand postures of Fig. 42. This structure is similar to the one that gave us the best effect prediction performance

in Sec. 5.3.2. In Table 7 we list the hyper-parameters used at training time. Thanks to the dimensionality reduction of this type of network, the number of edges and the computational complexity are also reduced. Most importantly, this type of network reduces the amount of training data required to observe the emergence of some learning effect.

The nodes of the network are the same ones of Sec. 5.3.2, but the PCA block is different due to the following reason.

There are now two PCA blocks: one for the manipulator visual features, one for the affected object visual features. This is because in this work ([Sap+17b]) the concept of toolness is a starting hypothesis, and the focus is on the developmental transition from hand to tool affordances. By contrast, in Sec. 5.3.2 ([Gon+14a]) there was one PCA block for the whole original visual feature space with 12 dimensions (6 features for the manipulator or held object, and 6 for the acted object, considered jointly). That was because the concept of toolness was not specified in that work, but was emerging from experiments.

5.4 EXPERIMENTAL RESULTS

We now show the results obtained with our tool use affordance reasoning model. In Sec. 5.4.1 we evaluate the various inter–object Bayesian Network (BN) structures both in simulation and on the real robot, then Sec. 5.4.2 focuses on the experiments about hand affordances, and on the developmental link from hand affordances to tool affordances.

5.4.1 *Evaluation of the Inter–Object Bayesian Networks*

To compare the multitude of possible values for the nodes of the BNs described in Sec. 5.3.2, it is not feasible to collect robotic data with the real robot (for thousands of experiments), therefore we collect most data in simulation. In particular, we gather 2353 experimental trials in the iCub simulator [Tik+08], and 21 trials in the real iCub robot.

For each trial, the *experimental protocol* consists of performing one of the 4 directional movements (see Sec. 5.3.1) upon the acted object while holding a manipulator object (i.e., held object or tool) in the robot's hand, as shown in Fig. 44. Both objects are chosen from a set of 8 possibilities displayed in Fig. 45).

Of the whole set of experiments, a part is used to learn the proposed Bayesian Network models and a part is used for testing, as described in Sec. 5.3.2.

For our tests, we use two *evaluation criteria*:

ACCURACY: defined as the number of correct predictions (e.g., discrete predictions of effect, or tool, or action, depending on the query) over the number of total predictions.

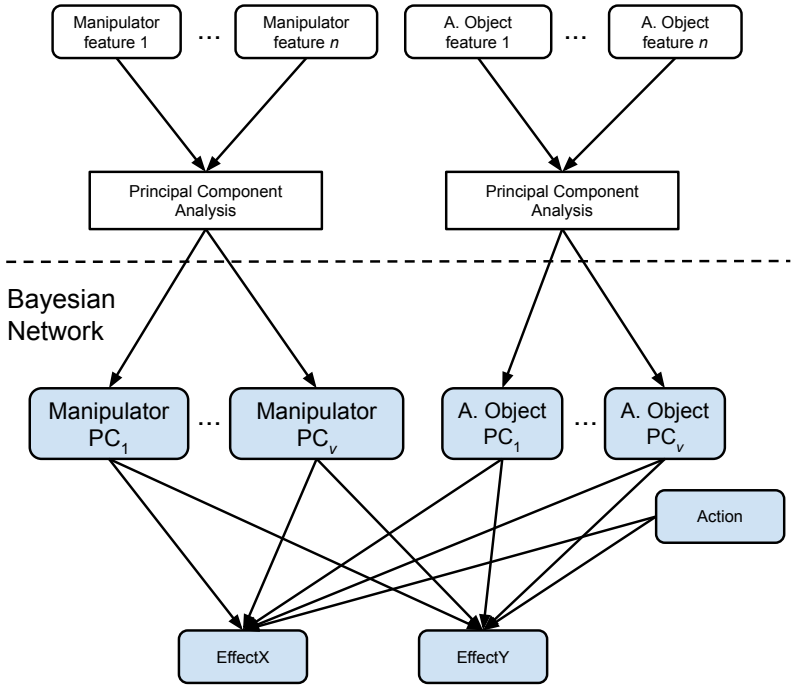


Figure 43: Dimensionality-reduced Bayesian Network structure to encode hand to tool affordances, from [Sap+17b]. A. Object means Acted Object, Manipulator refers to the bare robot hand or to a held object (tool). This structure was specified manually, after having empirically tried different PCA hyper-parameters, see Table 7. The PCA dimensionality reduction is computed on the continuous vectors of the visual features.
In this case, there are two PCA blocks: one for the manipulator visual features, one for the acted object visual features. This is because in this work the concept of toolness was a starting hypothesis, and the focus was on the developmental transition from hand to tool affordances.

Table 7: Hand to tool transition: hyper-parameters used to train the Bayesian Network of Fig. 43 for predicting the distribution of the effects.

parameter	value (and comment)
number of PCA blocks	2 (one for manipulator, one for object)
number of components of each PCA block	2
number of discretization values (bins) of each PCA component	2
number of discretization values (bins) of each Effect node	5
intervals (in meters) of the Effect bins	$]-\infty, -0.06]$, $]-0.06, -0.025]$, $]-0.025, 0.025]$, $]0.025, 0.06]$, $]0.06, \infty[$

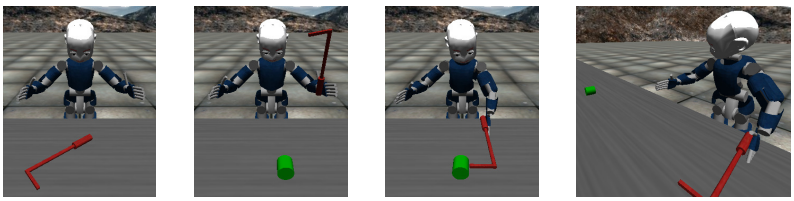
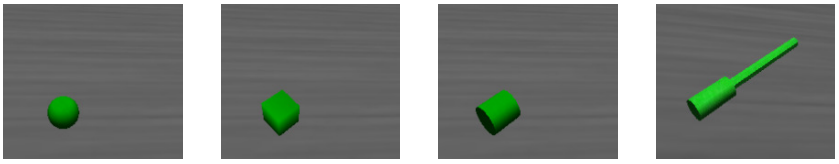
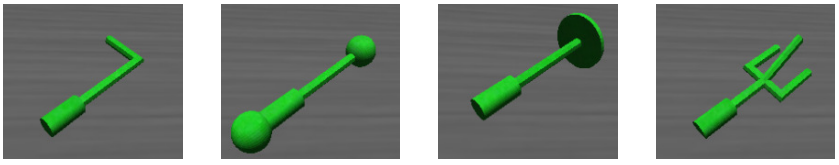


Figure 44: Exploration sequence of tool use in the iCub simulator: (1) the robot acquires the visual descriptors of a manipulator (held object) and its two halves while it is on the table; (2) the robot acquires the visual descriptors of an acted object in its initial state (position); (3) the robot exerts one of the motor actions onto the acted object using the held manipulator; (4) the robot observes the final position of the acted object after a fixed number of frames, permitting to compute the resulting effect compared to the initial state.



(a) Ball. (b) Cube. (c) Cylinder. (d) Stick.



(e) L-stick. (f) Bone. (g) Umbrella. (h) Fork.

Figure 45: Objects used in robot simulation to train affordance Bayesian Networks. Object (h) is only used in one of the evaluation tests of Sec. 5.4.1.1.

Table 8: Data splitting scores when randomly selecting 80% of observations as training data, the remaining observations as test data. Accuracy: higher is better. Distance: lower is better. R.p. stands for random predictions.

	Baseline (13.55% r.p.)	PCA (0% r.p.)	Structure Learning BDe (0% r.p.)	Structure Learning K2 (0% r.p.)
Accuracy	75.90%	80.57%	83.28%	**83.73%**
Distance	9.11%	6.10%	**5.12%**	**5.12%**

DISTANCE: defined as the absolute difference between the prediction (e.g., discrete predictions of effect, or tool, or action, depending on the query) and the real value (i.e., Ground Truth, also discrete). In Tables 8 and 9 it is shown as a percentage, relative to the maximum possible distance.

5.4.1.1 *Effect Prediction*

We evaluate the BNs regarding their capability of predicting effects, given two objects' visual descriptors and the action performed with them, with previously unseen test data. To do this we use two different evaluation techniques: (i) data splitting and (ii) leave-one-out validation.

The first evaluation consists of randomly *splitting* the data in a training set with 80% of observations, the remaining 20% for testing. The exploration data is relative to the 1663 trials corresponding to the seven objects of Fig. 45a–45g. Results are presented in Table 8. The original baseline network is the one with the lowest performance: due to its huge complexity, this network does not generalize well what it learned. 13.55% of the time, this network made a random prediction because an event where all the instantiated variables were seen with the exact same values observed in the test data was never seen during training. The Principal Component Analysis (PCA) network yields a good score, because it has the smallest complexity of all the networks considered. However, the two networks obtained with Structure Learning (i.e., BDe and K2) provide very similar results, being the networks with the best performance on the test data.

The second evaluation is a *leave-one-out* validation, using the same networks as in the data splitting one, but the unseen object of Fig. 45 (h) as test data (690 samples). Results are shown in Table 9. The PCA network has the best performance: being the least complex network makes it the most capable network for generalization to unseen objects. The performance of the other networks gets significantly worse, showing that these networks are too dependent on the training data

Table 9: Leave-one-out scores, testing networks against an object unseen during training. Accuracy: higher is better. Distance: lower is better. R.p. stands for random predictions.

	Baseline (57.25% r.p.)	PCA (0% r.p.)	Structure Learning BDe (52.61% r.p.)	Structure Learning K2 (53.04% r.p.)
Accuracy	44.20%	**73.91%**	48.42%	47.93%
Distance	25.60%	**7.28%**	23.72%	23.97%

(overfitting), so their use on the real robot with a changing environment should be accompanied with an online Structure Learning and parameter learning algorithm, which we do not do (the K2 and BDe structures are learned offline).

5.4.1.2 *Generalization from Simulation to Reality*

In this experiment, the robot executes the left lateral tap action while holding a straight stick. It repeats this action 10 times acting on a ball, 11 times acting on a box. From each iteration, we acquire the Ground Truth (GT). The GT is the discrete index of the displacement bin where the object finished after being acted upon and moving: see p. 42 for the names of the bins, Table 5 for the parameters. Then, we compare the GT to the computed prediction of the resulting effect, given the manipulator and acted objects, by the K2 and PCA network.

In this experiment, we do not present results of the baseline and the BDe networks: they provide random answers, i.e., equal probability for all values, because those networks' structures do not represent well the exact combination of observations in the experiment.

Results for the query $p(\text{Effect}|\,\text{parents}(\text{Effect}))$, where Effect is EffectX or EffectY, and the GTs, are shown together in Table 10.

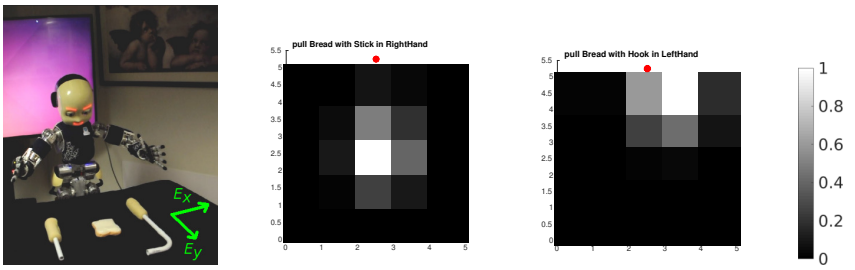
We evaluate how well the predictions match the GTs by computing the *match distance* [RTG00] between their histogram distributions. Being a cross-bin dissimilarity measure, the match distance is suited to cases where the bin order matters. Our bin order for the effects (VN,LN,NM,LP,VP), as defined on p. 42, places more similar displacements in neighbor bins. The maximum value of the distance, in our case, is $d_{\text{MAX}} = 4$, the distance between histograms $(1, 0, 0, 0, 0)$ and $(0, 0, 0, 0, 1)$. It is a special case of the Earth Mover's Distance [RTG00], so it can be interpreted as the amount of mass transported between bins times their distance, to transform one histogram into the other.

Both the PCA network and the K2 structure provide acceptable results (average match distances below 10% of d_{MAX}), with K2 being slightly more accurate (about 7% lower match distances), although the

Table 10: Comparison between Ground Truth (GT) and effect prediction by K2 and PCA networks. See p. 42 for the abbreviations of the five effect bins, Table 5 for the parameters. PCA provides better matches for the ball experiments, K2 for the box ones. Overall, PCA has a match distance 7.3% higher than K2.

	VN			LN		
	GT	K2	PCA	GT	K2	PCA
ball EffectX	0	0	0	0.3	0.0233	0.0137
ball EffectY	0	0.01	0	0.1	0	0.0137
box EffectX	0	0	0	0.0909	0.0233	0.0337
box EffectY	0	0	0.0112	0	0.0204	0.0449

NM			LP			VP			match distance	
GT	K2	PCA	GT	K2	PCA	GT	K2	PCA	K2	PCA
0.5	0.8372	0.7945	0.1	0.1395	0.1644	0.1	0	0.0274	0.4372	**0.3671**
0.5	0.2	0.3699	0	0.23	0.3014	0.4	0.56	0.3151	0.65	**0.3872**
0.9091	0.8372	0.7416	0	0.1395	0.2247	0	0	0	**0.2071**	0.2819
0.4545	0.5306	0.7079	0.5455	0.4490	0.1348	0	0	0.1011	**0.1169**	0.4781



(a) (b) (c)

Figure 46: (a): the iCub robot using affordance reasoning to select the most appropriate tool for achieving a given action, i.e., pulling the bread closer to the agent. (b–c): effect prediction probabilities of the bread motion using the tools, where the robot location is marked with a red dot, and the table area is divided into a grid of squares.

K2 structure has the peculiarity of the EffectX node being conditionally independent from acted object features (see Fig. 40a). This explains why the K2 EffectX rows of Table 10 have equal values, regardless of the acted object.

5.4.1.3 *Tool Selection*

In the example of Fig. 46, the robot has to *select the most appropriate tool* for pulling the bread object closer to it. Figs. 46b and 46c show the motion effect predictions (posterior probabilities) using the Stick and the Hook tools, respectively. In the figures, the table is discretized into

a grid, and the robot location is represented by the red dot. The values of the effect predictions represented by the figures are the following:

$$p(\text{EffectX}, \text{EffectY} \mid A = \text{pull}, T = \text{Stick visual features}, O = \text{Bread visual features}) = \begin{bmatrix} 0 & 0.0030 & 0.0290 & 0.0118 & 0 \\ 0 & 0.0185 & 0.1780 & 0.0722 & 0 \\ 0 & 0.0374 & 0.3602 & 0.1461 & 0 \\ 0 & 0.0099 & 0.0952 & 0.0386 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{21}$$

$$p(\text{EffectX}, \text{EffectY} \mid A = \text{pull}, T = \text{Hook visual features}, O = \text{Bread visual features}) = \begin{bmatrix} 0.0085 & 0.0085 & 0.2257 & 0.3704 & 0.0681 \\ 0.0037 & 0.0037 & 0.0973 & 0.1597 & 0.0294 \\ 0.0003 & 0.0003 & 0.0083 & 0.0136 & 0.0025 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{22}$$

The posterior distributions in (21) and (22) show the expected pulling movement effects when using the Stick or the Hook, respectively. The latter achieves higher values along the desired direction (i.e., in the values along the first two rows, corresponding to the motion of the acted object Bread towards the agent, as desired when performing a pulling action). Therefore, the robot selects the Hook. This happens because the Hook possesses shape characteristics similar to the ones of tools that have achieved successful *pull* actions during learning, therefore it yields a higher probability of the desired motion compared to the Stick.

5.4.2 *Evaluation of the Hand to Tool Transition*

We train a probabilistic model of hand affordances, relating visual features of (i) different robotic hand postures and (ii) different objects, with the resulting effects caused by the robot motor actions onto such objects. Training data are collected during several experiments in which the iCub robot (see Sec. 3.1) performs manual actions on objects located on a table. We publicly release a novel dataset of hand posture affordances[4], and we test it for generalization against an available dataset of tool affordances [Deh+16a].

We now present the results obtained from our hand affordance model, and we assess its performance.

5.4.2.1 *Hand Affordances Dataset*

Our experimental data is obtained by making manipulation experiments on an iCub humanoid robot, in a setup like the one shown in

[4] `https://github.com/vislab-tecnico-lisboa/affordance-datasets`

Table 11: Hand to tool transition: accuracy of the Bayesian Network with different training and test data. Chance level is 4% (see text).

training set	test set	accuracy
80% hand	20% hand	72%
80% tool	20% tool	58%
100% hand	100% tool	53%

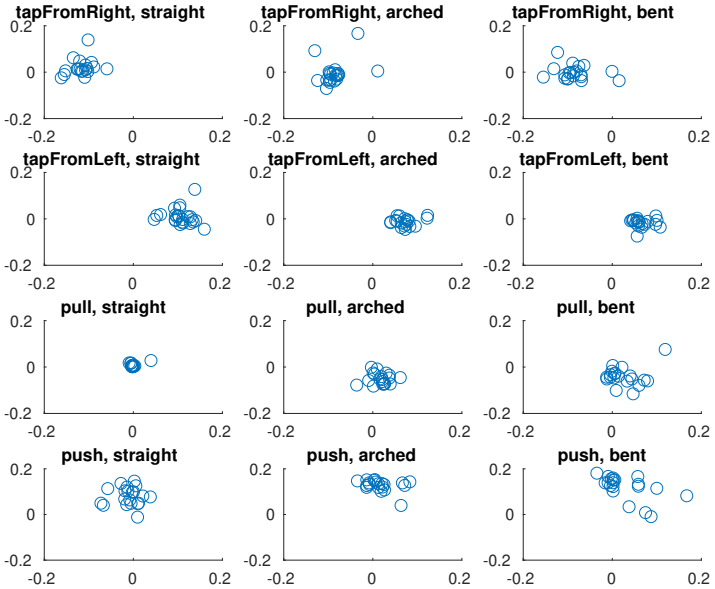
Fig. 41, using its left arm for data collection. We consider 4 motor actions A (tapFromRight, tapFromLeft, pull, push), 2 objects O (lego piece, pear), 3 hand postures H (straight fingers, bent fingers, arched fingers; shown in Fig. 42). We extract the visual features from both O and H (before performing the actions). The dataset is publicly available: see footnote [4] on p. 99.

In Fig. 47 we show the distributions of the motion effects onto acted objects caused by the robot influence when it touches objects with its manipulator. In particular, Fig. 47a shows the effects of using the different hand postures. For comparison, Fig. 47b depicts the effect of using the elongated tools (Fig. 48) on the same objects. Visual inspection reveals the similarities in the effect of using tools or hands, for example, tapping from left usually results in the object moving to the right. Another prominent similarity is that pulling with a stick or with the straight hand posture causes only minimal movement.

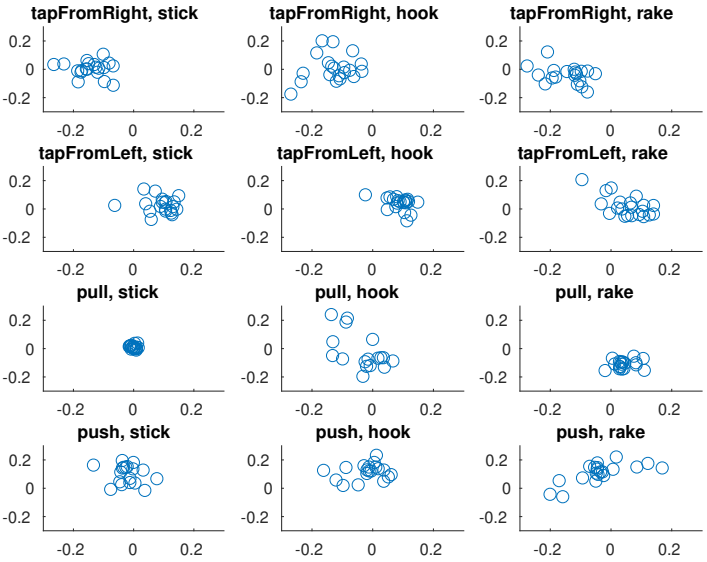
We also do data augmentation on the Hand Affordances Dataset. We assume that the affordance of an object and of a robot manipulator is viewpoint-invariant. By exploiting this notion, it is possible to artificially augment the trials data using *multiple views* of manipulators and objects. In all of the following experiments, we have used at least 10 viewpoints of each object and manipulator, effectively multiplying the number of available samples by more than 100 times.

5.4.2.2 *Effect Prediction*

One way to assess the quality of the learned BN of Fig. 39 is to predict the effect distribution, given the descriptors of manipulator, object, and action, i.e., the direct application of (20). As before, we have empirically divided the effect distribution along each axis into five bins (a list of the hyper-parameters that we used for training our network is reported in Table 7 for reproducibility). We use the accuracy metric from Sec. 5.4.1, which is the fraction of correct predictions by the network (i.e., when it predicts the correct effect bin out of five) among the total number of predictions performed. Since there exist two axis directions, a random predicting machine would be correct 1/25 of the time. At the end of this assessment, we divide by the number of total predictions performed.



(a) Motion caused with the robot *hands* when using different actions and hand postures, as observed when interacting with 2 objects multiple times in our experiments.



(b) Motion caused with *tools* when using different actions and tool types, taken from [Deh+16a]. Here we show only the interactions with 2 objects, to be consistent with Fig. 47a.

Figure 47: Motion caused by different robotic manipulators (hands and tools) when using different actions and manipulator morphologies: in Fig. 47a we use different hand postures, whereas in Fig. 47b we vary tool types for comparison. Each plot displays the geometrical displacement along horizontal and vertical direction (in meters, measured from the object initial position) from the point of view of the robot (the robot is at the 0 in the x-axis marker). For example, tapping an object from the right (tapFromRight action) usually results in making the object shift to the left direction; pulling an object closer only works if the manipulator morphology is appropriate.

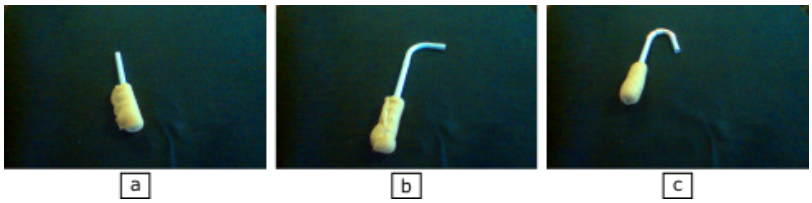


Figure 48: Hand to tool transition: the three baseline tools used in [Deh+16a], (a) stick, (b) rake and (c) hook. They provide different affordances when grasped by the hand of the robot and employed for performing motor actions onto objects. In this chapter, we consider these tool affordances as a comparison term to assess our novel hand affordances.

Using the same network parameters but training with different data, we obtain the accuracy scores reported in Table 11. To explain these scores, we note that motor control on the iCub is noisy, and actions on this platform are not deterministic or repeatable (e.g., when commanding the robot twice starting from an initial position, the same motor command can produce two slightly different configurations). Even so, in Table 11 and in Fig. 47 we see that tool effects are more varied than hand effects, making tools less reliable (i.e., more noisy) than hands. Nevertheless, by only training on the hand data, we obtain an accuracy that is comparable with the case where the network is trained on tool data, demonstrating the generalization of our proposed method.

5.4.2.3 *Tool Selection from Hand Affordance Knowledge*

One question that we wish to investigate is the following: if an agent gains the knowledge of how its hand postures can affect the environment, can it generalize this knowledge to other tools which look similar to its hands? To answer this question in the scope of the presented scenario, we conduct the following experiment. We suppose that an agent has defined a goal, for example to pull an object towards itself. It knows that the correct action for this task will be to *pull* ($A = \text{pull}$), however the object is out of the hand's reach and one of the presented tools of Fig. 48 must be selected for the task.

In this scenario, an agent looks at the available tools and at the acted object and performs a mental simulation of the known action along the two effect displacement directions:

$$p(\text{EffectX} \mid A = \text{pull}, M = \text{tool visual features}, O = \text{object visual features}),$$
$$p(\text{EffectY} \mid A = \text{pull}, M = \text{tool visual features}, O = \text{object visual features}),$$
for each tool available in the scene.

The above expressions return a posterior distribution, in the form of a 5×5 matrix, because of the way that we discretized the table in front of the robot.

Table 12: Tool selection results obtained from our "hand to tool" (HT) network, compared to ones obtained from the baseline "tool to tool" (TT) network [Deh+16a].

action	stick	hook	rake
tapFromRight	HT: 1.0 (TT: 1.0)	HT: 1.0 (TT: 1.0)	HT: 1.0 (TT: 1.0)
tapFromLeft	HT: 1.0 (TT: 1.0)	HT: 1.0 (TT: 1.0)	HT: 1.0 (TT: 1.0)
pull	HT: **0.5385** (TT: 0.1538)	HT: **0.6154** (TT: 0.1538)	HT: **1.0** (TT: 0.4615)
push	HT: 1.0 (TT: 1.0)	HT: 1.0 (TT: 1.0)	HT: 1.0 (TT: 1.0)

A tool is selected if it is expected to cause a movement of the target object *along the desired direction*, and it is rejected if no movement is predicted, or if the object is predicted to move against the desired direction. Because in this work we divide the direction into five bins (see Sec. 5.4.2.2), *we compare the sum of the predictions in the two desired-movement bins against the sum of the predictions in the remaining bins.* Since there was no interaction with the tool, it is necessary to generalize from the knowledge of previous hand explorations to tools in a zero-shot manner.

As an example of a successful generalization, the agent should predict that pulling an object with a stick is pointless, because the agent has already experimented pulling objects with a straight hand posture, and the visual descriptors of straight hands are similar to those of a stick. Table 12 shows the result of this inquiry. We have also implemented a baseline in which the agent has already experienced with the tools and is asked to select the correct tool. As expected, all the tools can be used for the desired effects, and it is only the pull action which requires a tool with a specific shape. The numbers are normalized, as they correspond to different views of the tool and object, and they reflect the percentage of the cases where that specific tool was selected.

In this experiment, being familiar with the available tool shape in advance (i.e., encountering a tool that is similar to one of the three baseline tools from Fig. 48) provides an advantage.

5.5 CONCLUSIONS AND FUTURE WORK

In this chapter, we have presented a computational model which permits a robot to use tools, and showed a number of experiments to this end, both in simulation and on a real robot.

First, we specialized our system for visual object affordances to let it support pairs of simultaneous objects, their mutual relationships, and the relationships between object sub-parts (e.g., handle part and effector part of a tool). This specialized model is a Bayesian Network (BN) that relates robot actions, visual features of manipulators (e.g., tool tips), visual features of objects and produced effects, allowing a humanoid robot to predict the effects of different manual actions.

Being probabilistic, our model is robust in dealing with the uncertainty that exists in real world signals (in the next chapter, we will see a case study of this robustness used in the context of robotic action planning in uncertain environments).

Second, we investigated different structures of the BN that implement our computational model, obtained either through Structure Learning (K2 and BDe algorithms) or Principal Component Analysis (PCA) dimensionality reduction: we compare them in terms of complexity, representation capability and generalization, with respect to a baseline fully connected structure. Our results show that both Structure Learning and dimensionality reduction techniques allow to reduce the complexity of the model while improving the estimation performance. Specifically, the PCA model is characterized by the lowest complexity and the best performance in generalization to novel objects (Tables 6 and 9), while the K2 model performs slightly better in representing the experienced data (Table 8). Moreover, the model learned in simulation can be used to reasonably predict the effects of the actions on the real robot; in this case, the structure obtained with the K2 algorithm shows the best average performance (Table 10).

Finally, we used our model to learn a representation of hand affordances (i.e., affordances perceived when using different hand apertures), and we investigated how such a hand affordance model can adapt to a tool affordance model (i.e., affordances perceived when using tools). Interestingly, we show how the hand affordance knowledge, acquired by the robot through autonomous exploration of different actions, *hand postures* and objects, can be generalized to *tool use*, and employed to estimate the most appropriate tool to obtain a desired effect on an object, among a set of tools that were never seen before.

Regarding the developmental link from hand affordances to tool affordances, we should clarify that our results show that, in *some specific cases*, it is indeed possible to generalize what was learned about hand affordances to tools that were never seen before. This is limited to a subset of all the possible human-made tools that a humanoid robot could see and possibly use. However, the previous knowledge about

hand affordances can give the robot the possibility to make a good initial estimate of how a tool could be used.

In terms of future work, it would be interesting to investigate how further sensorimotor experience with tools can be integrated in the learned model, and possibly permit better predictions. Also, another possible avenue is to study the developmental link in the opposite direction, from tools to hands: can the knowledge acquired with a specific tool be re-used to estimate the effects of manual actions without the tool, or to shape the robot hand in the best posture to achieve some effects?

We believe that the results from this chapter can be useful for the developmental robotics community, because we propose a robot learning framework that presents practical advantages for robot autonomy, at least in the limited number of situations that we analyzed in our experiments, since it permits to generate meaningful predictions about a non-finite set (i.e., tools) from experiences in a finite set (i.e., hand postures).

6

AFFORDANCES AND PLANNING

This chapter presents a case study of the application of the ideas presented in the previous chapters (namely affordances, language, and tool use) within the scope of the European research project POETICON++. See Fig. 49 for a brief description of that project.

We show how robot sensorimotor knowledge (learned affordances) can be combined with symbolic reasoning, forming a unified *planning architecture*. We use probabilistic reasoning to permit a robot to carry out a complex manipulation task, requested by a human user with verbal language, under challenging conditions and external disturbances.

This chapter is the subject of the following publications:

- Alexandre Antunes, Lorenzo Jamone, Giovanni Saponaro, Alexandre Bernardino, and Rodrigo Ventura. "From Human Instructions to Robot Actions: Formulation of Goals, Affordances and Probabilistic Planning". In: *IEEE International Conference on Robotics and Automation*. 2016, pp. 5449–5454. DOI: 10.1109/ICRA.2016.7487757.

- Alexandre Antunes, Giovanni Saponaro, Anthony Morse, Lorenzo Jamone, José Santos-Victor, and Angelo Cangelosi. "Learn, Plan, Remember: A Developmental Robot Architecture for Task Solving". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017, pp. 283–289. DOI: 10.1109/DEVLRN.2017.8329819.

- Giovanni Saponaro, Alexandre Antunes, Rodrigo Ventura, Lorenzo Jamone, and Alexandre Bernardino. "Combining Affordance Perception and Probabilistic Planning for Robust Problem Solving in a Cognitive Robot". In: *Autonomous Robots* (2018). Under review.

The rest of this chapter is structured as follows. Sec. 6.1 gives the background and motivation for the case study. Sec. 6.2 overviews the literature of robot reasoning architectures similar to ours. Sec. 6.3 states our main objective, assumptions and approach. Sec. 6.4 illustrates our proposed architecture and its constituent parts. We report experimental robot results in Sec. 6.5. Finally, in Sec. 6.6 we give our concluding remarks.

6.1 MOTIVATION

As robots are increasingly moving to unstructured settings (e.g., homes and public places, see [Pra+16]), they must be able to carry out *com-*

Figure 49: Logo of the POETICON++ project (`http://www.poeticon.eu/`). The main objective of this project was to develop computational models and machinery for robots that would allow them to generalize motor execution and visual experiences beyond the ones known at learning time, resorting to natural language reasoning.

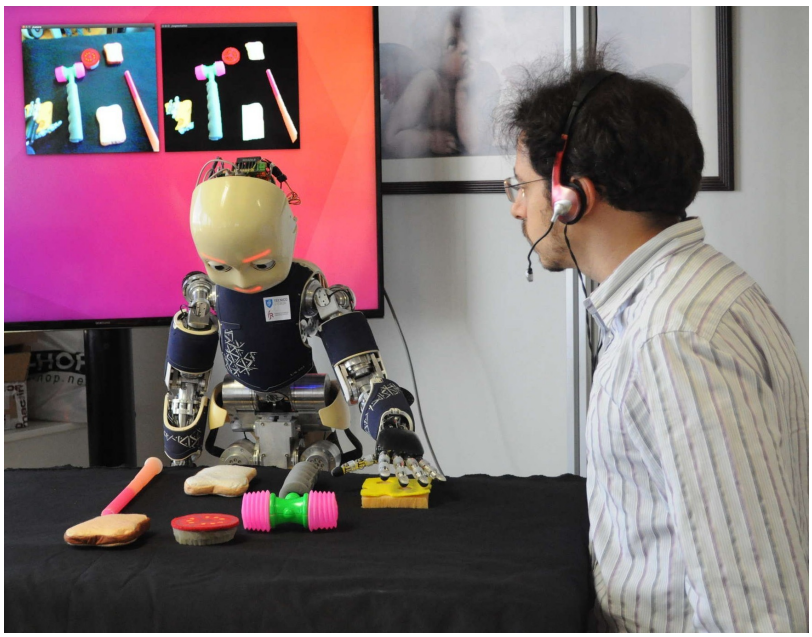


Figure 50: A robot performing a complex manipulation task after receiving a verbal instruction from a human. In the back screen, images from the robot cameras showing visual perception routines.

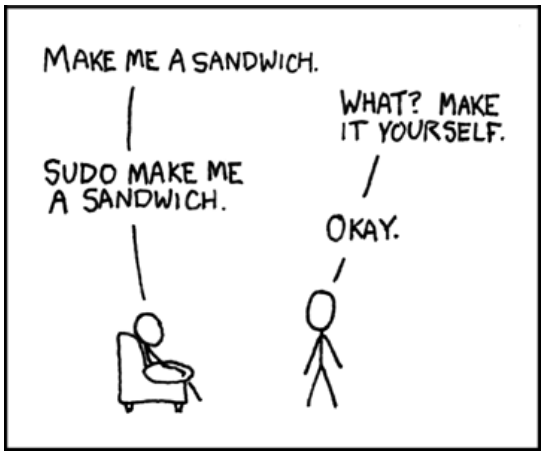


Figure 51: A comic strip about the subtleties of asking another agent to prepare a sandwich. Reproduced under the CC BY-NC 2.5 license from `https://xkcd.com/149/`[1]. In the POETICON++ project, the final demonstration also consisted of asking a robot to make a sandwich, using the available tools and ingredients present in the scene, as explained in this chapter.

plex manipulation tasks alongside humans, also in the presence of *uncertainty*. Indeed, the shift from industrial to service robots bears the issue of how to design artificial agents that can work effectively with humans performing manual tasks, in a scenario like the one of Fig. 50: a human verbally instructing a robot to perform a manipulation task. In particular, major challenges faced by the robot in such a situation are: (i) how to understand and execute instructions provided by human users (see Fig. 51), taking into consideration the properties of the available objects and the uncertainty in robot action and perception; (ii) how to monitor and adapt to an unstructured (non-industrial) environment which will be constantly evolving and changing during task execution [HDH11].

Bearing these challenges in mind, in this chapter we present a robot action selection system that combines (i) robot sensorimotor knowledge (in the form of learned affordances) with (ii) symbolic reasoning, by resorting to a unified probabilistic representation; moreover, this system is seamlessly integrated in a bigger control architecture which includes (iii) formulation of robot goals from human verbal requests,

[1] Explanation of the comic strip of Fig. 51: on UNIX and Linux computer systems, users can be assigned to all kinds of rights, for example rights to access to certain directories and to execute certain commands. The *sudo* command lets certain authorized users override these policies by executing the command (everything after the word *sudo* on the command line) as the administrator root user. Forgetting to start the command with *sudo* is a fairly common and frustrating mistake for people who administer UNIX-like systems. They then need to repeat the command with *sudo*, at which point the computer responds obediently, and everything works smoothly.

(iv) continuous perception of the world through robot sensing, (v) execution monitoring and re-planning with heuristics.

This chapter contains: a thorough description of our framework from the systems perspective, including the strategies (e.g., heuristics) devised for applying our system profitably; qualitative tests of our architecture on a real humanoid robot; a quantitative evaluation of the architecture, simulating the execution of a human-specified instruction under varying levels of uncertainty and with different planning strategies and heuristics.

We have implemented our architecture on the iCub humanoid robot (see Sec. 3.1), validating our system, showing how the overall architecture allows complex robotic problem solving of manual tasks specified by humans, coping efficiently with different levels of uncertainty. We publicly release our code[2], including a simulated symbolic reasoner for validating the probabilistic planner under challenging conditions, and real robot sensorimotor data used for affordance learning. The public repository contains additional material, e.g., a video of the system implemented on the iCub.

6.2 RELATED WORK

In this case study we consider a *cognitive architecture* that allows a humanoid robot to understand generic instructions provided in natural language (i.e., *Natural Language Understanding* or NLU), and to execute them by combining *affordance perception* and *action planning*: below, we report relevant previous works in these areas.

6.2.1 *Cognitive Architectures*

In Artificial Intelligence (AI) and robotics literature, several comprehensive cognitive architectures for making robots accomplish complex tasks have been proposed [VHF16]. Given their interdisciplinary scope, these architectures are typically modular. They contain multiple components which address the specific requirements (e.g., NLU, planning under uncertainty, probabilistic inference, sensor fusion, execution monitoring, robot manipulation), although there is no single framework that is suited for all applications, given their great diversity [Bee+16]. Examples of comprehensive cognitive architectures for robots are [PRM10; HDH11; SA12; Lem+17; Mou+18]. While these systems present solid theoretical foundations in behavior-based control and robot simulation results, they do not focus on robustness to uncertainty and noise, on re-planning, or on the applicability to general scenarios (i.e., not being restricted to one specific task) on real robot platforms such as humanoids equipped with many degrees of freedom. An interesting

[2] `https://github.com/robotology/poeticon`

work is [RBC15], which shows a reasoning system for transferring goal-oriented skills by imitation between two agents, human and robot. This system gives an iCub humanoid robot the ability to recognize the action performed by a human demonstrator, to extract semantic properties from that action, and to replicate the goal of that action autonomously at a later moment. Our work has analogies with it, given that we also introduce a goal reasoning and execution system deployed on the iCub robot, however (i) we consider actions expressed in natural language, being then grounded and translated to a (potentially long) sequence of sub-goals for autonomous robot planning and execution; (ii) we focus on the unified probabilistic integration of robot affordances with probabilistic planning, which together provide our system some leeway to recover from unexpected events and failures, as well as generalization ability when presented with novel objects not seen in the training phase.

In [Cac+17], a framework for imitation learning of sequential tasks is proposed, using human demonstrations to have a robot execute a pizza topping task. That work is focused on learning movement primitives without supervision in dual-arm assembly settings, assuming for simplicity that the downmost ingredient of the structure to assemble (i.e., the pizza dough) is known *a priori* and that it is located within an area reachable by both robot arms; additionally, there is no explicit fault detection mechanism to monitor task execution and to react to failures. By contrast, our work addresses arbitrary locations of the objects to be used by the robot (including ones that are not reachable with bare robot hands but might become reachable when resorting to tools). In addition we explicitly incorporate mechanisms and heuristics for re-planning and recovery from failure, as mentioned above.

In [Mou+18], an architecture for complex collaborative tasks between a human and an iCub humanoid robot is proposed, integrating different robot skills such as perception, manipulation, and social interaction capabilities with a speech interface and generation of verbal descriptions of events. That work has similarities with our proposed approach, being targeted to a human–robot collaboration task using language. However, the authors employ planning at a local level, meaning that they envision a list of successive actions requested explicitly by a human user to a robot, one after another (i.e., “take the cube”, “point to the octopus toy”), and then, for each action, a sequence of sub-actions with pre-conditions and post-conditions is considered, selecting and executing the one with the shortest length. Each motor action is repeated (up to a pre-defined timeout) until the post-conditions are met, and visual-linguistic knowledge is incorporated to guide action selection. By contrast, (i) we use probabilistic planning at a global level, going from a single human instruction in natural language (i.e., the final goal) to an entire plan that instantiates a list of sub-goals and robot motor actions, reasoning on their probabilities of success, which are updated during

task execution and monitoring; (ii) when an object is needed but it is not reachable by the robot, we use tool affordance knowledge to have the robot select the best tool in order to bring the object closer autonomously and carry on with the greater plan, whereas in [Mou+18] the robot asks the help of the human partner to position the object closer to the robot; (iii) to address failures of individual actions, we use probabilistic strategies and heuristics (e.g., the robot tries to use an alternative arm or an alternative ingredient if the first one has failed repeatedly) instead of pre-defined timeouts.

6.2.2 *Natural Language Understanding*

The work by Tellex [Tel+11b; Tel+11a] is geared at interpreting language commands given to mobile robots with statistical symbol grounding, i.e., mapping words to syntactic structures of concrete objects, paths and events [Har90], possibly with unsupervised learning [Tan+16]. Along with grounding, many works also handle symbol anchoring [CS03; Lem+12; Elf+13]. Anchoring refers to the process of linking language symbols to real-world representations acquired with robot sensing: it requires appropriate strategies when the process takes place over time in a dynamic environment. We implement the anchoring aspect in our world modeling component (see Sec. 6.4.4). In [Mat+12; Mat+13], language and perception signals are learned jointly by a robot, which is then able to disambiguate generic instructions (e.g., "go") using contextual cues; however, no error recovery mechanism is present. Similarly, in [CM11] a semantic parser is learned by observing a human instructor perform a set of motor actions related to navigation, also without recovery from failure.

When it comes to available *general* knowledge bases of natural language applicable to robotics (in the sense that they are not just geared towards a specific problem like understanding navigation instructions, but span multiple domains), *semantic reasoning* engines for translating human language into robot instructions have been proposed, for instance PRAXICON [Pas08; MP16] and DIARC [Dzi+09]. DIARC is an architecture for translating natural language instructions and executing them on a robot. However, while that system has a failure detection mechanism, it does not re-plan, nor does it use prior robot action knowledge to generalize to unseen objects and tools as done in the present work. In [Mis+16], a statistical method for grounding natural language instructions to specific robot environments in kitchen and home scenarios is proposed, being able to handle missing and incomplete instructions in the human language input, such as in "heat up the water, then cook the ramen" (inferring how to cook the object in this scenario). Similarly, the framework of [ETF16] analyzes specific language understanding problems and ambiguities that frequently arise in human–robot interaction. In PRAXICON, which we adopt in our

architecture, a human instruction is decomposed into a set of deterministic human-like actions such as "hand grasps knife, knife cuts tomato, ...". However, this type of sequence does not take into account the geometric world around the robot, thus requiring planning over each instruction for a physical implementation.

6.2.3 *Affordance Perception and Planning*

Affordances are useful in robotics because they model essential properties of environment objects in terms of the actions that a robot is able to perform with them (see Ch. 1 and 2). We now cite a few works that link affordances with action planning, being relevant for this chapter.

In [Mol+18] the relational affordances between objects pairs are exploited to plan a sequence of actions to achieve a desired goal, using probabilistic reasoning; however, how these interactions are affected by different geometrical properties of the objects is not investigated. Some authors have suggested an alternative computational model called *Object–Action Complexes (OACs)* [Krü+11], which links low-level sensorimotor knowledge with high-level symbolic reasoning hierarchically in autonomous robots, similarly to how we will use affordances for planning in this chapter. However, the practical uses reported up to the time of writing this thesis correspond to simple examples, pre-defined transition rules and high-level relations, excluding typical problems that characterize real-world robot behaviors: noise, errors in perception, execution failure, unexpected events. By contrast, the literature on robot affordances reports results from real (i.e., not simulated) robot experiments that tackle those problems explicitly.

In [UP15a] a robot first learns affordance categories and then high-level logical rules, which are encoded in Planning Domain Definition Language (PDDL), enabling symbolic planning with off-the-shelf AI planners. In a follow-up work [UP15b] the generated plans are used in a real-world object stacking task and new affordances that appear during plan execution are discovered. The robot is able to build stable towers exhibiting interesting reasoning capabilities, such as stacking larger objects before smaller ones, and the ability to generalize by object types (assuming that the robot is able to visually detect objects, to extract their features before learning, and that object relations such as relative differences in diameter have been previously learned in simulation).

The system of [UP15a; UP15b] focuses on learning a parameterized symbolic representation from robot manipulation experiments, and is useful for planning probabilistically with unknown objects. Our system focuses on combining affordance perception with probabilistic planning, too, but we include tool use affordance capabilities (to overcome the geometric difficulties of objects being far away from the robot), as well as heuristic strategies for recovering from failures and re-planning, thus

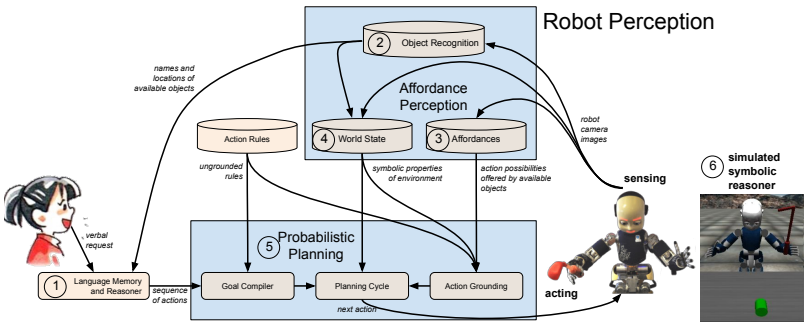


Figure 52: POETICON++ architecture exposing the main components of our system. In the bottom left, a human user expresses an instruction in natural language. In the bottom right, a robot reasons about the environment and executes the instruction. The robot can be real or simulated. The components (software modules) of the system are represented as gray rectangles, or gray cylinders in case they incorporate a knowledge base. Arrows indicate data flow.

providing a system that is robust under manipulation noise. Furthermore, the tools that our system is able to use do not need to be previously learned by object recognition.

Task and motion *planning* have been combined together in several AI and robotic works [Loz+87; Sri+14]. Hierarchical planning [Nou98] has provided algorithms like SHOPS2 [Gol09; WMR10]. These methods combine symbolic and geometrical planning, however they usually employ one static plan to be followed and completed by the agent, not allowing failures or re-planning. Further work exists on simultaneous plan and execution (Hierarchy Planning in the Now or HPN [PZK07; KL11]), focused specifically on the execution of geometrical problems, merging symbols with geometric planning. The problem of real-time planning and execution requires an algorithm that can adapt to changes in the state. HPN introduces this by updating a world state model at each step, and planning from there. By using a hierarchy of actions, it transforms a hard problem into several smaller ones that are more easily solved by a planner. This approach was initially suggested by Nourbakhsh [Nou98], where a big problem would be turned into smaller ones by completing sub-goals and re-planning.

Multi-level planning based on [Nou98] has been explored in [WMR10; KL11; LK14], using a set of pre-determined robot instructions. By contrast, our approach creates a full chain, from a very abstract human instruction to specific motions at the lower control level, as shown in Fig. 52. Our proposal uses Nourbakhsh's concept applied to *probabilistic planning*. In particular, we use the PRADA probabilistic engine [LT10], which has proven to be both fast and accurate in planning the first best action of the plan, ultimately permitting real-time robot operations, and that allows to incorporate the prior robot knowledge

encoded in probabilistic terms (in our case, through the affordance perception described in Ch. 5).

6.3 MAIN OBJECTIVE, ASSUMPTIONS AND METHOD

Our architecture aims to equip a robot with the ability to realize manual tasks specified by humans with natural verbal instructions. We assume that the robot possesses basic action and perception capabilities to interact with the environment, in line with Sec. 3.2. However, we assume that these have unmodeled uncertainty, probability of failure and unpredictability: we refer to the combination of these phenomena as *noise*. Our main objective is to be robust to such *noise*.

The core aspect that makes our architecture robust is an action selection system that combines affordance perception and probabilistic planning, which are aligned through a common probabilistic framework: affordance perception is realized by approximate inferences over a joint distribution of variables estimated through a Bayesian Network (see Sec. 2.1.2 and Ch. 5), whereas symbolic planning is achieved with probabilistic relational rules encoded with a structured Dynamic Bayesian Network [LT10]. Affordance perception allows to predict the (probabilistic) effects of a certain action on a certain object, and the planner uses those predictions to discover the action sequence that has the highest probability to lead to a desired final effect (i.e., problem solving); in other terms, the probabilities associated to the probabilistic symbols used by the planner (i.e., the effects of individual actions) are inferred through affordance perception.

Indeed, the integration of affordance perception and symbolic planning is particularly interesting since they are complementary in different ways.

First, both processes have a learning component, but with different dynamics. Learning how to perceive object affordances is a long-term process that involves repeated sensorimotor experiences, resulting in a permanent knowledge which can be reused and generalized later (e.g., the robot learns what actions should be "seen" in an object, and what effects could be predicted, based on its sensorimotor capabilities, and it is then able to infer the action possibilities of never-seen-before objects). By contrast, probabilistic planning is a real-time process in which the probabilities associated to the symbols (that can be initialized with the affordance predictions) can be adaptively corrected and fine-tuned during the execution of a specific plan (e.g., the robot realizes that one action is not causing the predicted effect, and the probability of that effect is decreased temporarily). Therefore, affordance perception is based on previous learning, and probabilistic planning permits real-time adaptation.

Second, the rules of symbolic planning need grounding [KKL14; KKL18], which depends both on robot sensorimotor capabilities and on object

properties. Affordance perception provides such grounding, by instantiating the probabilities of the symbols based on the robot perception, subject to its previous sensorimotor experiences. Notably, describing these symbols with a probabilistic representation, instead of deterministically, allows to better cope with the noisy and uncertain nature of robot perception and action, by taking such information into account at a planning level, based on the previous sensorimotor experience of the robot: this makes our system robust to unmodeled sources of noise and uncertainty, such as robot miscalibration, limited dexterity in manipulation, and noisy visual perception.

6.4 PROPOSED APPROACH

Fig. 52 is a sketch of our system. It shows that two agents are involved: a human one who expresses an instruction in natural language (bottom left), and a robot which reasons about the environment and acts on it to execute the instruction (bottom right). Individual components (software modules) are represented as gray rectangles, or gray cylinders when they incorporate a knowledge base. Arrows indicate data flow. Note that the robot can be real or simulated: this choice does not affect other components.

Next, we describe the system parts following the numberical indexes within Fig. 52: (6.4.1) language-based semantic knowledge about the task, used when the interaction starts through speech; (6.4.2) object recognition capabilities; (6.4.3) prior robot knowledge in the form of learned object affordances (action possibilities); (6.4.4) perceptual information about the current robot context, including world sensing and modeling; and (6.4.5) probabilistic planning to formulate sub-goals, use them to solve the task, and recover from failures.

6.4.1 *Language Memory and Reasoner*

The PRAXICON semantic memory and reasoner (see Sec. 6.2.2 and [Pas08; MP16]) interprets task-oriented natural language human instructions (e.g., "prepare a salad") and computes a possible sequence of motor actions to accomplish the task, conditioned on the list of currently available object names in the robot surroundings, provided by the Object Recognition module. The resulting individual actions are expressed in a quasi-natural language that is comprehensible by us (e.g., "hand grasps knife, knife cuts tomato, ..."). Such a sequence is intuitive for humans, but it has limited applicability for robots, because those action formulations in natural language do not take into account constraints of the environment surrounding the agent, its motor capabilities, and the possibility of failing one or more actions.

6.4.2 *Object Recognition*

We use a state-of-the-art object recognizer based on deep learning, specialized for humanoid robots [Pas+16][3]. From robot camera images, this provides real-time classifications that are robust to changes in scale, light and orientation. Objects are visually segmented from the background based on their luminosity; they are encoded as the output of the highest layer of a convolutional neural network. Then, each segmented object in the robot view is assigned to a label with a Support Vector Machine classifier.

Note that, in Fig. 52, the Object Recognition visual pathway is separate from the one associated to Affordance Perception. The former is concerned with obtaining the *labels* of the objects, the latter with reasoning about their functionalities. This is consistent with the two-streams hypothesis of neuroscience (see Sec. 1.4.1).

6.4.3 *Affordance Perception*

Affordance perception allows to predict the use of novel objects, not previously seen or trained. To do this, we adopt the scenario from Sec. 3.2 and the computational model from Ch. 5, with the environment variables being: action, manipulator (held object) shape descriptors, acted object shape descriptors, resulting effects. Refer to the detailed example of affordance prediction from Sec. 5.4.1.3. In short, what determines the affordances of an object are its shape characteristics, combined with the agent sensorimotor experience and learning.

6.4.4 *World State*

This component collects information about the environment objects and robot parts. It is a dynamic database containing a short-term memory of symbolic properties needed by the probabilistic planner for next action selection. We use the World State for a robust anchoring (see Sec. 6.2.2, [CS03; Lem+12; Elf+13]) of physical entities to symbols, accounting for persistence in time and issues like occlusions and failures originating from robot mechanics and control issues, or perception errors (e.g., vision, object recognition).

Object entities and hand entities share common properties, listed in Table 13. In addition, they also possess entity-specific properties, reported in Table 14 and 15. The symbols that are monitored are: (i) entity ID, (ii) label name, (iii) type (hand or object). An object entity also includes: (iv) spatial position on the table, (v) shape descriptors of the whole segmented shape, (vi) shape descriptors of the top and bottom sub-parts, (vii) in which hand it is grasped, (viii) which

[3] `https://github.com/robotology/iol`

Table 13: World State symbols that pertain to all types of entities (hands and objects).

symbol	description	domain
(i) `id`	numerical identifier	integer number
(ii) `name`	human-readable label	string
(iii) `isHand`	flag to distinguish hands	true if hand, false if object

Table 14: World State symbols that pertain to object entities.

symbol	description	domain
(iv) `position`	2D spatial coordinates	vector of numbers
(v) `desc`	shape descriptors of whole object	vector of numbers
(vi) `tooldesc`	s. d. of top and bottom parts	vector of numbers
(vii) `inHand`	name of hand holding this object	left, right or none
(viii) `onTopOf`	objects that are below this one	vector of IDs
(ix) `reachableList`	entities that can reach this object	vector of IDs
(x) `pullableList`	entities that can pull this object	vector of IDs

Table 15: World State symbols that pertain to hand entities.

symbol	description	domain
(xi) `isClear`	availability of the hand	true if free, false if busy

objects are below, (ix) which entities can reach it, (x) which entities can pull it. A hand entity also includes (xi) its availability.

Note that, although these symbols are quite general and could be used in a wide range of manipulation tasks, different or additional symbols might be needed for other tasks (e.g., a room cleaning scenario). Thanks to the modular nature of our architecture, new World State symbols can be easily defined for new tasks, leaving the other modules of Fig. 52 unchanged.

Finally, the World State incorporates strategies for: (i) *occlusions* (i.e., after obj_1 has been picked and placed behind a different obj_2, we maintain the knowledge that obj_1 is there, even if it is not visible); (ii) *partial re-planning* (i.e., when a new verbal request is received by the system, we re-initialize the World State, but we take care in not resetting objects that are currently positioned in the robot's hands, keeping that information for the next re-planning).

6.4.5 *Probabilistic Planner*

The planning components of Fig. 52 are:

6.4.5.1 *Action Rules*

List of symbolic rules, or ungrounded actions, where objects are indicated by *obj* and hands are indicated by *hand*[4]. Each rule is defined by: (i) an action symbol, e.g., *graspWith(obj,hand)*; (ii) the necessary pre-conditions to execute the action, e.g., *clear(hand)*, *reachableWith(obj,hand)*; and (iii) a list of possible outcomes with associated probabilities, ordered from most to least likely, summing up to one, e.g., { *inHand(obj,hand)* 1.0 }. In our realization of the system, the probabilities associated with the outcomes (i.e., the effects of the actions) are initialized with either the effect predictions coming from affordance perception or with default values (if no affordance is perceived for that action), and then they can be updated during execution.

Considering our scenario, the Action Rules that we model are: grasping an object, pulling an object towards the robot, pushing an object farther from the robot, putting it on top of another object, and dropping a currently-grasped object. We provide the full specification below. The control routines that implement the motor actions have been developed by other researchers within the iCub community [Pat+10; Tik+13].

Table 16 shows the list of ungrounded Action Rules that we define. When grounding these actions, the symbol *ALL* is expanded to all existing world entities (hands and objects): for instance, if the entities are *LeftHand, RightHand, Tomato, Bread*, then *reachableWith(obj,ALL)* when *obj=Tomato* is expanded to the conjunction of *reachableWith(Tomato,LeftHand)*, *reachableWith(Tomato,RightHand)*, *reachableWith(Tomato,Bread)*. The symbol *OTHERHAND* is expanded to *LeftHand* when *hand=RightHand*, or to *RightHand* when *hand=LeftHand*.

6.4.5.2 *Goal Compiler*

This translates instructions from human language-like format into symbolic robot goals and sub-goals, which the robot can then use to plan its own actions. Human instructions provided by the PRAXICON semantic reasoner come in the form *(object action object)*, e.g., "hand grasp cheese". The Goal Compiler searches the list of ungrounded Action Rules for a rule with a similar symbol (string matching), e.g., *graspWith(obj,hand)*, since "grasp" is common to both the human instruction and the rule present in the ungrounded rule list. Finally, it creates a sub-goal from the most likely outcome of the action, e.g., *graspWith(obj,hand)* $\rightarrow$ *inHand(obj,hand)*. Each sub-goal is obtained by applying the most likely effects of the matching ungrounded rule to the previous sub-goal. This can be seen as the ideal behavior of the system, or as an optimistic default prediction of the effects of the

[4]We use first-order logic syntax, where variables and predicates start with a lowercase letter (e.g., *obj*, *graspWith*), and constants start with an uppercase letter (e.g., *Tomato* is an instance of *obj*).

Table 16: List of Action Rules.

Action symbol	Pre-conditions	Probabilistic outcomes
push(obj,tool,hand)	$\neg isHand(obj), inHand(tool, hand),$ $reachableWith(obj, tool)$	$\neg reachableWith(obj, ALL)$ 0.85 <unpredictable outcomes> 0.15
pull(obj,tool,hand)	$\neg isHand(obj), inHand(tool, hand),$ $reachableWith(obj, tool), pullableWith(obj, tool)$	$reachableWith(obj, ALL)$ 0.85 <unpredictable outcomes> 0.15
graspWith(obj,hand)	$\neg isHand(obj), isClear(hand), isHand(hand),$ $reachableWith(obj, hand), \neg on(ALL, obj),$ $\neg inHand(obj, OTHERHAND)$	$inHand(obj, hand), \neg isClear(hand)$ 0.95 <unpredictable outcomes> 0.05
dropWith(obj,hand)	$\neg isHand(obj), inHand(obj, hand), isHand(hand)$	$\neg inHand(obj, hand), isClear(hand),$ $reachableWith(obj, hand)$ 0.95 <unpredictable outcomes> 0.05
putOnWith($obj_1, obj_2, hand$)	$\neg isHand(obj_1), reachableWith(obj_2, hand),$ $\neg isHand(obj_2), inHand(obj_1, hand), \neg on(ALL, obj_2)$	$on(obj_1, obj_2), \neg inHand(obj_1, hand),$ $isClear(hand)$ 0.7 $\neg inHand(obj_1, hand), isClear(hand)$ 0.15 <unpredictable outcomes> 0.15

robot actions. Such predictions will be made more realistic through the Action Grounding process below.

6.4.5.3 *Action Grounding*

This component generates a list of grounded actions given the objects that are present in the current environment and the possible probabilistic outcomes estimated from affordance perception. This combination is realized by the mapping shown below (on the left side; the right side contains an example):

$A \rightarrow a$	e.g.:	$graspWith($
$O \rightarrow arg_1$		$Bread,$
$T \rightarrow arg_2$		$LeftHand)$
$E \rightarrow$probabilistic outcomes		$\{inHand(Bread, LeftHand)$ 0.9, $\neg inHand(Bread, LeftHand)$ 0.1$\}$

The next best action is predicted with these grounded actions.

6.4.5.4 *Planning Cycle*

This is the main planning loop, where the robot updates its perception of the world, it checks if the current sub-goal or final goal have been met, it plans the next action using the PRADA probabilistic planner engine [LT10], and it executes the planned action with the robot controllers. In Algorithm 1 we show the full pseudocode of the main planning loop used in our architecture, highlighting the *heuristics* that we implement for coping with challenging events, unpredictability of the real world and disturbances that may occur in our considered domain.

6.4.5.4.1ADAPTABILITY HEURISTIC This enables to adjust the previous probabilistic knowledge of the world (acquired by the robot after long-term affordance learning), by operating on it in a short-term local fashion, dynamically *adapting* the knowledge of action success and failure during operation.

We consider the environment model to be Markovian, that is, the effects of an action given the current state are conditionally independent from past states. However, this assumption fails to capture unobserved features of the world that may hinder the success of a given action. For example, let us consider the action *graspWith(Bread,LeftHand)*, with outcome *inHand(Bread,LeftHand)* with an associated default probability of 85%. However, suppose that this action repeatedly fails during one execution of a specific task; this may be caused by noise in the perception of the target object, or by a temporary malfunctioning in the robot hardware or control algorithms. While on the one hand we

Algorithm 1 POETICON++ Planning Cycle

Input: sequence of sub-goals $G = \{g_1, \ldots, g_N\}$
Output: boolean indicating if plan completed or failed
Parameters: initial horizon H_0, maximum horizon $H_{\max}$

$i \leftarrow 1$ ▷ current sub-goal
$h \leftarrow H_0$ ▷ current horizon
while true **do**
  update World State

  if last action produced no observable changes **then**
    apply Adaptability heuristic (24)
  end if

  if $i > 1 \wedge g_{i-1}$ not satisfied **then**
    $i \leftarrow i - 1$ ▷ apply Goal Maintenance heuristic
  end if

  if g_i satisfied **then**
    if $i = N$ **then**
      return true ▷ plan completed
    else
      $i \leftarrow i + 1$ ▷ process next sub-goal
    end if
  else
    $A \leftarrow \text{PRADA}(g_i, h)$ ▷ plan next action A
    if A is valid **then**
      execute A on robot
    else ▷ no plan found
      $h \leftarrow h + 1$ ▷ extend horizon
      if $h \leq H_{\max}$ **then**
        continue ▷ re-plan with increased horizon
      else if $i < N$ **then** ▷ able to skip sub-goal
        $i \leftarrow i + 1$ ▷ apply Creativity heuristic
      else ▷ no more sub-goals
        return false ▷ plan failed
      end if
    end if
  end if
end while

would like the system to adapt to this situation, and avoid repeating the same action again and again if it proves to be not effective, we would not want the default probability to change permanently.

To address this problem, we model the success probability of a given action as a parameter to be estimated, which is initialized with a default value, and it is then temporarily updated during task execution based on a Bayesian estimation approach [Pea88] explained below.

6.4.5.4.2FORMALIZATION OF ADAPTABILITY HEURISTIC The idea behind the Adaptability heuristic is simple: if an action fails, its assumed success probability should be reduced. According to this heuristic, this probability is reduced by a factor σ. Even though this reduction is intuitive, we provide here a theoretical support for this reduction, under a Bayesian estimation framework [Pea88].

We model the success probability of a given grounded action as a parameter to be estimated. It has an initial value, given by the affordances model, and it is updated during task execution, namely whenever an action fails.

Let p_{a,o_k} be the outcome probability of o_k corresponding to a grounded action a. For example, a could be *graspWith(Bread,LeftHand)*, having possible outcomes $p_{a,o_1} = 0.85$ (success), $p_{a,o_2} = 0.15$ (failure). Thus, the parameters to be estimated are these probabilities. Given a grounded action a, let $\Theta_a = [p_{a,o_1}, \ldots, p_{a,o_N}]$ be the vector containing the probabilities of the possible outcomes, where $\sum_i p_{a,o_i} = 1$.

We now want to estimate Θ_a, after performing action a and observing the outcome o_k. Taking a Bayesian estimation approach, the posterior of Θ_a can be written as

$$P(\Theta_a \mid a, o_k) \propto P(o_k \mid a, \Theta_a) P(\Theta_a), \tag{23}$$

assuming that the prior of Θ_a is independent from the performed action a, that is, $P(\Theta_a \mid a) = P(\Theta_a)$. In other words, the knowledge of the action alone does not provide any information about the outcome probability vector Θ_a. The first term of the factorization in (23) is the probability of outcome o_k given the outcome probability vector itself and the grounded action, that is, $P(o_k \mid a, \Theta_a) = p_{a,o_k}$.

Since Θ_a parameterizes a categorical distribution, its conjugate prior is a Dirichlet distribution, parameterized by a vector $[\alpha_1, \ldots, \alpha_N]$ of parameters, such that $p_{a,o_k} = \alpha_k / \sum_i \alpha_i$. Now consider that action a was performed and the observed outcome was o_k. The posterior estimator for p_{a,o_j}, corresponding to the outcome o_j where $j \neq k$, is given by

$$\hat{p}_{a,o_j} = \frac{\alpha_j}{1 + \sum_i \alpha_i} = \frac{\sum_i \alpha_i}{1 + \sum_i \alpha_i} \frac{\alpha_j}{\sum_i \alpha_i} = \sigma p_{a,o_j}, \tag{24}$$

where $\sigma = \frac{\sum_i \alpha_i}{1 + \sum_i \alpha_i}$. Note that $0 < \sigma < 1$ is a factor that depends only on the normalization sum $\sum_i \alpha_i$ of the prior, not on any particular distribution. Thus, we have shown that a geometric decay of the

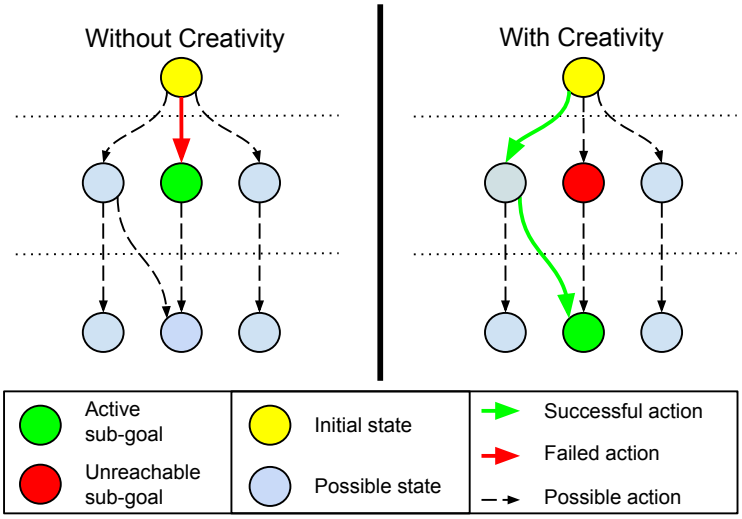


Figure 53: Creativity heuristic. Each row represents a temporal step. When no action is found for a certain sub-goal, and the planning horizon is already too large, this heuristic allows the planner to jump to the next sub-goal and attempt that one instead, bypassing deadlocks.

probability of the outcome o_j can be cast as a Bayesian estimation problem.

In the experimental part of this chapter we set $\sigma = 0.2$, since a strong decay rate was intended, in order to prevent the same action to be tried after it has just failed. In practice, our chosen value implies that the probability of an action drops sharply (we divide it by 5), therefore the system adapts an action for 2 or 3 times maximum before it gives up. This parameter impacts the speed–success trade-off plots (see Sec. 6.5). By lowering its value, we can make the robot try the same action more times: it can be worthwhile in terms of higher **success** in noisy situations, though at the expense of worse **speed**. Note that for $\sigma = 1.0$ we have no adaptation, so this is actually a special case of a general Adaptability update rule.

6.4.5.4.3CREATIVITY HEURISTIC Considering the list of sub-goals created by the Goal Compiler, there might be cases in which forcing the system to necessarily go through all the sub-goals might prevent reaching the final goal (e.g., if one of the sub-goal symbols which is not part of the final goal proves to be not reachable). We want to simplify the planning problem by taking advantage of the sub-goals, yet we want to leave the system a certain flexibility to avoid sub-goals, if it turns out that there are better ways to reach the final goal.

For example, suppose that the robot has to cut a cake, a slicing knife is present but not reachable, whereas a fork is near the robot and reachable. The semantic reasoner suggests to use the knife (being available in the scene, but ignoring geometric constraints). However,

because the knife is not reachable (this is captured by the World State symbolic representation), the sub-goal symbol *inHand(Knife,LeftHand)* fails. Using Creativity, the planner can jump to the next sub-goal, which is *isCut(Cake)*. Then, the agent can achieve this sub-goal using the fork which is near and reachable. More precisely, we allow the system to *jump one step forward in the sub-goal list*, reset the horizon, and replan: see the green arrow in Fig. 53. Since all the information required for the successful completion of the task is present in the final goal, one can jump back and forth in the sub-goal list without loss.

6.4.5.4.4GOAL MAINTENANCE HEURISTIC It may happen that necessary conditions that are met at a certain moment in time are later un-met by accident or because of an external influence, e.g., when an object is removed from a stack. Since the goals are kept in memory after their compilation, we make the system perform a consistency check at each step, verifying that all previous goals are still satisfied, and backtracking when this check fails, allowing the planner to fix the problem first, then continue with the execution. Fig. 54 depicts this idea. The check is performed on the symbols that are common to both the current and previous sub-goal, verifying if they are met. If one of these symbols is not present in the World State (i.e., a symbol has not been met), the system detects that a previous sub-goal has been undone, and it jumps back by one step in the sub-goal list.

6.4.6 *Simulated Symbolic Reasoner*

For this case study, we developed a simulator to investigate the system robustness at a deep, quantitative level, with a focus on robustness to robot noise (see Sec. 6.3). We now explain how this noise interplays with the planner operation, and we introduce some metrics to assess our experiments.

6.4.6.1 *Noise*

It accounts for all possible hard-to-identify and hard-to-model causes of robot action failures in the manipulation setting and action types that we consider, for example: robot miscalibration due to mechanical backlash in the joints, inaccuracies in the motor control, noise in the visual perception of the objects' positions. Within a simulated *episode* (experiment), we model the noise probability associated to each capability (e.g., left arm manipulation capability, right arm manipulation capability, visual recognition capability) by setting a *noise level* threshold, corresponding to the capability under interest, to a fixed discrete value ranging from 0.0 (perfectly reliable) to 0.95 (very unreliable).

At the beginning of a simulated episode we fix the *initial environment* conditions of objects available to the agent, in the form of one

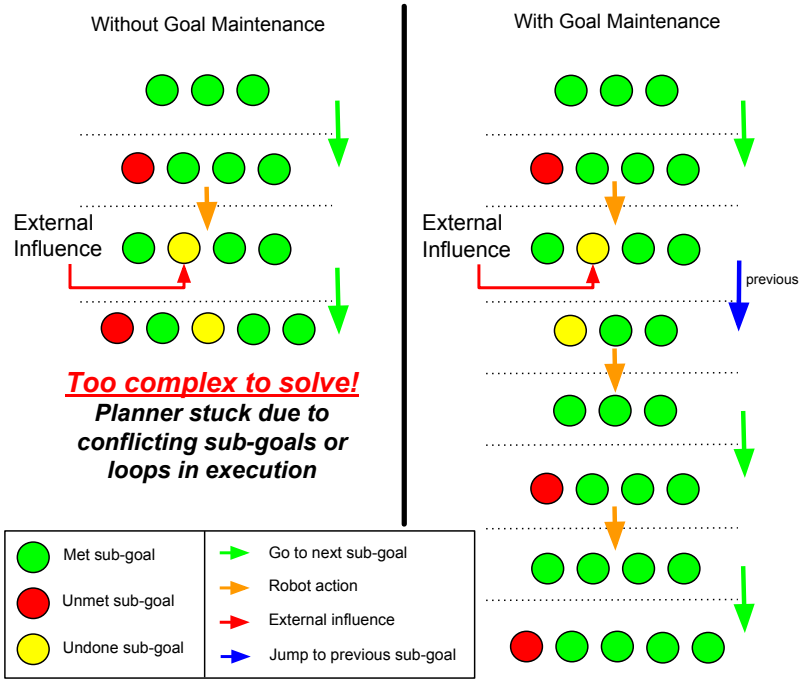


Figure 54: Goal Maintenance heuristic. Each row represents a temporal step. At each planning step, a check is performed on the symbols that are common to the current and previous sub-goals. If one of those symbols fails the check, the planner backtracks along its sub-goal list until this check passes, and re-plans from there.

image (recorded with the real robot cameras), and the perceived coordinates, object recognition labels and other World State properties of each object, as well as previously-learned affordance knowledge. This fixed initial environment makes the visual, geometric and affordance information at the beginning of an episode controllable by the experimenter, thus easily reproducible. Then, we vary the value of the robot noise level and/or the possibility of using the Adaptability and Creativity heuristics by the planner[5], and we repeat the process. When we have run a significant number of episodes (i.e., 20+ repetitions) for each configuration of the system (i.e., values of the noise levels, activations or not of the heuristics), we extract statistics from that set of episodes.

When the planning system requests the execution of an action from the robot, *we simulate the outcome of the action stochastically*, by extracting a random uniform number and comparing it against the previously-fixed robot noise level: if the random number is above the noise level threshold, we will consider the action successful (as a result, the World State will change); otherwise, we will declare it unsuccessful. Note that this simulator only affects the low-level robot components in the bottom right of Fig. 52: the rest of the system, in particular the Probabilistic Planning, carries on with its strategies and action selection, as if it were operating with the real robot.

6.4.6.2 *Metrics to Characterize One Episode*

At the end of one episode, we save the following pieces of information: (i) #good: number of motor actions which succeeded; (ii) #total: number of total attempted motor actions, including failed ones; (iii) **success**: boolean indicating whether the planner attained the goal or not:

$$\textbf{success} = \begin{cases} \top & \text{planner achieves goal} \\ \bot & \text{otherwise,} \end{cases}$$

being allowed at most t total motor actions in a plan, where $t > 1$ is a data-derived constant. We use $t = 50$, obtained by multiplying the number of consecutive repetitions that we want to permit for each action of a plan (between 8 and 10), times the ground-truth number of actions necessary to reach the final goal from the considered initial conditions (4 in the simple scenario, 7 in the complex scenario, see Sec. 6.5).

For example, an episode with $\{\#\text{good} = 5, \#\text{total} = 8, \textbf{success} = \top\}$ represents that the system commanded 5 motor actions which suc-

[5] We did not test the Goal Maintenance heuristic in the simulator (i.e., we did not implement the possibility of an object being moved in the world by an external agent) because, from our qualitative tests (see Sec. 6.5.1), we already deem this as the heuristic with the clearest impact. Either our system is able to detect and deal with external changes (i.e., Goal Maintenance is on), or it does not have that capability and in the latter case it simply fails consistently.

ceeded, it attempted 8 actions in total, and it achieved the final goal autonomously.

We define **speed** as a quantity that captures how fast the planning process is at finishing an experimental episode, regardless of the final outcome. The rationale is that sometimes it is important for a robotic system to be fast in reporting a failure. For example, in our scenario, we favor robot speed over success, because it is undesirable to keep trying the same noisy robot action over and over, and we accept to fail at times (the robot can ask the human for help). The **speed** metric is a function to the #total counter introduced above:

$$\textbf{speed} = \max\left(1 - \frac{\#\text{total}}{t}, \quad 0\right), \tag{25}$$

where $\#\text{total}/t$ is a penalty term, and t makes the fraction be less than one (in our case, each extra action attempt incurs a cost of making the speed metric decrease by $1/50$). Basically, $\textbf{speed} \simeq 1$ corresponds to an episode with high speed (low number of total attempted motor actions), whereas $\textbf{speed} \simeq 0$ is a very slow episode (high number of total actions).

6.4.6.3 *Metrics to Characterize a Set of Episodes*

Recall how for each simulated configuration we repeat the process 20+ times (number of episodes in a set). We are interested in analyzing the behavior of the system in that configuration according to the two statistical metrics **success** and **speed**, averaged over the number of episodes. Both metrics have a range between 0 and 1: (i) the **average success** is the fraction of episodes within the set where the system managed to complete the goal (i.e., with $\textbf{success} = \top$). In a sense, this metric expresses the difficulty of one experimental configuration (i.e., initial condition, noise levels, heuristics activations) from the point of view of the system. For example, an average success of 0.9 indicates that the system managed to complete the goal in 90% of the episodes pertaining to the current experimental configuration; (ii) the **average speed** expresses how fast the planning process is at completing the experiment on average, regardless of the final outcomes.

6.5 RESULTS

In this section, we evaluate our full system on the iCub humanoid robot (see Sec. 3.1) in a sandwich making scenario as the one of Fig. 55. We present (i) qualitative results observed by our implementation and tests on the real iCub robot platform in Sec. 6.5.1, then (ii) a quantitative analysis obtained by studying the response of our system when we simulate noisy robot action failures in Sec. 6.5.2. Lastly, (iii) we report statistics about the main contributors to the POETICON++ project code repository in Sec. 6.5.3.

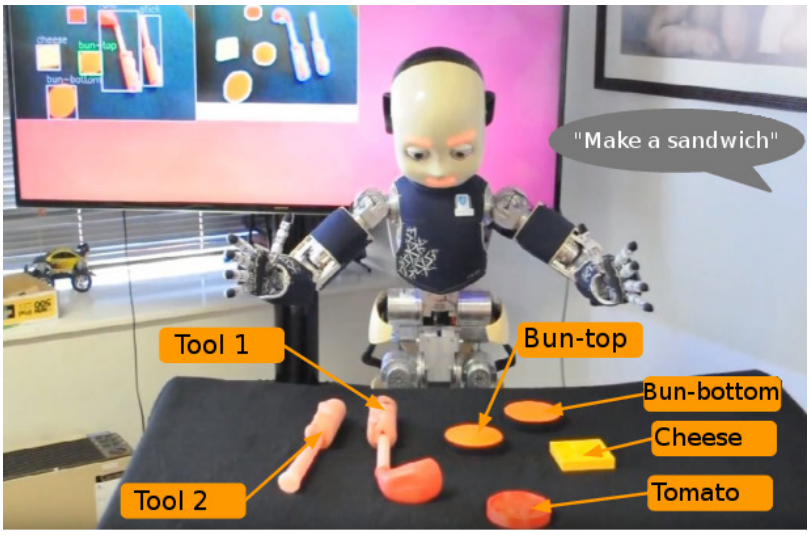


Figure 55: Example of the initial state of the sandwich making problem. The instruction provided by the human is reported in the gray balloon, the objects are annotated with orange boxes, and the output of the visual perception routines is shown in the back screen. Note that perceiving the affordances of tools does not require their names.

(a) (b) (c) (d) (e)

Figure 56: Temporal snapshots of the robot during the Sabotaged Plan qualitative example (see Sec. 6.5.1.2).

6.5.1 *Qualitative Results*

We now list qualitative behaviors and capabilities that we implemented on the real iCub robot.

6.5.1.1 *Object Out of Reach*

When the robot cannot grasp an object with its own end effector, it can bring that object closer by using elongated tools (see Fig. 57b). In addition, in the presence of more than one tool it can select the most appropriate one by querying its previously learned tool–object affordances knowledge base (see Sec. 6.4.3).

6.5.1.2 *Sabotaged Plan*

During human–robot shared tasks, the robot is able to monitor human intervention and exploit it towards the realization of the assigned goal.

Namely, with its visual perception the robot can detect if a human *sabotages* the robot plan execution by removing ingredients that have been already assembled for the sandwich, such as the tomato in the Fig. 56 example. The Goal Maintenance heuristic (see Sec. 6.4.5.4.4) detects that the sub-goal *on(Tomato,Cheese)* no longer holds, so it helps the planner by *backtracking*, fixing the goals that have been sabotaged, and going *forward* with the plan.

6.5.1.3 *Robot Action Failure*

The system can detect problems caused by the robot itself due to miscalibrations and unexpected events, such as an object falling from the robot hand after it had been grasped correctly. This can be detected with robot proprioception (measuring the weight at the end effector) or with vision (classifying between full hand and empty hand). The Adaptability heuristic (see Sec. 6.4.5.4.1) can only help to a limited extent, by repeatedly reducing the probability of the failing action, for example *inHand(Cheese,LeftHand)*. However, the Creativity heuristic (see Sec. 6.4.5.4.3) avoids a deadlock, by instead focusing on the underlying goal *on(Cheese,Bun-bottom)*, and choosing the right hand to achieve it anyway: *graspWith(Cheese,RightHand)*; *putOnWith(Cheese,Bun-bottom,RightHand)*, where the predicate *putOnWith*(obj_1, obj_2, h) means "put object obj_1 on object obj_2 with hand h".

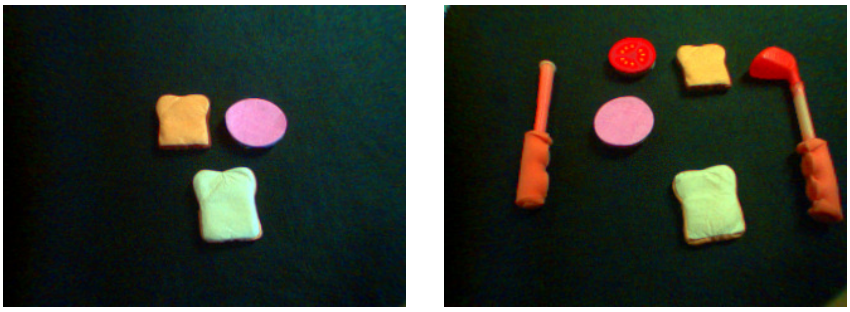
6.5.1.4 *Semantic Change*

In some cases, objects might go missing during the plan execution. For example, they can fall on the ground or disappear from the field of view. As long as the problem object is not part of the final goal, the probabilistic planner can still find a solution. Otherwise, it *reports the failure* back to the high-level planning (i.e., to the human user) and awaits further instructions.

6.5.2 *Quantitative Results*

In Sec. 6.5.1, we presented qualitative tests in a sandwich making scenario like the one of Fig. 55, showcasing the capability of the system to react to objects being far away from the robot, or being removed due to external intervention (sabotage), or disappearing from the scene entirely. In the present section, we evaluate our system *quantitatively*, with a detailed analysis in simulation obtained by studying the response of our system when we reproduce different types of robot failures and we activate our proposed heuristics.

We report experiments in different scenarios with two possible initial conditions, shown in Fig. 57. In both cases a human user asks the robot to "make a ham sandwich", and the semantic reasoner translates such a request in the following sequence of instructions: "hand grasp



(a) Simple scenario. (b) Complex scenario.

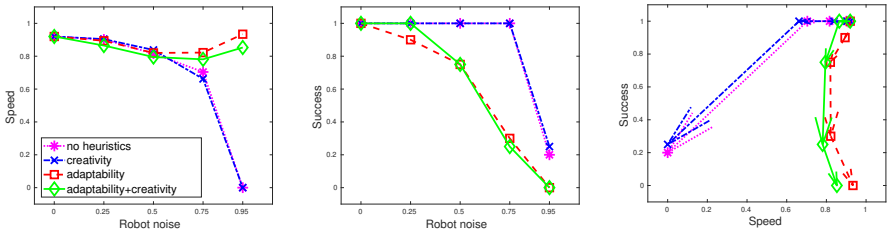
Figure 57: Possible initial conditions of the quantitative POETICON++ evaluation. (a): simple scenario with three objects available and reachable by the robot. (b): complex scenario with six objects, some of them not directly reachable by the robot.

Ham, Ham reach Bun-bottom, hand put Ham, hand grasp Bun-top, Bun-top reach Ham, hand put Bun-top". The PRAXICON language reasoner generates those instructions by taking into consideration the semantics of the available objects (i.e., the labels recognized from object recognition, and their semantic relationship with the "ham sandwich"), excluding all environmental position constraints and the actual capabilities of the robot (e.g., the probability of success of the robot actions); these additional aspects are accounted for by affordance perception and probabilistic planning.

Our experiments are evaluated with the **success** and **speed** metrics. Note that, in a collaborative scenario, when the system reports a failure (**success** $= \perp$, i.e., impossibility to achieve the final assembly goal autonomously) it is not necessarily bad, as long as the **speed** is kept at an acceptable value. A failure means that the system asks for external help (e.g., from the human, or with another query to the semantic reasoner given the partially-completed plan). This is shown in the speed–success trade-off plots described next, and further discussed in the formalization of the Adaptability heuristic in Sec. 6.4.5.4.2. In the plots, each colored line corresponds to a system configuration, and it contains five markers corresponding to different levels of noise.

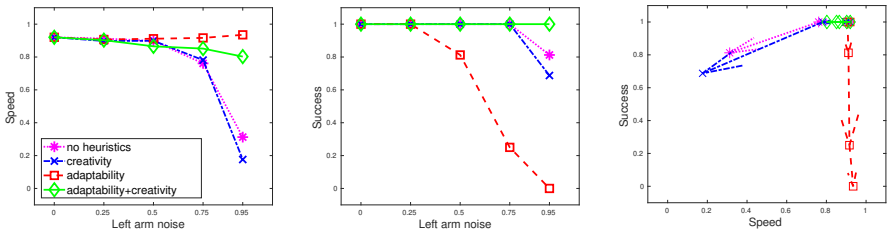
6.5.2.1 *Simple Scenario with Equal Arms Noise Level*

We first validate our architecture on a simple scenario which includes three objects, all within reach of the robot, as shown in Fig. 57a. Our system has to translate the semantic instructions to symbols, actions and goals usable by the robot. In this case, the semantic instructions are fairly similar to the planner ones, the only significant difference being the choice of the hand (instantiation of *hand*, which can be *LeftHand* or *RightHand*). Depending on the simulated robot noise and on



(a) Speed response. (b) Success response. (c) Speed–success trade-off.

Figure 58: Response of the system in the *simple scenario* when varying the *robot noise equally for both arms*, and activating the different planner heuristics.



(a) Speed response. (b) Success response. (c) Speed–success trade-off.

Figure 59: Response of the system in the *simple scenario* when varying the *left arm noise*, keeping the right arm noise constant at 0.25, and activating the different planner heuristics.

the type of planning strategy (no heuristics; with or without Adaptability and Creativity), the behavior of the system changes as follows.

Without any heuristic (magenta dotted line in Fig. 58), as we introduce more noise, **speed** decreases (i.e., it requests a high number of #total actions to reach the goal), and **success** is constantly 1 (i.e., $\top$) for noise ≤ 0.75. The absence of Adaptability implies that failing actions can be executed again, up to $t = 50$ #total motor actions within each episode. Despite the fact that the agent almost consistently manages to achieve the goal (**success** decreases only for very elevated noise), this result is disappointing from the point of view of the **speed**. Intuitively, it is not desirable that the robot try the same physical action over and over until it eventually succeeds, at the cost of considerable time wasted. The same considerations hold for the case with the Creativity heuristic (blue dash-dotted line in Fig. 58), again due the absence of Adaptability (however, the influence of Creativity would be higher if the robot noise affected only one arm instead of both arms, as we will see in the experiment of Sec. 6.5.2.2).

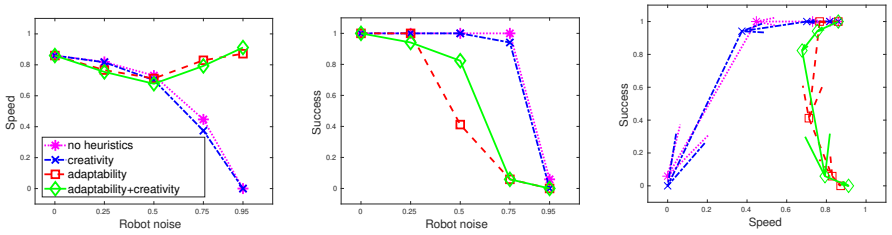
When we enable the Adaptability heuristic (red dashed line in Fig. 58), or Adaptability in conjunction with Creativity (green solid line), as we introduce more noise, **speed** is approximately constant (constant #total attempted actions), however **success** decreases. As the noise is increased, we observe some cases where the system *fails* to achieve the goal (lower **average success**). This is preferable to the no-heuristics and the Creativity cases, considering that #total is now lower, and the system can ask for external help to reach the final goal.

To summarize, when the noise is elevated we appreciate the advantage introduced by the heuristics, allowing the system to acknowledge the infeasibility of the goal when a motor action fails repeatedly, and react to it by asking for external help. This effect is more pronounced in the **speed** metric (see Fig. 58a), which, as defined in Eq. 25, is inversely proportional to the #total counter of attempted motor actions. Fig. 58c shows the trade-off between the two metrics.

6.5.2.2 *Simple Scenario with Unequal Arms Noise Level*

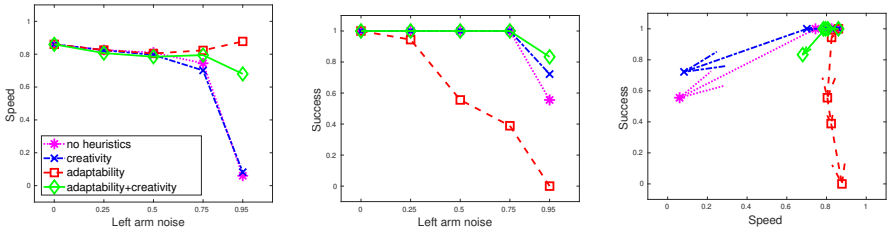
In this case we start from the simple scenario and we vary the *left arm noise*, keeping the right arm noise constant at 0.25. Fig. 59 shows the response of the system. The no-heuristics configuration (magenta dotted line) results in the **speed** metric decreasing quickly, whereas **success** decreases only at the very end (highest left arm noise). When we enable the Creativity heuristic (blue dash-dotted line) the response is similar, because Creativity alone is not effective in reacting to elevated noise in this case.

In the Adaptability case (red dashed line), **speed** does not decrease, but **success** does, and rather abruptly at that. Adaptability alone is not sufficient, in this case, because the probability of the failing action is penalized (e.g., *graspWith(Ham,LeftHand)*), but the planner



(a) Speed response. (b) Success response. (c) Speed–success trade-off.

Figure 60: Response of the system in the *complex scenario* when varying the *robot noise equally for both arms*, and activating the different planner heuristics.



(a) Speed response. (b) Success response. (c) Speed–success trade-off.

Figure 61: Response of the system in the *complex scenario* when varying the *left arm noise*, keeping the right arm noise constant at 0.25, and activating the different planner heuristics.

is not able to jump to the next sub-goal (e.g., choosing the sub-goal *on(Ham,Bread)* instead of *inHand(Ham,LeftHand)*).

Finally, in the Adaptability+Creativity configuration (green solid line): neither the robot **speed** nor the **success** decrease much. In particular, in Fig. 59c we see that the performance always remains in the optimal region in the top-right. Even for the most challenging left noise value, typical episodes are $\{\#\text{good} = 4, \#\text{total} = 9, \textbf{success} = \top\}$, meaning that if some left arm action fails for a few times, the planner quickly penalizes that action (Adaptability), and it is then able to jump to the next sub-goal, using the other arm (Creativity).

6.5.2.3 *Complex Scenario with Equal Arms Noise Level*

We now test the system on a complex scenario which includes six objects, some of which not within the direct reach of the robot, as shown in Fig. 57b. This time the planner has to generate the sequence of motor commands necessary to reach faraway objects. This involves reasoning about the affordances offered by available tools, grasping a tool, and using it to draw the target object closer. Note that, without resorting to our tool affordance model, this scenario would be always unfeasible due to the geometrical disposition of certain objects.

We first run our baseline system with no heuristics. The results can be inspected in the magenta dotted line of Fig. 60. In this challenging scenario, when there is zero noise, the agent needs 7 motor actions to accomplish the plan. Usually it manages to achieve it even in the presence of noise because, as in the simple scenario, these settings allow the system to retry failed actions, as long as the #total action counter is at most $t = 50$ (however, this effect gets penalized in the **speed** metric, which decreases faster than in the simple case, as can be seen in Fig. 60a when noise ≥ 0.75). When we enable the Creativity heuristic (blue dash-dotted line), the response corresponding to increasing noise is similar to the no-heuristics case.

Next, we introduce the Adaptability heuristic (dashed red line) and then the Aptability+Creativity configuration (solid green line). We should note that, in general, the architecture manages to use the tool affordances correctly, selecting the Hook tool rather than the Stick one in order to perform a pulling action successfully (to draw faraway objects closer to the robot workspace). Besides, we can observe that when we introduce a considerable degree of noise (0.5), the **success** drops (in the Creativity configuration): less than 50% of the episodes complete the plan, the majority report a failure thus requesting external help. When the noise is even worse, **success** drops to zero: however, the corresponding #total number of attempted motor actions is kept low, minimizing the wasted time and yielding the possibility of completing the plan with further help (e.g., with the help of the human, or with another query to the semantic reasoner given the partially-completed plan). Enabling Adaptability and introducing noise, the system's **speed** stays good (a consequence of low #total attempted actions upon realization of failure), whereas **success** decreases. The Adaptability and the Adaptability+Creativity configurations are similar, except when noise = 0.5, where Adaptability+Creativity fares better. Fig. 60c shows the trade-off between the two metrics.

6.5.2.4 *Complex Scenario with Unequal Arms Noise Level*

In this case we start from the complex scenario and we vary the *left arm noise*, keeping the right arm noise constant at 0.25. Fig. 61 shows the response of the system. The Adaptability+Creativity configuration outperforms the other settings. This is visible in terms of **speed** (Fig. 61a) when the left arm noise is > 0.75, and in terms of **success** (Fig. 61b) when the left arm noise is > 0.5. For comparison, Adaptability setting has a very good **speed**, to the detriment of **success**.

Fig. 61c shows that, in this challenging experiment, the performance of our system with heuristics always remains in the optimal region in the top-right. With the worst left noise value, on average the experimental episodes result in $\{\#\text{good} = 9, \#\text{total} = 16, \textbf{success} = 0.83\}$, meaning that if some left arm action fails for a few times, the planner quickly penalizes that action (Adaptability), and it is then able to jump to

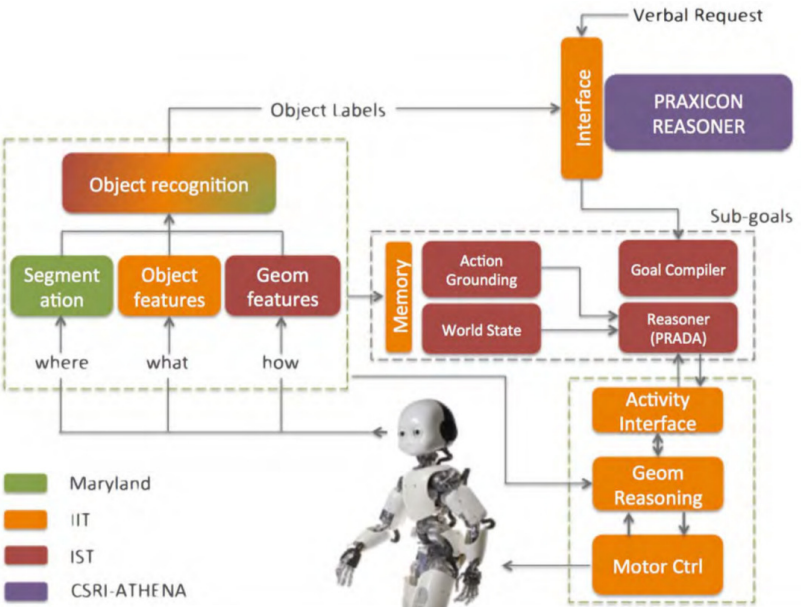


Figure 62: POETICON++ project software architecture. The contributions by IST are marked in red.

the next sub-goal, using the other arm (Creativity) and being able to achieve the final goal autonomously 83% of the times.

With the above experiments, we have presented a quantitative evaluation which demonstrates that the combination of affordance perception with probabilistic planning and the use of planning heuristics permits to deal with high levels of noise.

6.5.3 *Contributions to Code Repository*

Fig. 62 shows the final software architecture of the POETICON++ project. The contributions by IST are marked in red and can be further divided conceptually into: (i) geometrical features extractor for tool use affordances; (ii) world state memory; (iii) planning. Giovanni Saponaro wrote the first two components, ran the quantitative tests of the third one (see Sec. 6.5.2), and integrated all the modules together.

Fig. 63 shows the top contributors of the POETICON++ open source project repository. Giovanni Saponaro is #1 contributor in terms of number of commits, with 38.92% of them (~316 out of ~882 total). All users had commit streaks during the periods of February–March 2015 and February–March 2016, coincident with the two last review meetings and demonstrations of the project. If we account for actual Lines of Code (LoC) excluding binary files, Giovanni Saponaro had the most contributions with over 65 000 lines.

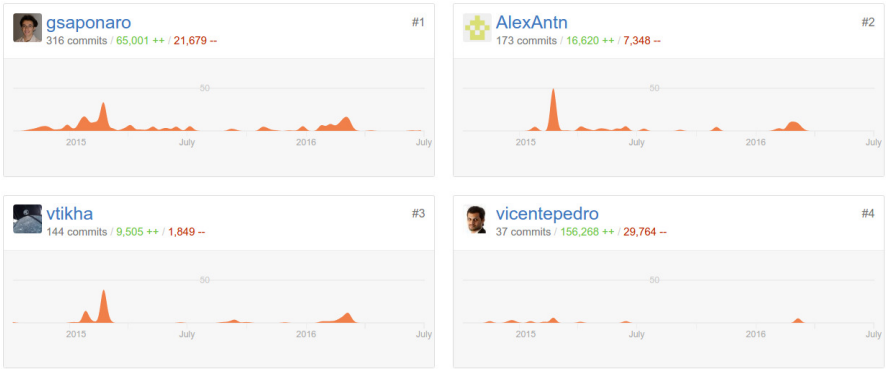


Figure 63: Top contributors of the POETICON++ project repository (see footnote [2] on p. 110). Giovanni Saponaro is #1 contributor in terms of number of commits.

6.6 CONCLUSIONS AND FUTURE WORK

This chapter presented a case study about the POETICON++ project scientific achievements, leveraging robot perception of affordances, world modeling and probabilistic planning, and supporting human–robot collaboration behaviors.

We have described a cognitive architecture that supports action selection and complex robot manipulative task execution under challenging conditions in unstructured environments. We combine affordance perception, world modeling and probabilistic planning to ground the semantic action plans to a robotic representation that is suitable for problem solving. We introduce some heuristics that make the system robust to failures during task execution.

In terms of results, (i) we show qualitative tests on a real robot in a number of situations encountered and modeled during the POETICON++ project; (ii) we perform a quantitative evaluation of the system, modulating the level of noise and the use of planning heuristics. We publicly release the code that implements our system on the iCub robot, however several software modules of our architecture can be used in other applications and platforms that include manipulative tasks. Our release includes a simulated symbolic reasoner for validating the probabilistic planner under challenging conditions, and real robot sensorimotor data used for affordance perception.

As future work, we foresee two main aspects. First, making the system more generic by having the robot learn the Action Rules of the domain autonomously. Second, improving the social robot behavior aspect with further human-in-the-loop sophistication, for example by monitoring the consequences of helpful human actions in a shared human–robot goal, and having the robot ask for confirmations (e.g., “did you just place this object on the table?”).

7 FINAL REMARKS

The core *goal* of this thesis was to develop reasoning algorithms for robots, such that they would be able to operate in unstructured human environments and deal with elements not seen during training, under certain assumptions. To this end, the *approach* that we followed for this work was to develop and test computational models that account for perceived object affordances (i.e., action possibilities) and other environment elements contingent on the scenario being studied (e.g, manipulative gestures, tool use, planning of complex motor tasks under noise). The main *conclusion* is that this type of affordance-based models is indeed a powerful means for service robotics, providing interesting results without the burden of large robot sensorimotor data acquisition, yet still offering the flexibility of a well-founded probabilistic framework that can be inspected and engineered.

For pursuing the above-mentioned goal we made several contributions, summarized in Sec. 7.1. Finally, Sec. 7.2 brings to light some existing limitations in the proposed work, and potential future directions for tackling them.

7.1 MAIN CONTRIBUTIONS

There are four main contributions of this thesis.

First, in Sec. 3.3 of Ch. 3 we illustrated a software framework for visual robot affordance learning. It is based on an original implementation by Montesano (see [Mon+08] and Sec. 2.2.1). Our contribution was to extend this framework for making it *modular* and versatile. In particular, we integrated this framework with modern versions of the C++ language and of the Yet Another Robot Platform (YARP) middleware (see Sec. 3.1) that support research with the iCub humanoid robot. We added the possibility of extracting additional shape descriptors, which users can select depending on the experimental context (e.g., convexity defects for experiments involving hands and tools). This framework now supports real time operations with robot sensors: e.g., 30 Hz color cameras with 640×480 pixel resolution, shape extraction with an i7-4700MQ processor and 16 GB of memory. We made this framework publicly available: see footnote [5] on p. 43.

Second, in Ch. 4 we proposed a computational model that combines object affordances with communication (gestures and language). We do this by contributing a gesture recognizer for manipulative hand gestures (detailed in Appendix A), and embedding this gesture recognizer into a computational model of object affordances and word meanings,

previously proposed by Salvi (see [Sal+12], Sec. 2.2.2 and Sec. 4.2.1). The resulting model allows a robot to interpret and describe the actions of human agents by reusing the robot's previous experience. In doing so, the robot shifts from reasoning in an ego-centric manner to reasoning about actions performed by external human users, thus being social to some extent. Our model can be used flexibly to do inference on variables that characterize an environment (e.g., to do prediction, belief revision, early action recognition). In addition, by reasoning on the probabilities of words, the model shows the emergence of semantic language properties. We made this model publicly available: see footnote [3] on p. 50.

Third, in Ch. 5 we proposed a computational model of affordances that involves multiple objects and gives rise to tool use, which is a desirable skill for having robots operate in complex manipulation tasks that are typical of unstructured environments. In particular, for this model we contribute a visual feature extraction component that processes multiple objects in their entirety and in their sub-parts. In the learning phase, we evaluate various types of computational affordance models and parameters for assorted tasks (e.g., generalization to unseen objects; transfer of learned knowledge from a simulated robot to a real one). In addition, we contribute a method for learning the affordances of robot hand postures (i.e., different apertures of the fingers). This allows to investigate the developmental link from hand affordances (i.e., action possibilities by using the hands) to tool affordances (i.e., action possibilities by using the hands). Our dataset of hand posture affordances is made publicly available: see footnote [4] on p. 99.

Fourth, in Ch. 6 we reported a case study about the application of the ideas presented in the previous chapters (namely affordances, language, and tool use). These are used for supporting robot planning of manipulation tasks, developed in the context of the POETICON++ research project. We illustrate a robust problem solving system that combines affordances with symbolic reasoning probabilistically. This system is capable of dealing with uncertainty, using heuristics that allow the robot to adapt to the current situation, and to find creative solutions for a task given by a human person via verbal instructions. We made this system publicly available: see footnote [2] on p. 110. We contribute much of the code, the integration and testing of all components under several conditions, and a novel simulated symbolic reasoner for validating the probabilistic action planner under challenging conditions.

7.2 LIMITATIONS AND FUTURE WORK

We now discuss current limitations, and we provide possible research directions pertaining to the topics of the thesis.

7.2.1 *Restricted Scenarios*

Collecting large amounts of robot sensorimotor data (with real robots) is challenging and costly. This is one reason for the scenarios considered in this thesis being restricted (see Sec. 3.2).

These scenarios are simple in the sense that they consider a humanoid robot in a fixed position (being able to move its torso, arms and head), next to a table (with a known height) that has a few objects on top, with some visual perception algorithms available, and a limited repertoire of three, pre-defined *motor actions* (which can be exerted on any reachable part of the table).

Nevertheless, using simple scenarios is adequate to explore the key concepts touched by this thesis, for instance to make experiments feeding real robot sensory data into the computational models that we propose, and making inferences on such models.

In addition, the experiments in Ch. 4 include the presence of a human in front of the robot and the table, however the human is subject to similar constraints to the robot; in the language part of that chapter, a vocabulary of about 50 *words* is considered for grounding the language in the robot sensorimotor experience, and for describing the experiments verbally. Still, it would be desirable to have a richer sets of concepts (e.g., actions and words), making our model more *scalable*. This can be done either by using large amounts of data [Lev+18], or by devising machine learning methods that can generalize efficiently from very few observations. With regard to the number of words in Ch. 4, it would be useful to make the model extract syntactic information from the observed data autonomously, relaxing the current bag-of-words assumption.

Furthermore, it would be desirable to have a robot affordance model that is able to extract features autonomously (end-to-end learning) instead of using pre-specified, engineered features suited to the particular tasks studied in this thesis. Some works already explore this possibility [Deh+16b; Deh+17].

7.2.2 *Notion of Action*

In this thesis, we have considered the motor action to be a discrete symbol, within our computational models. This is a limitation because, in the real world, the space of actions is continuous and with complex dynamics. The issue of action representation in robotics has drawn ample attention: a recent survey about it is [Zec+19]. For example, many researchers have resorted to the concept of Dynamic Movement Primitives (DMPs) [Sch06], used for trajectory control and planning. It allows to learn from example trajectories, and to generate approximate full or partial trajectories from starting and final points. It would be

interesting to incorporate this concept in the transfer or action from human to robot, a concept that we explored in Ch. 4.

7.2.3 *Action Anticipation*

The ability to foresee the action performed by other agents onto physical objects is fundamental for successful action recognition and anticipation, therefore for social interaction too. We started exploring this aspect in Sec. 4.4.1.4 by merging the information from human body gestures, recognized with an algorithm based on Hidden Markov Models (HMMs), with the information from affordances. One way to extend this is to employ Recurrent Neural Networks (RNNs), which can express more complex dynamics than HMMs (e.g., dependencies between states that are far in time; continuous instead of discrete states), thus permitting the prediction of multiple and variable-length action sequences in the future [Sch+18]. Another promising avenue for research on action anticipation is that of include eyes and gaze into the estimation model. Eye cues give additional information (earlier information), besides body joints, during human–robot collaboration [Dua+18].

7.2.4 *3D Perception*

The visual perception algorithms adopted in this thesis are based on 2D data. It would be profitable to augment it with 3D information. On the iCub robot, this has been done, for example with stereo vision [Fan+14; MTN18]. However, this technique suffers from issues such as miscalibration of the robot cameras during head and eye movements. In addition, acquiring good quality 3D data of thin objects (such as the sandwich ingredients used in Ch. 6, approximately 1 cm thick) located on a tabletop at a short distance from the robot (whose stereo eyes have a short baseline distance between themselves), remains a challenging problem, manifested with noisy disparity maps.

A

GESTURE RECOGNITION MODEL

This appendix describes a human gesture recognition model for spotting manipulative hand gestures, inspired by statistical techniques from Automatic Speech Recognition (ASR). This model was used in Ch. 4, as one of the blocks of the system described therein.

In this appendix, we adopt the following *terminology*. We use the word *uninterrupted* to refer to a sequence without temporal breaks in between (e.g., an uninterrupted sequence of gestures). We use the word *continuous* to refer to mathematical real number objects (e.g., the continuous probability between 0 and 1 associated to the output of a gesture recognition algorithm).

This appendix is the subject of the following publication:

- Giovanni Saponaro, Giampiero Salvi, and Alexandre Bernardino. "Robot Anticipation of Human Intentions through Continuous Gesture Recognition". In: *International Conference on Collaboration Technologies and Systems*. International Workshop on Collaborative Robots and Human–Robot Interaction. 2013, pp. 218–225. DOI: 10.1109/CTS.2013.6567232.

The outline of this appendix is as follows. Sec. A.1 gives the motivation for building a recognizer of hand gestures in cognitive systems, as well as related work from the literature. Sec. A.2 provides our proposed approach, including different models that were considered for addressing specific issues. Sec. A.3 lists the gesture recognition results, and finally Sec. A.4 contains our conclusions.

A.1 BACKGROUND AND RELATED WORK

Gesture recognition is an important area of research in pattern analysis [AR11], with applications in diverse fields, including biometrics, surveillance, health and assistive technologies, as well as human–computer interaction [WW11; RA15] and human–robot interaction [WRT00; YPL07; BWB08].

Several approaches have been proposed for allowing users of artificial systems to employ body gestures for expressing feelings and communicating their thoughts, in the fields of human–computer interaction [WW11; RA15] and human–robot interaction [WRT00; YPL07; BWB08]. In particular, these fields have seen a surge of interest in interfaces whereby users perform sequences of *uninterrupted* (therefore natural) physical movements with their hands, body and fingers to interact with smartphones, game consoles, kiosks, desktop computer screens and

more. Therefore, it is important to develop pattern analysis techniques suited for recognizing physical gestures and motion in the context of human–robot collaboration [Kan+03; DLS13; DS14; Dra+15].

We now overview specific works about gesture recognition.

The nature of human gestures is ambiguous and context-dependent [McN96; MC99; KAC17]: there exist many-to-one mappings between gestures and conveyed concepts, making gesture recognition a difficult problem. In addition, in the action aspect, the same gesture can serve different purposes depending on the acted object: for example, the action of pointing and the action of pressing a button are realized by a similar gesture, but they are distinct because of different affected objects. As such, the ambiguity in mappings between gestures and concepts is also one-to-many.

Different approaches have been proposed to design automatic gesture recognition systems, both to decide which *features* are salient for recognition [Cam+96] and which *model* best classifies them. For a comprehensive reviews of these systems, we refer the reader to [WH99; MA07; KAA11]. In particular, designing a recognizer for dynamic gestures (see footnote [5] on p. 51) poses two main issues:

1. spatio-temporal variability: the same physical gesture can differ in shape and duration, even for the same gesturer;

2. segmentation: the start and end points of a gesture are difficult to define and identify.

Dynamic gestures are essentially a manifestation of body movement, therefore the *high-level features* to recognize them are also related to motion: positions, velocities, accelerations, angles of body joints (including fingers). These are the features employed in existing gesture recognition solutions based on machine learning, such as Microsoft Visual Gesture Builder for Kinect[1], or the Gesture Recognition Toolkit (GRT)[2].

The high-level features described above arise from pre-processing algorithms applied to the raw data captured by vision sensors. We refer to this kind of data and to the associated signal processing techniques as *low-level features*, which are commonly: skin color segmentation, optical flow (the apparent visual motion caused by the relative motion of objects and viewer), arm–hand tracking in 2D or 3D, full body tracking.

Many gesture recognition systems are designed to work in a controlled environment, or they make strong assumptions:

- limited and fixed lexicon of permitted gestures;

- availability of the whole test data sequence to classify (system only works offline);

[1] `https://developer.microsoft.com/en-us/windows/kinect`, `https://channel9.msdn.com/Blogs/k4wdev`

[2] `http://www.nickgillian.com/grt/`

- constrained physical space (hands must move only within a certain region of upper body);
- unnatural interaction (isolated gestures, to be preceded and followed by a relaxed pose lasting several seconds);
- users must wear hardware tracking devices, which can be impractical and expensive.

Our gesture recognition model is loosely inspired by neuroscience in the following sense. Neuroscience experiments have suggested that the area of the human brain responsible for gesture processing is also employed for speech processing [Xu+09], functioning in fact as a modality-independent semiotic system, connecting meaning to various types of symbols: words, gestures, images, sounds, or objects. The ability to understand and interpret our peers has also been studied in psychology, focusing on internal simulations and re-enactments of previous experiences [SLH12; Bil+16].

From Sec. 1.4.2, recall that mirror neurons are visuomotor neurons that respond to action and object interaction, both when the agent acts and when it observes the same action performed by others, hence the name "mirror".

In applying the mirror neuron theory in robotics, as we and others do [Gaz+07; Lop+09], an agent can first acquire knowledge by sensing and self-exploring its surrounding environment (see Sec. 3.2). Afterwards, it can employ that learned knowledge to novel observations of another agent (e.g., a human person) who performs similar physical actions to the ones executed during prior training. In particular, when the two interacting agents are a caregiver and an infant, the mechanism is called *parental scaffolding*, having been implemented on robots too [Ugu+15a; Ugu+15b]. These works tackle a problem that is crucial to (artificial) imitation: how to map action sequences observed in an external agent to action sequences performed by an imitator agent, which in general may have different affordances and a different body morphology (this issue is also known as the *correspondence problem* [ND02]). In our case, we consider a simple collaboration scenario and we assume that the two agents are capable of applying actions to objects leading to similar effects, enabling the transfer, and that they operate on a shared space (i.e., a table accessible by both agents' arms). The morphology and the motor realization of the actions can be different between the two agents.

Some authors have studied the ability to interpret other agents under the deep learning paradigm. In [KYL17], a recurrent neural network is proposed to have an artificial simulated agent infer human intention (as output) from joint input information about objects, their potential affordances or opportunities, and human actions, employing different time scales for different actions. However, in that work a virtual simulation able to produce large quantities of data was used. This

is both unrealistic when trying to explain human cognition, and limited, because a simulator cannot model all the physical events and the unpredictability of the real world. In contrast, we use real, noisy data acquired from robots and sensors to validate our model.

We propose that the link between gesture and speech justifies the usage of machinery that, as in Automatic Speech Recognition (ASR), is suited for capturing dynamic time series data (i.e., a series of data points, listed in time order, that represent the measurement of some quantity over time). Hidden Markov Models (HMMs), which we explain in Sec. A.2.1, are one such statistical tool. We adopt an HMM-based approach to recognize body gestures that follow *temporally dynamic patterns*.

A.2 PROPOSED APPROACH

We now describe the design of our human gesture recognition model.

In this section, we present a human action recognition method for manipulative hand gestures, its theory (Hidden Markov Models), properties and training phase, and how to evaluate the tests.

From the beginning of this appendix, recall the online, real-time nature of our approach, which analyzes human gestures uninterruptedly, classifying them statistically. The set of body gestures which we use consists of grasp, tap, and touch movements[3]: these gestures pertain to manipulation tasks, and they are shown in Fig. 25.

Each of the gestures under consideration is represented by a Hidden Markov Model (HMM), which we will first define formally in Sec. A.2.1. Then, we will present two baseline models and our final model in Sec. A.2.2: these are models of increasing complexity and power, for combining the gesture HMMs together, permitting loops of gestures, and to treat noise information (i.e., the transition frames between two consecutive gestures) appropriately.

A.2.1 *Hidden Markov Models*

HMMs [Rab89] are a statistical tool for modeling time series data. They have been applied to the segmentation and recognition of sequential data with spatial and temporal variability such as speech, machine translation, genomics, financial data, among others. One of the advantages of HMMs, and a reason behind their popularity, is the fact that they are computationally tractable thanks to dynamic programming

[3] In an earlier version of our gesture recognizer [SSB13] we also trained a fourth gesture (touch, shown in Fig. 69), but we later discarded it in order to combine three gestures with the three actions of the pre-existing Affordance–Words system [Sal+12]. In terms of hand kinematics, the touch gesture is identical to the grasp gesture (shown in Fig. 66), the only difference being that in the former the object is not grasped by the person, whereas in the latter it is grasped.

techniques: marginal probabilities and samples can be obtained from an HMM with the FORWARD–BACKWARD algorithm [Rab89, Sec. III.A], and the most likely sequence of hidden states can be estimated with the VITERBI algorithm [Rab89, Sec. III.B].

An HMM with continuous outputs is defined by a set of discrete states $\mathcal{S} = \{s_1, \ldots, s_Q\}$ and by a set of parameters $\lambda = \{A, B, \Pi\}$, where $A = \{a_{ij}\}$ is the transition probability matrix, a_{ij} is the transition probability from state s_i at time t to state s_j at time $t+1$, $B = \{f_i\}$ is the set of Q observation probability functions (one per state i) with continuous values (typically mixture-of-Gaussians Probability Density Functions), and Π is the initial probability distribution for the states.

In our case, the model for each action is a left-to-right HMM, where the transition model between the Q discrete states $\mathcal{S} = \{s_1, \ldots, s_Q\}$ is structured so that states with a lower index represent events that occur earlier in time.

The continuous variables g_i are measured at regular time intervals. At a certain time step t, the D-dimensional feature vector can be expressed as $\boldsymbol{g}[t] = \{g_1[t], \ldots, g_D[t]\}$. The input to the model is a sequence of T such feature vectors $\boldsymbol{g}[1], \ldots, \boldsymbol{g}[T]$ that we call for simplicity G_1^T, where T can vary for every recording.

At recognition (testing) time, we can use the models to estimate the likelihood of a new sequence of observations G_1^T given each possible action, by means of the FORWARD–BACKWARD inference algorithm. We can express this likelihood as $\mathcal{L}_{\text{HMM}}(G_1^T \mid A = a_k)$, where a_k is one of the possible actions of Fig. 25. By normalizing the likelihoods, assuming that the gestures are equally likely *a priori*, we can obtain the posterior probability of the action given the sequence of observations (see Sec. 2.1.1 for an explanation of Bayesian inference) as

$$p_{\text{HMM}}(A = a_k \mid G_1^T) = \frac{\mathcal{L}_{\text{HMM}}(G_1^T \mid A = a_k)}{\sum_h \mathcal{L}_{\text{HMM}}(G_1^T \mid A = a_h)}. \tag{26}$$

A.2.2 *Baseline Models and Final Model*

We propose and compare three different models to recognize dynamic human gestures, with increasing complexity and expressive power for combining the gesture HMMs together with noise information. These are two baseline models (Model 1, Model 2) and our final model (Model 3). Model 3 is the one that permits uninterrupted gesture recognition, loops of gestures, and also to treat noise information (i.e., the transition frames between two consecutive gestures) appropriately.

All of the three models are composed by a set of HMMs (one for each dynamic human gesture), and an additional HMM for modeling noise. By noise we refer to *nongesture* data points, also known as *garbage* in the Automatic Speech Recognition (ASR) literature. Depending on the model, these garbage points will be represented either with a

single-state HMM, or with a multi-state HMM. Each state of all our HMMs emits a mixture of Gaussians as output, also known as Gaussian Mixture Model (GMM). By GMM we mean a linear superposition of components with Gaussian densities (a GMM can be thought as a single-state HMM).

Having established that all of the models are characterized by multi-state HMMs, one for each gesture, let us explain the main difference among the three models with respect to the *garbage* representation. The three different graphical models, represented in Fig. 64, are:

- Model 1: the garbage is modeled as a single-state HMM (therefore it coincides with a simple GMM without temporal transitions). In the experimental part (see Sec. A.3) we will see how this model is not able to represent transitions during nongesture phases;
- Model 2: the garbage is modeled as a multi-state HMM, thus providing a richer representation able to capture nongesture transitions. However, after a gesture is done being recognized, this model does not permit to recognize another gesture (absence of transitions between gestures);
- Model 3: like the previous one, in addition transitions between gestures are allowed. This permits us to finally capture the uninterrupted aspect of natural human gesture sequences.

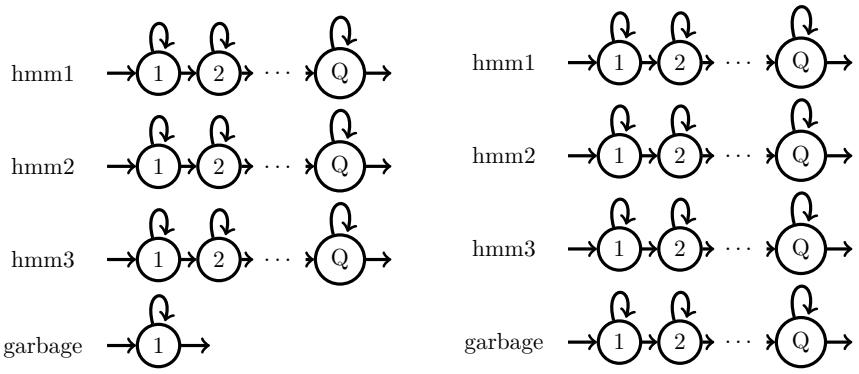
A.2.3 *Feature Selection*

The features that we use to train our gesture classifier are the spatial 3D coordinates of a human's hand joint being tracked[4], and they can be calculated online without having to wait for an input sequence to be finished. For this reason, we perform no normalization or filtering that requires knowledge of the completed sequence (e.g., global minima and maxima). The 3D joints coordinates can be obtained with general-purpose depth cameras like the Microsoft Kinect or the Asus Xtion Pro. Fig. 65 illustrates the idea of a time series of 3D coordinate features from a dynamic gesture.

For the simple one-hand actions that we consider as in Fig. 25, tracking one hand/arm is sufficient. While we do not apply normalization steps to the coordinates, we do apply a geometric transformation to the coordinates obtained with depth cameras and skeleton recognition algorithms: we set our reference frame to be *recentered on the human torso*, instead of the default sensor-centered reference frame[5]. This

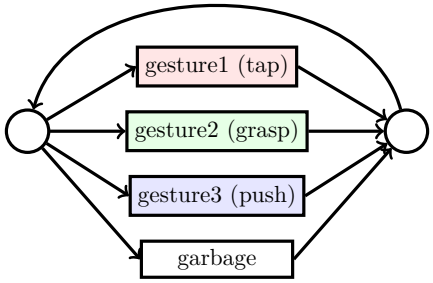
[4] It is possible to use more joints as features, for example the concatenation of hands, elbows, shoulders, torso and head coordinates, as mentioned in Sec. A.1. In our domain, data, and tests, using just the hand joint features yielded the highest performance.

[5] The orientation is kept with respect to the camera, because (i) we focus on frontal person views, not sideways views; (ii) we will not use orientation features but only positional ones.



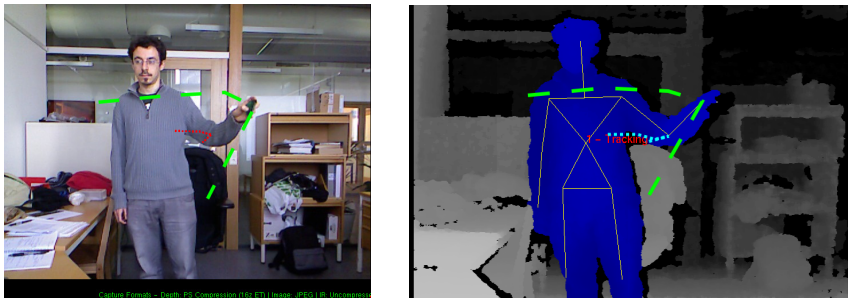
(a) Model 1.

(b) Model 2.



(c) Model 3.

Figure 64: Different Hidden Markov Model structures considered when developing our gesture recognizer. Every state is associated to an emission Probability Density Function which is a mixture of Gaussians.
Model 1: one multi-state HMM per human gesture, one single-state HMM (i.e., a GMM without transitions) for garbage data. Each model in {hmm1, hmm2, hmm3} is independent from the other ones and can have an arbitrary number of states.
Model 2: one multi-state HMM per human gesture, one multi-state HMM for garbage data. Each model in {hmm1, hmm2, hmm3, garbage} is independent from the other ones and can have an arbitrary number of states
Model 3. one multi-state HMM per human gesture, one multi-state HMM for garbage data. These four configurations are then merged with an outer transition loop. Each rectangle represents a gestural HMM like the ones shown in Fig. 64b, however, because of the merging, the original state indexes of {hmm1, hmm2, hmm3, garbage} must be now uniquely renumbered.



(a) RGB view. Hand trajectory shown in green, elbow trajectory in red.

(b) Depth skeletal view. Hand trajectory shown in green, elbow trajectory in light blue.

Figure 65: A *tap* human gesture, with temporal trajectory of selected joints being highlighted. The 3D coordinates of the joints of interest constitute the inputs of our statistical models of Fig. 64.

transformation has two motivations, a conceptual and a practical one. Conceptually, it gives more importance to the human user, by placing a virtual mobile point attached to the human user, instead of relying on a fixed point attached to a camera. From a practical perspective, this transformation provides *invariance to starting point* of a physical gesture. In other words, the user can perform actions at any distance or angle from the sensor, and these actions will always be measured with regards to his torso coordinate.

A.2.4 *Training*

Following the notation of the HMM MATLAB toolbox by Kevin Murphy [Mur12], we introduce the following quantities relative to mixtures of Gaussians, also known as GMMs. A mixture of M Gaussian components is the weighted sum of multivariate Gaussian distributions $m = 1, \ldots, M$, each with mean μ_m and covariance Σ_m:

$$p(x \mid w_m, \mu_m, \Sigma_m) = \sum_{m=1}^{M} w_m \mathcal{N}(x \mid \mu_m, \Sigma_m), \tag{27}$$

where x is a data point (in our case the 3D hand coordinates), and w_m are the mixing weights satisfying $0 \leq w_m \leq 1$, $\sum_{m=1}^{M} w_m = 1$.

In addition, for an HMM with Q temporal states we define a *weights matrix* containing all the w_m as follows: each row represents a state $q = 1, \ldots, Q$, each column represents a mixture component $m = 1, \ldots, M$.

Let O be the size of an observation vector (e.g., the length of a sequence that we wish to fit to the model during training, or to classify during testing: a segment of O hand input data points). Then, we define the *means matrix* with size $Q \times OM$ to contain the $\mu_m^{(q)}$ associated to each state q, observed points and mixture components. Finally, we

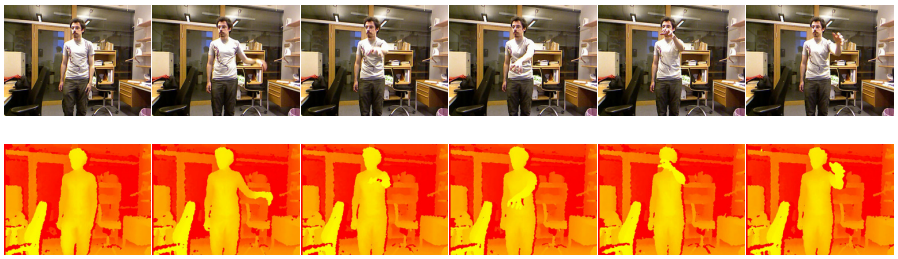


Figure 66: Gesture recognition data: example sequence of the *grasp* human gesture. Top: image frames, bottom: depth frames. Amplitude: wide, recording number: 2.

define a *covariance matrix* with size $OQ \times OM$ to contain all the $\Sigma_m^{(q)}$ for each state, observed point and mixture component.

Algorithm 2 Gaussian mixture splitting. M_{des} is the final desired number of Gaussians; pertDepth is a perturbation depth constant (we set it to 0.2).

1: **procedure** UPMIX(weights matrix, means matrix, covariance matrix, M_{des})
2: **while** $M < M_{\text{des}}$ **do** ▷ M: current no. of Gaussians
3: weights: split heaviest entry in two parts with equal weight
4: means: duplicate corresponding entry
5: means: perturb new entries to be
$\text{means}_{1,2}(i) \pm = \sqrt{\text{cov}(i,i)} \cdot \text{pertDepth}$
6: covariances: duplicate corresponding entry
7: $M := M + 1$
8: **end while**
9: **end procedure**

For the models described in the remainder of this section, we collected *training data* of one person performing actions without manipulated objects, in other words we trained the gesture recognizer with gesture *pantomimes*. Each action was performed in *three different amplitudes*: wide gestures (emphatic arm movements), medium-width gestures and narrow gestures (subtle movements). Each amplitude class was acquired multiple times (12–14 times), thus providing around 40 training repetitions for each of the manipulation actions considered. We show examples of our dataset, with the different amplitudes, in Figs. 66, 67, 68.

This dataset was used to train all the statistical models described in this section, and we empirically determined suitable initialization characteristics and meta-parameters for our HMM:

- left-to-right HMM transition probability matrix (initially every state can either transition to itself or to the next state with equal

Figure 67: Gesture recognition data: example sequence of the *tap* human gesture. Top: image frames, bottom: depth frames. Amplitude: medium, recording number: 4.

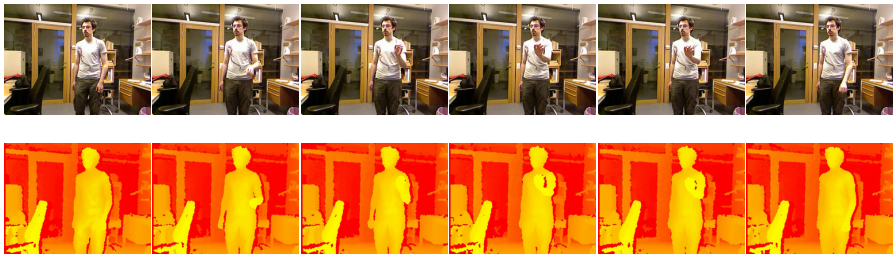


Figure 68: Gesture recognition data: example sequence of the *push* human gesture. Top: image frames, bottom: depth frames. Amplitude: narrow, recording number: 6.

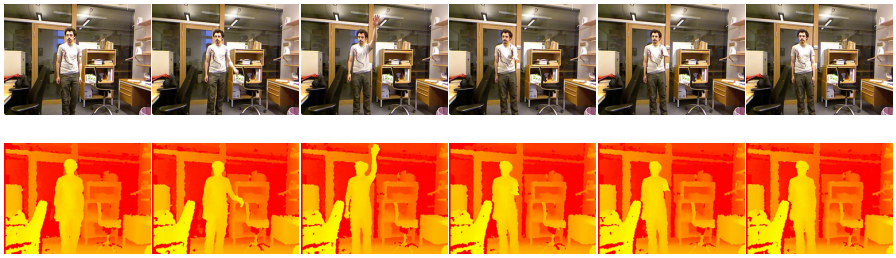


Figure 69: Gesture recognition data: example sequence of the *touch* human gesture (see also footnote [3] on p. 146). Top: image frames, bottom: depth frames. Amplitude: wide, recording number: 3.

probability – with the exception of the last state which only transitions to itself):

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 & \cdots & 0 \\ 0 & 0.5 & 0.5 & 0 & \vdots \\ \vdots & \dots & \ddots & \dots & \vdots \\ 0 & \dots & 0 & 0.5 & 0.5 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix};$$

- state probability distribution (initially we impose that we start from the first state, all others having zero probability of starting):

$$\Pi = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^\top.$$

In all of the models described below, HMMs were trained with the incremental *mixture splitting* technique, commonly used in ASR[6], in order to obtain the desired number of output Gaussians M_{des}. With this approach, initially every mixture has $M = 1$ Gaussian (with mean initialized to empirical mean and covariance initialized to empirical covariance of the input data, respectively); we run the BAUM–WELCH algorithm[7] to improve HMM parameter estimates; then we enter a cycle, in which we run UPMIX (adapted from [You+06, Sec. 10.6], sketched in Alg. 2) and BAUM–WELCH, increasing the counter M; the cycle terminates when the weights matrix contains M_{des} Gaussians as desired. This technique allows us to achieve higher likelihoods than with simple BAUM–WELCH (EM), as shown in Fig. 70.

The first statistical model that we define as a baseline for our experiments (Model 1, shown in Fig. 64a) consists of several multi-state HMMs with continuous outputs, one per gesture, and one single-state HMM with continuous outputs (i.e., a GMM) for garbage. We use the latter to capture the *noise* (i.e., *nongesture* points), and we use the multi-state HMMs to model the actual human gestures: each HMM is trained for one gesture (many repetitions of the same gesture with different spatial amplitudes and speed). However, the single-state nature of the garbage model does not allow to capture the dynamic nature which is present in the noisy transitions between subsequent gestures in an uninterrupted, spontaneous human sequence of hand movements. In other words, this noise model can only capture the noise at the very beginning and at the very end of a sequence of many gestures, but not the noise between two consecutive gestures within the sequence.

A second baseline statistical model that we train (Model 2, shown in Fig. 64b) is similar to the previous one, but it tackles the main

[6] However, in recent years, several research groups in the ASR community have adopted approaches based on deep neural networks [Hin+12], rather than on HMMs.

[7] The BAUM–WELCH algorithm is an instance of the EXPECTATION–MAXIMIZATION (EM) algorithm used to estimate HMM parameters: in our case the weights matrix, the means matrix and the covariance matrix.

limitation of Model 1 by assigning a dynamic, multi-state HMM nature to the garbage model, thus improving the separation criterion between gestures and nongestures. In Model 2, the garbage model consists of an HMM with several states trained with garbage data, and the remaining HMMs capture the gestures as before. For simplicity, we fix the number of states Q to be equal for all gestures and for the garbage.

So far, Model 1 and Model 2 have considered the individual gesture models to be independent from each other: each of them has its start, intermediate and final states, as well as its own prior probabilities, state transition probabilities and observation probabilities. In Fig. 64c, we now merge those models into one single HMM with many states and appropriately combined probability matrices (Model 3). Merging the previously trained statistical models into one new HMM entails the following steps:

- weights matrix, means matrix: horizontal concatenation of previous models' matrices;
- covariance matrix: block diagonal concatenation of previous models' covariance matrices. For example, from covariance matrices $\Sigma_1, \ldots, \Sigma_{\#\text{gestures}}$ we obtain

$$\begin{bmatrix} \Sigma_1 & 0\ldots & \ldots 0 \\ 0\ldots & \ddots & \ldots 0 \\ 0\ldots & \ldots 0 & \Sigma_{\#\text{gestures}} \end{bmatrix};$$

- initial probability vector: stochastic concatenation of previous models' priors, i.e., a column vector with $(Q \cdot \#\text{gestures})$ entries, all set to zero except for the first state of each gesture, set to $1/\#\text{gestures}$;
- transition matrix: $(Q \cdot \#\text{gestures}) \times (Q \cdot \#\text{gestures})$ block diagonal matrix built from the previous $(Q \times Q)$ matrices, allowing transitions from each of the previous HMMs' end states into the first state of any previous HMM (this allows the uninterrupted gesture recognition algorithm to enter a sequence j at the end of any finished sequence i).

A.3 EXPERIMENTAL RESULTS

In this section, we show recognition results obtained by employing common Hidden Markov Model (HMM) inference methods [Rab89] on our models: (i) FORWARD–BACKWARD algorithm for isolated gesture recognition, which computes the most likely single action recognized from a test data sequence; the major downside of this technique is that it requires the segmentation of test data, thus the availability of all test data offline; (ii) VITERBI algorithm for uninterrupted gesture recognition: this method does not require prior segmentation of test data, and

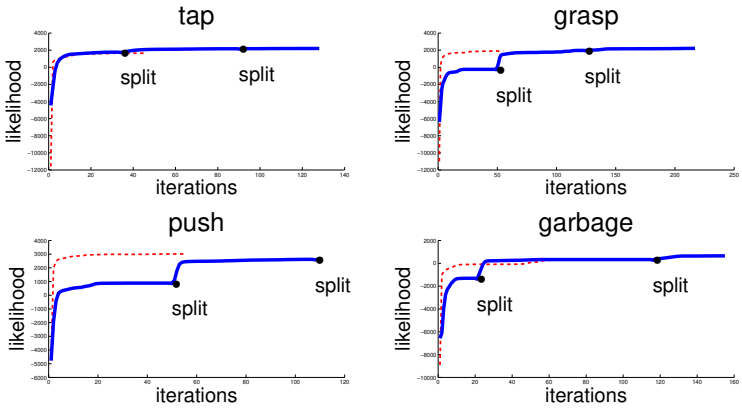


Figure 70: Gesture recognition: evolution of the likelihoods of the gesture models during training, comparing EM algorithm when initialized with M=3 Gaussian outputs from the headstart (dashed red line) and when employing the mixture splitting technique (solid blue line, with points where the number of mixtures was incremented being highlighted as circles). With the exception of the "push" gesture class, our method achieves a higher likelihood than simple EM.

it outputs the estimated sequence of actions (state path) that best explain the test data sequence. We will apply FORWARD–BACKWARD on all of our three models, and VITERBI on Model 3. This permits us to show early intention recognition performance in the case of Model 3.

For early intention recognition on Model 3, we consider for simplicity two possible cases: correct succession of gestures (the succession is defined *a priori*) or incorrect succession of gestures. We assume that the sequence Push-Tap-Grasp corresponds to the intention of "drinking" (i.e., the user is about the grab the drinking cup in the correct way), whereas any other sequence of gestures does not correspond to that intention. Model 3 is capable of providing an intention recognition result as soon as the model transitions into the (first state of the) last gesture in the sequence, thus before the whole action is over.

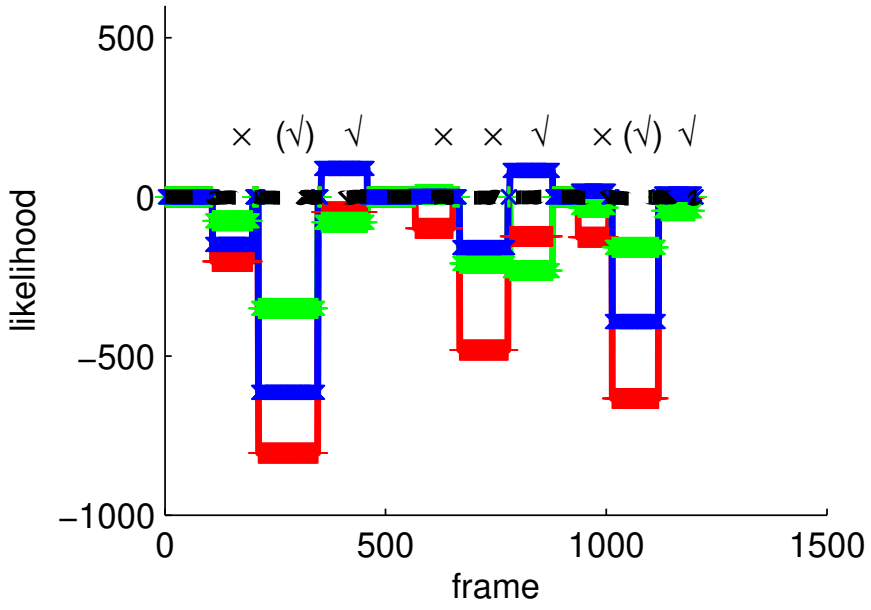
Gesture recognition tests for the different models and algorithms are shown in Figs. 71 for the baseline Models 1 and 2, and in Figs. 73 and 74 for the proposed approach which uses Model 3. Both training and test sequences were collected by the authors using a depth sensor recording gestures from one person. In order to make the system robust to different people with different heights and sizes, we apply a normalization step in the measurements, dividing them by the average shoulder width, which is obtained after a few seconds of skeleton tracking (this can be done in near-real time). The *feature space* that we use coincides with the 3D position coordinates of the hand joint in time; enriching the space with the coordinates of other joints such as shoulder and elbow actually decreased the recognition performance in our tests.

FORWARD–BACKWARD classification results with Model 1 are shown in Fig. 71a. The test sequence consists of nine consecutive gestures, specifically three triplets (tap, grasp, push), the first triplet occurring at slow speed, the next one at medium speed, and the final one at fast speed. In this experiment, the test sequence was segmented similarly to how training data was segmented. In general, this is not safe to assume in a real time scenario, unless a delay is added. The problem here is that the gesture threshold is "too strict", voiding many HMM assignment classifications, even where they are correct.

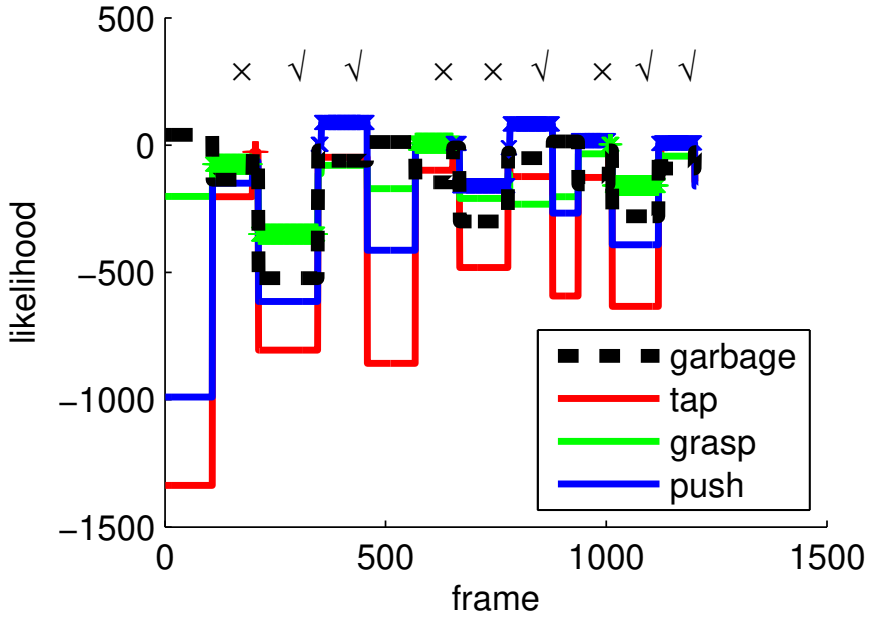
In the Model 1 experimental setup described above, gesture recognition performs poorly, with a recognition rate below 50%, mainly due to the fact that the garbage GMM cannot learn the temporal nature of nongesture (between-gesture) transitions.

Taking Model 2 (Fig. 64b) into account, Fig. 71b displays improved FORWARD–BACKWARD classification results. Compared to Model 1, this model is better in correctly separating garbage segments from gesture ones, which we expected because the gesture classifier is richer here, being able to capture the dynamic nature of between-gesture transitions with its dedicated HMM. However, classification still suffers during probabilistic gesture class assignment, confusing taps with grasps for all velocities of the input sequence.

Model 3 (Fig. 64c) allows us to illustrate the performance of our system with the VITERBI algorithm results of Figs. 73 and Fig. 74. The algorithm reconstructs the optimal (most likely) gesture state path resulting from a given test sequence. In these experiments, we assume that the context is described as the human–robot manipulation scenario shown in Fig. 72, whereby a user has to correctly move and grasp an object on a table, without making it collide with other objects: the correct strategy (intention) corresponds to the Push-Tap-Grasp sequence, a fact known *a priori* by the system. In Fig. 73 (left), the recognition accuracy is high (actions are detected in the correct temporal regions, and they are classified correctly 3/3 times) and the intention of the user (see p. 155) is inferred to be coincident to the correct Push-Tap-Grasp strategy. On the other hand, Fig. 73 (right) shows a case where the recognition is still correct (the action sequence is correctly identified as Tap-Push-Grasp), but the wrong intention or strategy on the part of the user (see p. 155) can be detected – thus allowing the robot to intervene, as motivated by the scope of this appendix. Finally, Fig. 74 shows a test sequence which the system failed to recognize correctly as Push-Tap-Grasp (the order of actions actually performed by the user), even though it still classified correctly most of the actions (2/3). The failures are due to limitations in training data, in the sensor employed and in the general statistical robustness of our model.



(a) Model 1 (Fig. 64a) performance on segmented input sequence.



(b) Model 2 (Fig. 64b) performance on segmented input sequence.

Figure 71: Gesture recognition likelihood computed with FORWARD–BACKWARD algorithm. √: correct gesture classification, ×: wrong classification, (√): classification is correct but is voided by GMM nongesture threshold.

(a) Initial configuration. (b) Intermediate configuration. (c) Final configuration.

Figure 72: Scenario for testing early intention recognition, by spotting the correct or incorrect successions of gestures: a human user sitting on the left has to move the mug next to the bottle, avoiding the red obstacle on the table, so that a robot bartender can fill the mug. The repertoire of permitted actions corresponds to the three gestures tap, grasp, push. The robot system knows that Push-Tap-Grasp is the correct strategy considering the initial table configuration, while for instance Tap-Push-Grasp is an incorrect strategy due to geometrical constraints. Fig. 73 (left) and Fig. 73 (right) reflect these two situations from the pattern recognition perspective.

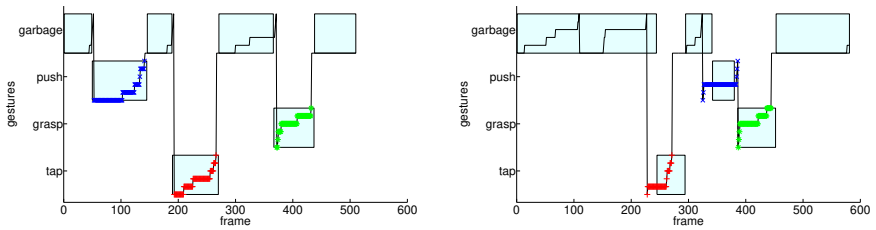


Figure 73: Gesture recognition VITERBI results on the early intention recognition scenario of Fig. 72 without noise. Red plus signs: tap states, green stars: grasp states, blue crosses: push states, rectangles: human-labeled ground truth segmentation.
Left: a Push-Tap-Grasp action sequence performed by the user is correctly recognized (3/3 score), the user intention (see p. 155) is found to be correct too, meaning that it is feasible given the contextual geometric configuration of table and objects. Right: a Tap-Push-Grasp action sequence is correctly recognized (3/3 score), although the user intention can be detected by the system as being incorrect considering the current context – allowing the system to alert the user.

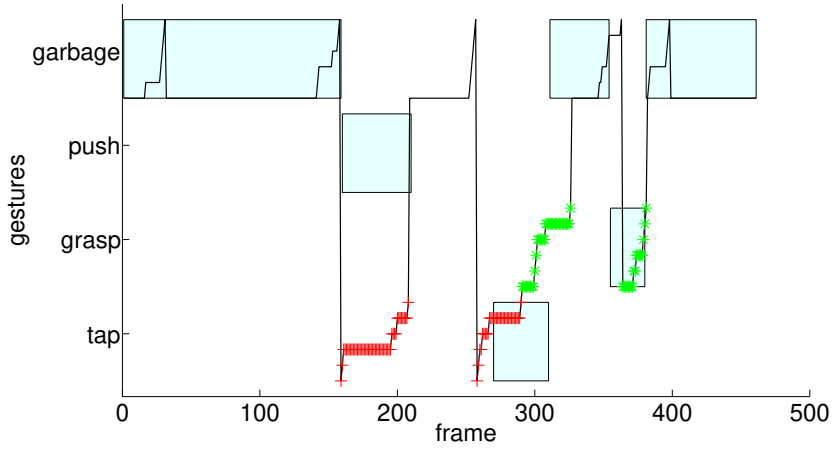


Figure 74: Gesture recognition VITERBI result of the early intention recognition scenario of Fig. 72 showing the limitations of our approach in the presence of noise. The Push-Tap-Grasp action sequence performed by the user is not correctly classified by the statistical model, because the test data was noisy compared to the trained model, both in terms of space (amplitude of the gesture) and time (speed). Red plus signs: tap states, green stars: grasp states, blue crosses: push states, rectangles: human-labeled ground truth segmentation.

A.4 CONCLUSIONS AND FUTURE WORK

Gestures are a paramount ingredient of communication, tightly linked with speech production in the brain: the ability to interpret the physical movements of others improves the understanding of their intentions and thus the efficiency of interactions. In Sec. A.2 we proposed a method to recognize gestures in an uninterrupted, real time setting with statistical methods.

The results of our gesture recognition models are discussed in Sec. A.3 and summarized in Table 17.

Table 17: Summary of classification methods and results obtained with the gesture recognition models of Fig. 64.
*: Model 1 yields 33% when considering the garbage/gesture decision threshold, 55.6% without it.

model	classification method	performance
Model 1	GMM garbage, HMMs each gesture	33*% (FORWARD–BACKWARD)
Model 2	HMMs garbage and each gesture	55.6% (FORWARD–BACKWARD)
Model 3	HMMs garbage and each gesture, loop	66.6–100% (VITERBI)

B

HUMAN PERCEPTION OF ROBOT GESTURES

In this appendix, we study the *social attitude perceived by humans when a humanoid robot moves its body parts, with no facial expressions involved*[1].

We conduct a human experiment in which human subjects are sitting in front of the iCub robot (see Sec. 3.1), they observe it while it performs pre-programmed head and arm movements, and they respond to a questionnaire. We select the questions in such a way that they are not boring or repetitive by using a machine learning algorithm based on the idea of active learning (i.e., an algorithm that actively chooses the data from which it learns). We report our motivation in terms of robot gesture design, our findings regarding the expressiveness of robot motion according to humans, and we propose an automated system to conduct this type of studies in a way that keeps the time devoted to making questions to humans to a minimum, while maximizing the information acquired for statistical purposes and robot gesture design.

A note on terminology: in this appendix, (i) we call *parameters* the numerical terms associated to robot gestures types (i.e., the variables controlling joint positions, velocities and timings, as listed in Tables 19–22); (ii) we call *parameterized gestures* the combinations of gesture types, parameters and corresponding values (i.e., gesture–parameter–value tuples, see Sec. B.2.2). However, (iii) we call *weights* the Bayesian Network (BN) probabilies which specify the Conditional Probability Distributions (CPDs) associated to the network. Those weights are usually called "parameters" in Bayesian learning literature (see Sec. 2.1 and [TK00; Bis07; Pea88]), but in this appendix we avoid using that term to prevent confusion with our specific meaning of robotic gesture parameters.

This appendix is the subject of the following publication:

- Giovanni Saponaro and Alexandre Bernardino. "Generation of Meaningful Robot Expressions with Active Learning". In: *ACM/IEEE International Conference on Human–Robot Interaction*. Late Breaking Report. 2011, pp. 243–244. DOI: 10.1145/1957656.1957752.

The outline of this appendix is as follows: Sec. B.1 provides motivational considerations and related works for the study in this appendix.

[1] In a previous version of this work we referred to "emotions" [SB11], but we now use a broader expression: "social attitudes", in the sense of social states of mind. The reason for this change is that "emotions" typically refer to anger, disgust, fear, happiness, sadness, and surprise [EFE72]. The social attitudes that we consider are: agreement, anger, distraction, approval, and disapproval.

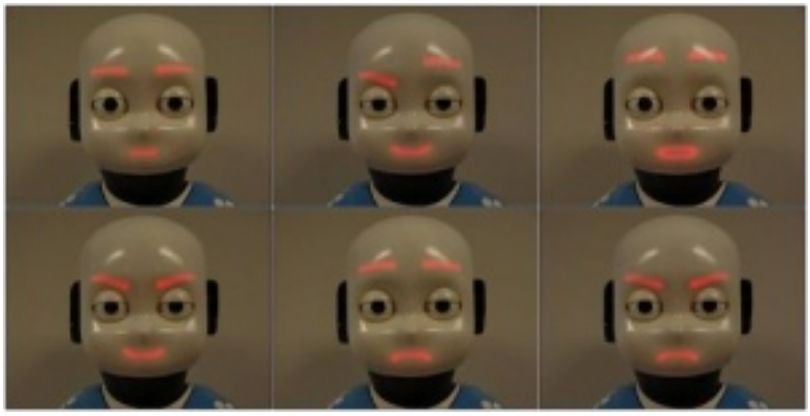


Figure 75: A set of iCub facial expressions using eye LEDs, mouth LEDs and eyelid motors.

Sec. B.2 illustrates the proposed approach. Sec. B.3 reports the results, and Sec. B.5 draws the conclusions.

B.1 BACKGROUND AND RELATED WORK

While the iCub robot has been used to display some basic emotional social states [BWW15; Raf+16; Pac+17], this task has been mainly concerned with the robot *face* by controlling its eyebrow LEDs, mouth LEDs and eyelid servomotors as shown in Fig. 75: all features that are indeed extremely informative for the transmission of feelings. By contrast, we intend to explore the social capabilities of *joint movements* located in the head, neck, torso and arms of the robot, and how human users interpret these movements when face expressions are disabled. We deliberately do *not* exploit the facial features of the robot, because face to face interaction, being the chief interaction modality for humans, would be too informative: using robot facial expressions drives and biases the perception of the other's features, such as the movements, so we choose to turn off the facial expressions, giving the robot face a neutral look, as in Fig. 76.

Our source of inspiration for transmitting social attitudes with movement while disregarding the face, is *puppetry* [LC11]. Based on literature related to communicative robot body movements designed by puppeteers, we build a library of robot gestures with their expected attitude value (ground truth). We then model a mapping between robot movements and social attitudes perceived by humans, as a multinomial distribution.

With this setup, we ask people to attribute social attitudes ratings to robot movements. The answers are used to update the gesture–attitude matches. In addition, we employ active learning to conduct the study in a way that minimizes time and maximizes the information gain. This framework gives the system the ability to inquire about

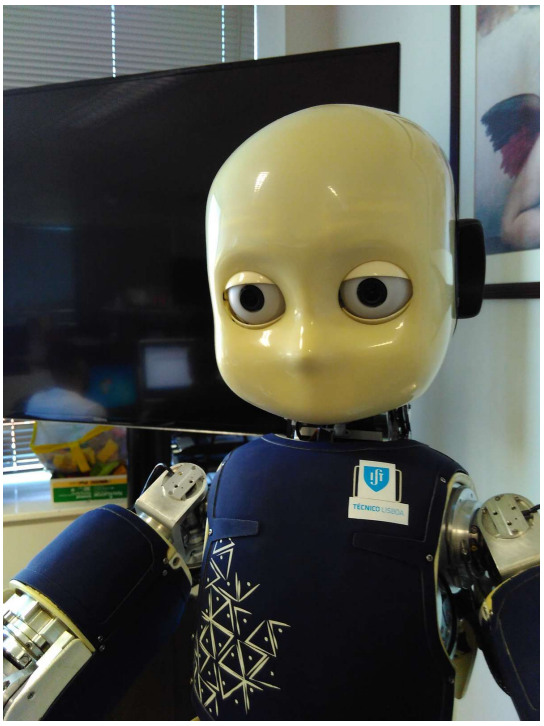


Figure 76: The iCub face in a neutral position, without facial expressions, as it was used during the human study of Sec. B.2.

movements that are ambiguous, showing them and getting feedback more often than easily-perceived ones.

In the context of social robotics, we now review literature concerned with the perceived value associated to *body gestures* and movement, in order to assess the clarity and the effectiveness of the iCub body gestures that we show to laypeople with the method explained in Sec. B.2.

The role of body expressions in *affective human–robot interaction*, and more generally in affective computing, has been the subject of several studies: for a comprehensive review, we refer the reader to [KB13]. Body movements are powerful means for conveying social attitudes, and they also facilitate multimodal interaction in which they assume other functions besides being mere gestures or controllers: for instance, in whole-body videogames with Microsoft Kinect, PlayStation Move or Nintendo Wii, movements capture and affect our own performance in the game [Bia13]. The information contained in body gestures can be incorporated into many applications, ranging from security and surveillance, to law enforcement, entertainment, videogames, education, and health care: for example, during rehabilitation exercises, certain specific movements and postural patterns inform clinical practitioners about the emotional conflict in the patients, and their (im)possibility to relax.

As far as the iCub robot is concerned, it has been used in specific human–robot interaction studies focusing on the role of certain modal-

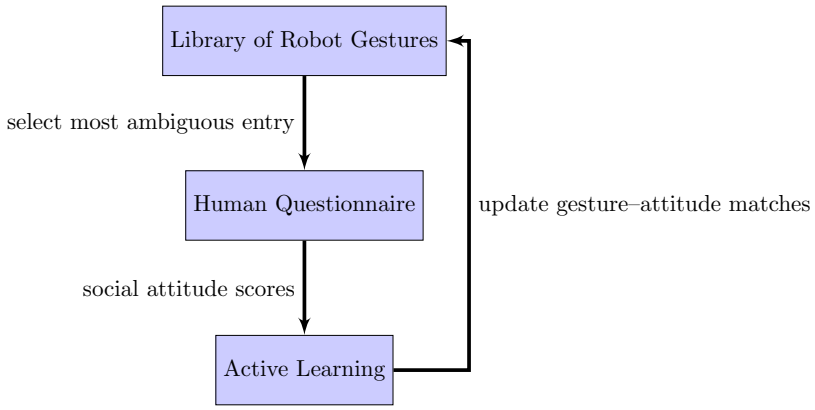


Figure 77: Block diagram of the proposed approach to study the matches between robot gestures and perceived social attitude in humans. The library of gestures is pre-programmed (see Table 18); the questionnaire is a multiple-scale Likert one; the scores are numbers corresponding to the Likert answers, and they are used by the Active Learning block to update an internal matrix of matches and select which robot gestures should be shown next.

ities like gaze [Bou+12] or touch [ASB10] during interactions. It has been used to display basic emotions with its face (Ekman's Six Basic Emotions [EFE72] as shown in Fig. 75: anger, disgust, fear, happiness, sadness, surprise). To the best of our knowledge, the iCub has not been used to study the social attitude value carried by gestures and movement, which we will do in the remainder of this appendix.

Li and Chignell [LC11] have researched the social attitude understanding of robot gestures by human users, in particular by comparing robot gesture movements designed by common people versus those designed by *puppeteers*, and asking users whether they perceived one of many social states of mind. The intuition is that puppetry artists are able to create engaging and communicative personalities merely by manipulating the kinematics of puppet figures. We apply the same methodology with the iCub robot.

B.2 PROPOSED APPROACH

We design a library of robot gestures (without using facial expressions) and we survey a number of people about what social attitude they perceive when a humanoid robot performs such gestures. We study the results and draw conclusions about which robot movements are clear and which are not, looking at the attitude scores attributed by human subjects to robot gestures.

Recall that we conduct the human study in such a way that minimizes the time necessary for the interview sessions (as well as the robot usage,

Table 18: Library of robot gestures, ordered by a sequential index, each one corresponding to an expected social attitude (ground truth according to the robot designer).

Gesture type index	*Name*: description	Expected social attitude (ground truth)
$T = 1$	*nod*: head tilts up and down	agreement
$T = 2$	*punch*: rapidly extend fist in front of robot	anger
$T = 3$	*look out*: abruptly deviate robot head and gaze to a side	distraction
$T = 4$	*thumbs up*: show fist and move thumb up	approval
$T = 5$	*thumbs down*: show fist and move thumb down	disapproval

preventing breakage of delicate robot parts and consequent downtime) by employing an active learning technique, which processes the human scores attributed to the robot gestures, and decides which of the gestures require more corrections and further human feedback information: accordingly, the chosen gestures is shown, in order to maximize the information of the matches from robot gestures to perceived social attitudes, modeled as probabilistic mappings with a Bayesian Network.

As we make the robot perform the gestures and we ask human users to attribute scores about the perceived attitudes, we address two key issues related with robot gestures: (i) whether simple robot gestures, in the absence of facial cues, convey the expected attitudes to viewers; (ii) what contextual and motion characteristics aid gesture understanding the most [LC11].

We now illustrate the details of the proposed approach.

B.2.1 *Design of Basic Robot Gestures*

Because we are interested in mapping simple robot gestures to social attitudes that we wish to transmit successfully, the first crucial aspect of the work consists in *designing a library of basic robot motions* which do not rely on facial information, and their expected perceived attitudes. We do this design task manually, i.e., a set of precise robot gestures and joint trajectories are pre-programmed by hand, modulating the available joint positions and timings, using the iCub kinematic structure (`http://wiki.icub.org/wiki/ICub_joints`) and evaluating the resulting movements qualitatively (this process being repeated a number of times). In order to actuate the robot joints, we use the position-based control, the Cartesian Interface [Pat+10] and the Gaze Interface [Ron+16], all available in the iCub software repository (`http://www.icub.org`). The end result of this process is summarized in Table 18, which lists the designed gestures with corresponding nu-

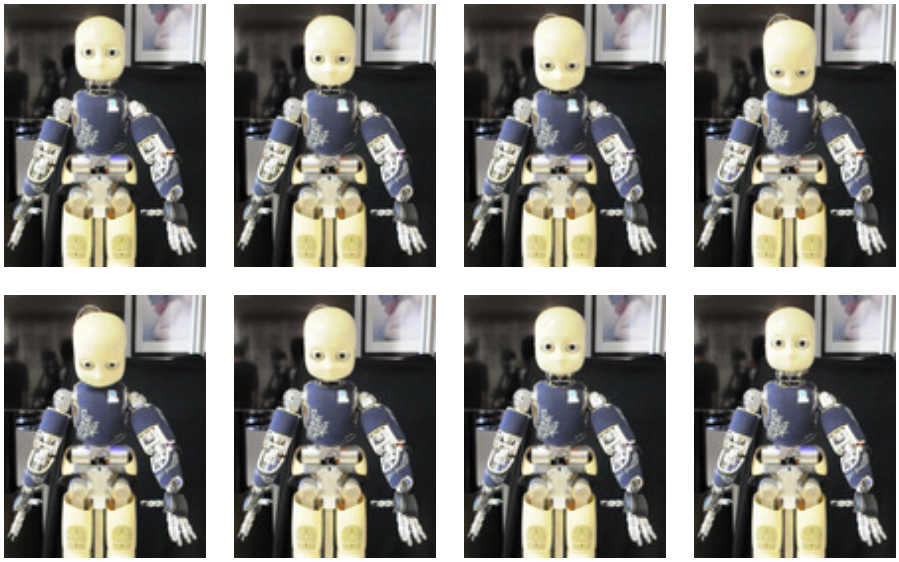


Figure 78: Temporal snapshots of the iCub *nod* gesture. Video available at `https://youtu.be/w0uvGzmTkQM`

Figure 79: Temporal snapshots of the iCub *punch* gesture. Video available at `https://youtu.be/9pL28w1juaU`

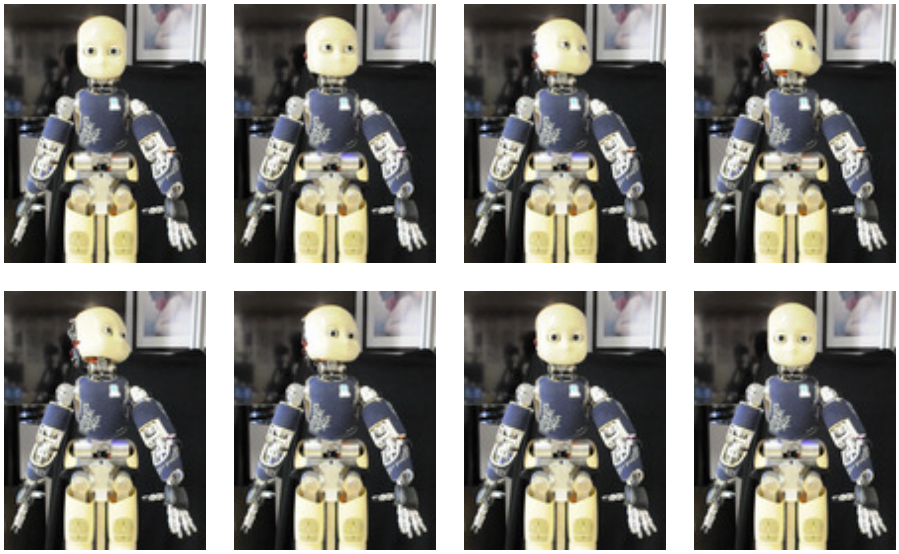


Figure 80: Temporal snapshots of the iCub *look out* gesture. Video available at `https://youtu.be/RJIx27xHJ34`

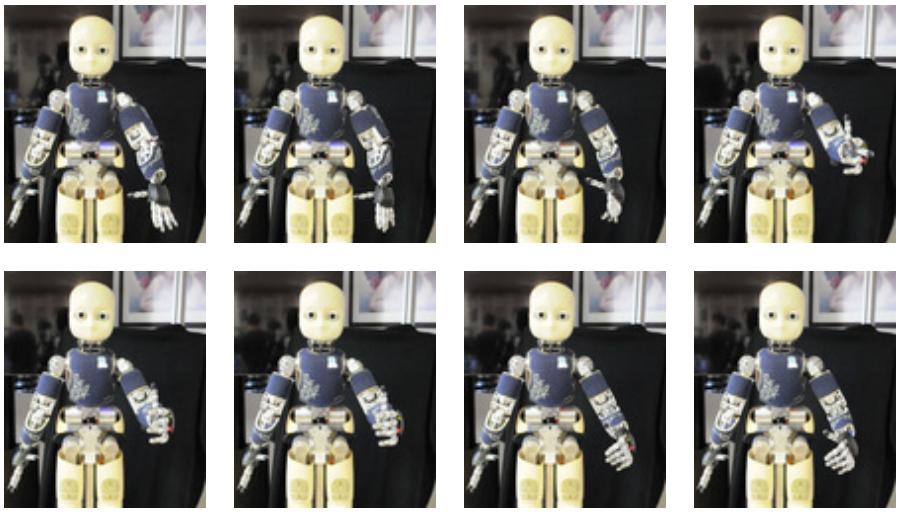


Figure 81: Temporal snapshots of the iCub *thumbs up* gesture. Video available at `https://youtu.be/ZHN91AN1t_o`

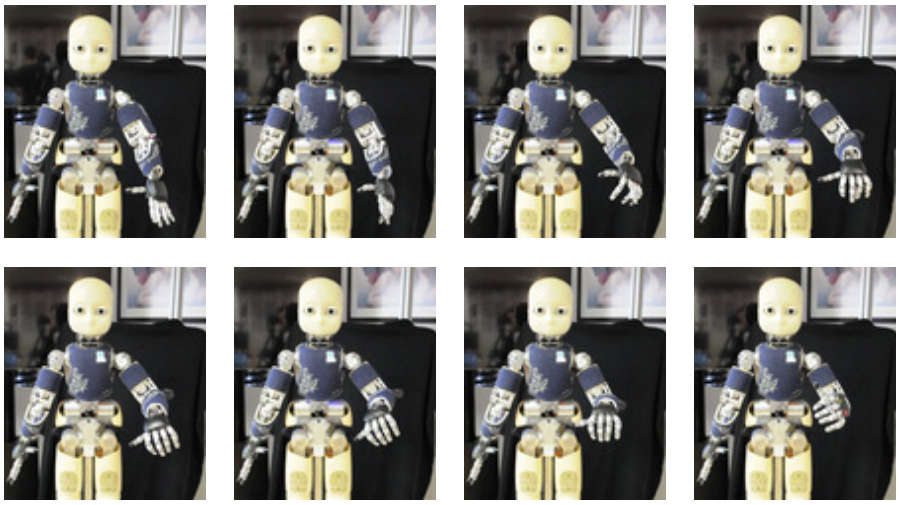


Figure 82: Temporal snapshots of the iCub *thumbs down* gesture. Video available at `https://youtu.be/AKvA6lIt25Q`

merical identifiers, names, verbal descriptions and, importantly, an expected attitude ground truth value. Recall that the main objective of this work is to study the match between robot gestures (A) and human-perceived social attitudes (E).

Each of the designed gesture types, listed in the first column of Table 18, consists of a trajectory computed between two or more points in space (either joint space or Cartesian space, depending on the gesture being designed) with adjusted velocities and timings between the points to interpolate. We show the *nod* robot gesture in Fig. 78, the *punch* robot gesture in Fig. 79, the *lookout* robot gesture in Fig. 80, the *thumbs up* robot gesture in Fig. 81, and the *thumbs down* robot gesture in Fig. 82. The main issue encountered during the interpretation of the gestures by interviewees was about the *punch* gesture: this gesture looked ambiguous, likely because the fist was not completely closed.

B.2.2 *Parameterization of Robot Gestures*

The robot gestures that we study are divided into a few basic types T, as listed in Table 18. However, to have greater flexibility while displaying robot gestures as well as during the machine learning phases, we enrich the model with a set of specific parameters P (different for each gesture type) that modulate the appearance of robot gestures, listed explicitly in Tables 19–22. Each of the parameters takes a discrete value from a set V (different for each gesture–parameter pair), which can be seen as a histogram (e.g., a velocity-like parameter value range can be divided into a low-velocity bin v_1, a medium-velocity bin v_2 and a high-velocity bin v_3).

Table 19: Parameters of the "nod" gesture.

Parameter index	Parameter symbol	Meaning
P_1	$x_0^{(0)}$	initial position of neck pitch joint
P_2	$x_0^{(1)}$	final position of neck pitch joint
P_3	$\dot{x}_0$	velocity of neck pitch joint
P_4	$t_{(0)\to(1)}$	time to transition from initial to final positions
P_5	$t_{(1)\to(0)}$	time to transition from final to final positions

Table 20: Parameters of the "punch" gesture.

Parameter index	Parameter symbol	Meaning
P_1	$\dot{x}_{7:15}$	velocity of finger joints when closing hand
P_2	$t_{(0)\to(1)}$	time to transition arm joints from initial to final positions
P_3	$t_{(1)\to(0)}$	time to transition arm joints from final to initial positions

As a result, in the general formulation of our framework described below, the matches (Bayesian Network weights) are between gesture–parameter–value tuples (rows) and perceived attitudes (columns). This means that the number of rows can potentially be much larger than the number of columns: active learning is especially useful in these scenarios, being able to select a query (row) among many of them, according to a probabilistic criterion. Even though our formalism is quite general, in the experimental results (Sec. B.3) we will make some simplifying assumptions as to the number of parameters and values.

Table 21: Parameters of the "look out" gesture.

Parameter index	Parameter symbol	Meaning
P_1	$x_{0:2}^{(0)}$	initial position of neck joints
P_2	$x_{0:2}^{(1)}$	final position of neck joints
P_3	$\dot{x}_{0:2}$	velocity of neck joints
P_4	$t_{(0)\to(1)}$	time to transition head joints from initial to final positions

Table 22: Parameters of the "thumbs up" and "thumbs down" gestures.

Parameter index	Parameter symbol	Meaning
P_1	$x_8^{(0)}$	initial position of thumb opposition joint
P_2	$x_8^{(1)}$	final position of thumb opposition joint
P_3	$\dot{x}_8$	velocity of thumb opposition joint
P_4	$t_{(0)\to(1)}$	time to transition arm joints from initial to final positions
P_5	$t_{(1)\to(0)}$	time to transition arm joints from final to initial positions

B.2.3 *Human Questionnaire*

To survey human interpretation of the robot movements, we present people with a five-level Likert questionnaire (see Fig. 77). The robot displays gestures selected from the library of Table 18 according to an order which is computed online (see Sec. B.2.5), and we ask subjects to rate their level of agreement to a number of statements relative to social attitudes:

- "This gesture expresses agreement."
- "This gesture expresses anger."
- "This gesture expresses distraction."
- "This gesture expresses approval."
- "This gesture expresses disapproval."

For each displayed movement, we ask people to rate each of the above statements with a score ranging from 1 (strongly disagree) to 5 (strongly agree). For each examined gesture–parameter–value we thus have a vector of scores r_l, where $l = 1, \ldots, L$ is the attitude index. We define the *normalized Likert score* as

$$c_l = \frac{1}{\sum_{i=1}^{L} r_i} r_l, \quad l = 1, \ldots, L, \tag{28}$$

which is a new score derived from r_l, but such that the elements $c_1, \ldots, c_L$ sum to unity. For example, if a vector of scores is $[5 \quad 1 \quad 2 \quad 4 \quad 1]$, then its normalized version is $[5/13 \quad 1/13 \quad 2/13 \quad 4/13 \quad 1/13]$.

B.2.4 *Probabilistic Model of Gesture–Attitude Matches*

We will now describe how to model the questionnaire scores provided by human subjects (see Sec. B.2.3) in a Bayesian Network composed of two nodes: A (parameterized gesture) $\to$ E (attitude).

Node A includes a *fixed* gesture–parameter–value tuple combining a gesture type, a gesture-specific parameter and a possible value for it, as follows: it contains the specific type of gesture $T = t_i$, where $i = 1, \ldots, M$ (M: number of possible gestures), as in Table 18, its parameters values $V_{ij} = \{v_{ijk}\}, j = 1, \ldots, P_i$ (P_i: number of parameters that describe gesture i, as in Tables 19–22), $k = 1, \ldots, K_{ij}$ (K_{ij}: number of possible values of parameter j for gesture i). For notational convenience, we use a unique discrete index $n = 1, \ldots, N$ to count all the possible gesture–parameter–value 3-tuples, thus incorporating the indexes i, j, k like this:

$$n = 1, \ldots, \overbrace{\sum_{i=1}^{M} \sum_{j=1}^{P_i} K_{ij}}^{N}, \quad \begin{aligned} &\forall i = 1, \ldots, M, \\ &\forall j = 1, \ldots, P_i, \\ &\forall k = 1, \ldots, K_{ij}. \end{aligned} \tag{29}$$

Node E encodes a pre-defined *set* of possible attitudes $e_l, l = 1, \ldots, L$, corresponding to the last column of Table 18.

The probability distribution $P(E \mid A)$ is modeled as a multinomial distribution $P(E = e_l \mid A = a_n) = \boldsymbol{\theta}_{ln}$, where θ are the Bayesian Network probability weights[2], l is the attitude index, n is the gesture–parameter–value index and $\sum_l \theta_{ln} = 1$ for each a_n:

$$P(E \mid A) = \begin{bmatrix} \theta_{11} & \cdots & \theta_{L1} \\ \theta_{12} & \cdots & \theta_{L2} \\ \vdots & \ddots & \vdots \\ \theta_{1N} & \cdots & \theta_{LN} \end{bmatrix}. \tag{30}$$

We have modeled a Bayesian Network from N multinomial tables, one for each gesture–parameter–value a_n (each row of the matrix in (30)), expressing the corresponding distribution of attitude perceived by human users.

Furthermore, we express each weight θ_{ln} of (30) as a fractional expression:

$$\theta_{ln} = \frac{s_{ln}}{\#a_n}, \tag{31}$$

where s_{ln} is the cumulative normalized score of attitude l to gesture–parameter–value tuple n (see (28)), and $\#a_n$ is the total number of cases where gesture–parameter–value tuple n was shown.

We assume that the structure of the Bayesian Network is given, and we focus on estimating (updating) the weights $\boldsymbol{\theta}$ by using data coming from human-provided social attitude scores. We keep a Probability

[2]Recall from p. 161 that we call *weights* the Bayesian Network probabilities which specify the Conditional Probability Distributions (CPDs) associated to the network. Those weights are usually called "parameters" in machine learning literature, but we avoid that term because we already employ it for robotic gesture parameters.

Density Function (PDF) over possible weight values, and we assume independence between weights [TK00], which allows us to represent the joint distribution $P(\boldsymbol{\theta})$ as a set of independent multinomial distributions, one for each gesture–parameter–value case.

B.2.5 *Active Learning Algorithm*

The scores that result from the human survey described in Sec. B.2.3 are sent to an *active* Bayesian Network learning program that learns (updates) the weights of the network, as proposed by [TK00]. In the active learning framework, the learner has the ability to guide the instances it gets, by querying for a particular input rather than proceeding randomly or sequentially from a set. In particular, in an unsupervised learning context, the system can request information in regions where the probability distribution that models the data is currently uninformative.

We will now define some quantities necessary for the algorithm, and we will describe the *selection step* as well as the actual *update step*, in accordance to Fig. 77.

For one gesture–parameter–value tuple a_n (i.e., one row of (30)), we denote its *entropy* as

$$H(\boldsymbol{\theta}_n) = -\sum_{l=1}^{L} \theta_{ln} \log(\theta_{ln}), \tag{32}$$

where log is the natural logarithm (the base of the logarithm does not affect the results), and θ_{ln} is one weight of the Bayesian Network (BN) as in (31) (i.e., one entry of the matrix in (30)).

The *expected posterior entropy* of one a_n tuple is computed [TK00, Eq. 1] by averaging the entropies that would arise from all particular choices of vectorial Likert scores ($r = 1, \ldots, R^L$, where L is the number of attitudes or elements in the vectorial scores, and R are the possible Likert levels), weighted by the probability of each choice, $P(r)$:

$$H'(\boldsymbol{\theta}_n) = \sum_{r=1}^{R^L} P(r) \left(-\sum_{l=1}^{L} \theta_{ln}^{(r)} \log\left(\theta_{ln}^{(r)}\right) \right), \tag{33}$$

$$\theta_{ln}^{(r)} = \frac{s_{ln} + c_{lr}}{\#a_n + 1}. \tag{34}$$

However, (33) and (34) are computationally costly due to the exponential number R^L of possible $\boldsymbol{c}$ score choices.

During the selection step we thus make a simplifying assumption: *we only consider score vectors where the answer is fully polarized to one attitude, and all the other ones are zero.* This is based on the empirical observation that our interviewed subjects generally lean towards attributing one clear attitude to a gesture, disregarding all the other ones. The benefit of this assumption is that it makes the expected

posterior entropy of (33) tractable. As a consequence, the prior $P(r)$ can now be taken to be the current weight value θ_{ln} (i.e., before the update), and we can write:

$$H''(\boldsymbol{\theta}_n) = \sum_{r=1}^{R} \theta_{ln} \left(- \sum_{l=1}^{L} \theta_{ln}^{(r)} \log \left(\theta_{ln}^{(r)} \right) \right), \tag{35}$$

$$\theta_{ln}^{(r)} = \frac{s_{ln} + \delta_{lr}}{\#a_n + 1}, \tag{36}$$

where δ_{lr} is the Kronecker delta:

$$\delta_{lr} = \begin{cases} 0 & l \neq r \\ 1 & l = r. \end{cases} \tag{37}$$

In (35) and (36), $\theta_{ln}^{(r)}$ is the "imagined" version of a weight (see (31)), computed according to the following update rule: we multiply the previous value θ_{ln} by the counter $\#a_n$ (number of previous experiments with $A = a_n$), we sum the maximum scores obtainable ((37)), and we divide by the incremented counter $\#a_n + 1$.

Finally, the *entropy gain* is the difference between the entropy before and after learning, or equivalentely, between the current entropy (i.e., before applying the learning step) and the expected posterior entropy after a trial [TK00, Eq. 7]:

$$H_{\text{gain}}(\boldsymbol{\theta}_n) = H(\boldsymbol{\theta}_n) - H''(\boldsymbol{\theta}_n). \tag{38}$$

Selection step. To select which is the most convenient parameterized gesture a_n^* to display from the library (i.e., which row of (30)), we measure the entropy gain of all the rows and we select the row that maximizes such quantity:

$$a_n^* = \arg\max_n H_{\text{gain}}(\boldsymbol{\theta}_n). \tag{39}$$

Rather than querying every person for the same entire sequence of robot gestures in the entire ordered database, the learner selects the next query (row a_n) using probability theory in an efficient way: efficient in the sense of reduced number of queries, and reduced overall time spent doing the experiment for an interview subject. In addition to the maximization criterion of (39), we also employ these heuristics:

NO REPETITIONS: we prevent the system from showing the robot gesture that it showed one iteration before;

RANDOMIZATION: if there are more than one winning a_n^* with the same entropy gain (e.g., when starting the experiment with a uniform prior), select one of them at random.

Update step. After having shown the robot performing the chosen parameterized gesture a_n to an interviewed subject, we obtain the vector

of questionnaire answers, we normalize it ($\boldsymbol{c}$, see (28)) and we update the weights of the model as follows:

$$\frac{s_{ln}}{\#a_n} \leftarrow \frac{s_{ln} + c_l}{\#a_n + 1}, \quad \forall l = 1, \ldots, L, \tag{40}$$

where the left-hand side is the previous score θ_{ln} (see (31)).

The main advantage of our approach is that the human–robot experiment session is shorter than it would be with an exhaustive search (we do not survey *all* people about *all* the possible $A \rightarrow E$ matches), thus making the survey relatively *brief and interesting*. Other advantages are that the system focuses its effort on the most ambiguous gestures (they are repeated more often, so the main share of queries and information gathering is concentrated on them), and that, by not showing exhaustively all the robot movements to all people, we reduce the wear and tear which affects fragile robot parts (e.g., steel tendons).

B.3 EXPERIMENTAL RESULTS

The proposed system maintains a multinomial map from (parameterized) robot gestures to perceived human attitude, as described in Sec. B.2.

In the remainder of this appendix, we assume a *simplification* with respect to the general formulation of Sec. B.2.2: we fix the number of parameters to be 1 for all gesture types, and the number of values (discretization bins) to be 1 for all parameters. This restriction is imposed for practical reasons: (i) from the machine learning perspective, to reduce the number of trials required to appreciate a learning behavior by the system, (ii) from the robotics perspective, to reduce the usage of the robot and the number of movements, especially of the arms and hands, having observed that some metal cables can break frequently when used many times with high accelerations, causing downtime and annoyance, (iii) from the human subjects perspective, to keep the time spent interviewing each human subject to a minimum, around 10 to 15 minutes per person, after which the possibility of the interviewee becoming bored or tired increases.

By forcing the number of gesture parameters to be equal for all gesture types ($P_i \equiv P \quad \forall i$), and the number of possible value discretization bins to be equal for all gesture parameters ($K_{ij} \equiv K \quad \forall i \quad \forall j$), the expression in (29) that enumerates the parameterized robot gestures becomes

$$n = 1, \ldots, \overbrace{\sum_{i=1}^{M} \sum_{j=1}^{P} K}^{N},$$

where P is the set of parameters for all gesture types, and K is the set of discretized values permitted for all gesture parameters.

Furthermore, because we also impose that $P = K = 1$, the expression further simplifies to

$$n = 1, \ldots, M = N,$$

which means that the multinomial map of (30) now consists of a matrix with M rows and L columns, where M is the number of robot gestures and L is the number of human attitudes.

Before starting the interview session, we initialize the matrix of the Bayesian Network weights (see (30) and (31)) to be uniform, considering the 5 robot gestures and human attitudes listed in Table 18:

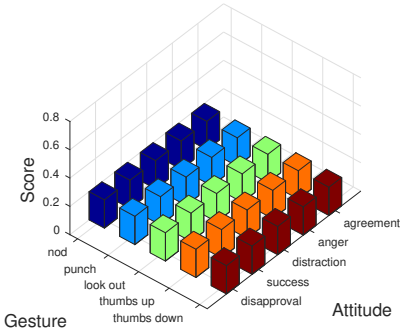
$$\begin{aligned} P(E|A) &= \boldsymbol{\theta}_{ln} \\ &= \frac{\boldsymbol{s}_{ln}}{\#a_n} \\ &= \begin{bmatrix} 0.2/1 & \cdots & 0.2/1 \\ 0.2/1 & \cdots & 0.2/1 \\ \vdots & \ddots & \vdots \\ 0.2/1 & \cdots & 0.2/1 \end{bmatrix}, \end{aligned}$$

where each row represents a robot gesture and each column the match of that gesture to an attitude. These initial values are visualized in Fig. 83a.

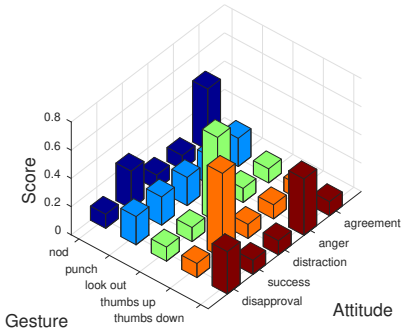
Then, we ask interviewed subjects to sit in front of the robot, to observe one robot movement (chosen online by the learner, initially randomly). We ask users to rank each movement with social attitude scores and we feed the resulting score vector into the system. The active learning system will then choose the next movement to display, and the process is repeated. The matrix starts being updated as the system gets more and more answers, as shown in Fig. 83.

In total we survey 20 people: to each of them we show 5 (± 1) movements and in the end we obtain the mapping displayed in Fig. 83f. It shows that gesture 1 ("nod") was the one with the clearest gesture–attitude correspondence; it soon acquired high "agreement" and "approval" scores, i.e., the system rapidly received information about it (the gesture's expected entropy gain as in (38) got lower) and was not bothered to query it again: it was displayed less time than the others. This also tells us that this particular robot gesture design was good. Gesture 3 ("look out") also resulted in a decent "distraction" score. The remaining gestures performed poorly, in fact several subjects were puzzled about them. Possible reasons are: (i) robot gesture design has to be improved; (ii) the iCub humanoid does not have mechanical capabilities (e.g., accelerations are limited) to convey that attitude solely with movement (excluding facial expressions); (iii) the questionnaire has to be adjusted, as some of the possible attitudes have an overlap, and the ground truth is too easy for people to guess.

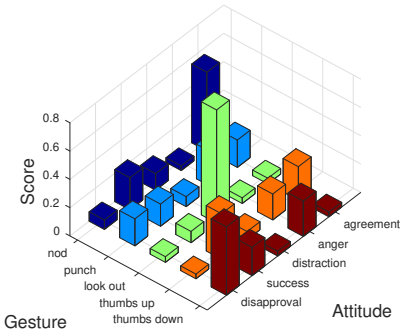
Fig. 84 shows the evolution of the entropy-related quantities during the human survey. The most significant aspect is the entropy gain (in



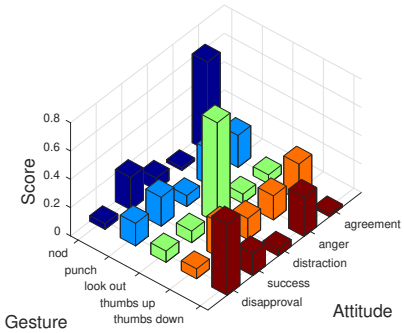
(a) Initial matches.



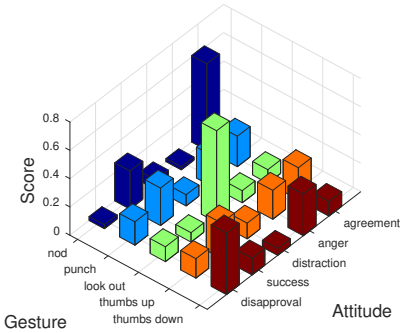
(b) Matches after interviewing 1 person.



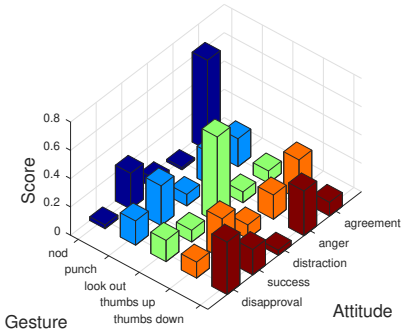
(c) Matches after interviewing 5 people.



(d) Matches after interviewing 10 people.

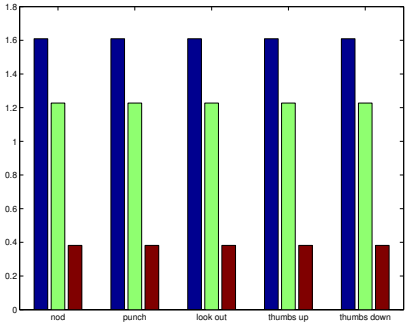


(e) Matches after interviewing 15 people.

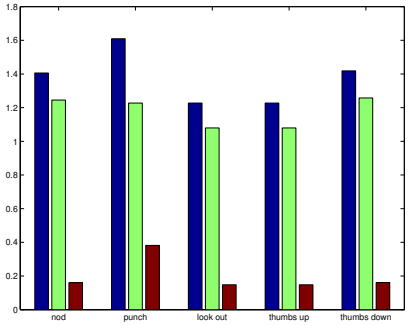


(f) Final matches after interviewing 20 people.

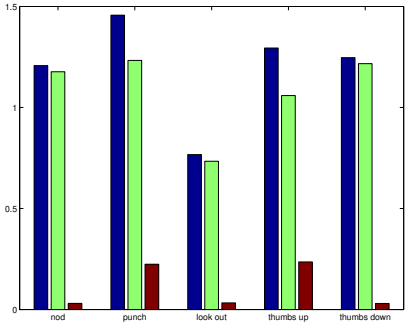
Figure 83: Temporal evolution of the gesture–attitude matches as the human survey is carried out and the questionnaire answers are fed into the active learning system.



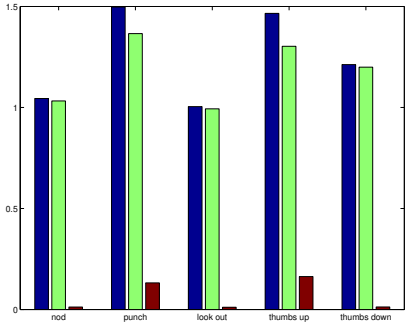
(a) Initial entropies (uniform distribution).



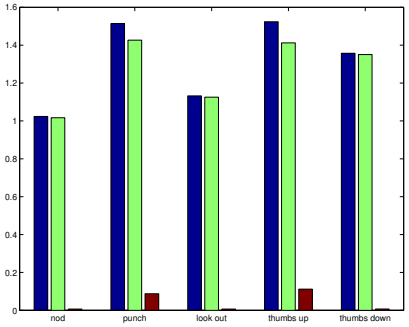
(b) Entropies after interviewing 1 person.



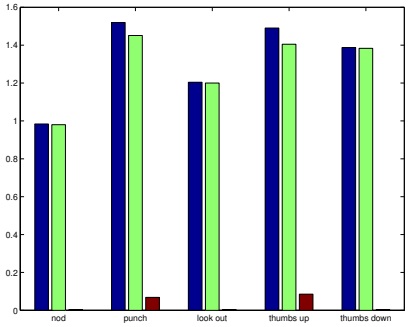
(c) Entropies after interviewing 5 people.



(d) Entropies after interviewing 10 people.



(e) Entropies after interviewing 15 people.



(f) Final entropies after interviewing 20 people.

Figure 84: Temporal evolution of the entropy quantities as the human survey is carried out and the questionnaire answers are fed into the active learning system.

red), which underscores how head movements are clear and unanimous, whereas arm movements are still considered ambiguous by the active learner at the end of the experiment.

Fig. 85 also displays the temporal evolution of entropy-related quantities during the human survey, this time sorting by robot gesture type. This figure highlights which robot gestures managed to convey the ground truth social attitude *clearly* and *quickly* (i.e., after about 5 interviews): "nod" (Fig. 85a) and "look out" (Fig. 85c) are such gestures, because their entropy gain bar quickly decreased to a negligible value. By contrast, other gestures which involve the use of robot arms and hands appear confusing, as their expected entropy gain remained high even after concluding all the interviews: for example, this is the case for "punch" (Fig. 85b) and "thumbs up" (Fig. 85d).

B.4 HUMAN STUDY DATA

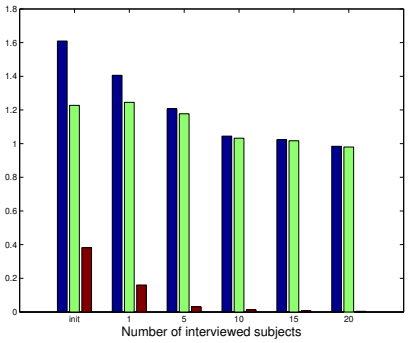
Table 23 shows demographic information about the people surveyed for the human experiment. The proportion between male and female subjects is even (10/10), as is the one between technology experts and non-experts (10/10). None of the people interviewed were roboticists or had interacted with robots at length before.

B.5 CONCLUSIONS AND FUTURE WORK

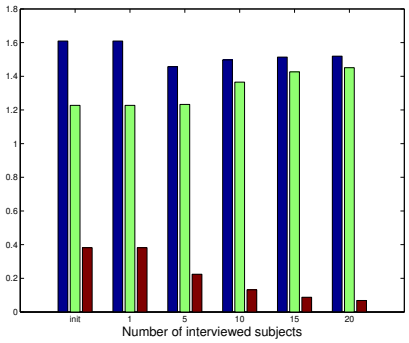
We address the problem of communicating social attitudes with a humanoid robot *without using the facial features* but employing movements of head, arms and torso. The proposed method is described in Sec. B.2 and can be summarized as follows: (i) design a library of simple robot movement gestures corresponding to a ground truth of attitudes; (ii) initialize a matrix of gesture–attitude matching scores; (iii) using an active learning algorithm and according to the current matches, make the robot display the most ambiguous gesture to human users and survey their social attitude perception of that movement based on a questionnaire; (iv) use the resulting human answer to update the gesture–attitude probabilistic matrix; (v) repeat the two previous steps until the learning algorithm has produced meaningful correspondences in the matching scores, or until there are no more subjects to interview.

By looking at the gesture–attitude correspondences obtained with human answers and with our model, we can reason about which robot body gestures are expressive, and also which movements should be performed by the robot in order to transmit a desired social attitude.

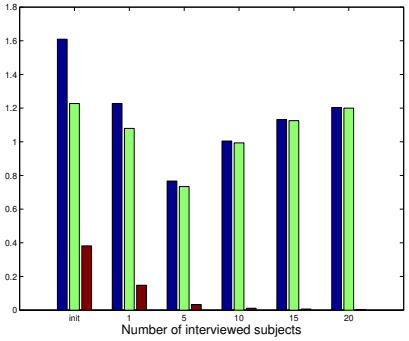
As opposed to a naive approach (e.g., random gesture selection, or querying all human subjects for a fixed sequence of gestures), our learning framework allows to perform human experiments in an optimized way, giving the system the ability to inquire about movements that are ambiguous, showing them more often than easily-perceived ones.



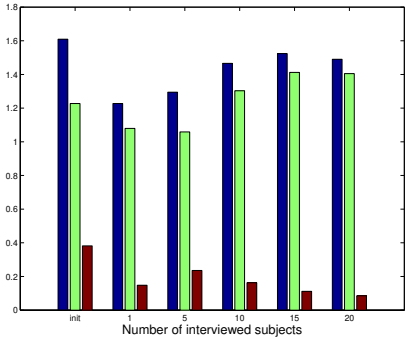
(a) Entropy evolution of the "nod" gesture.



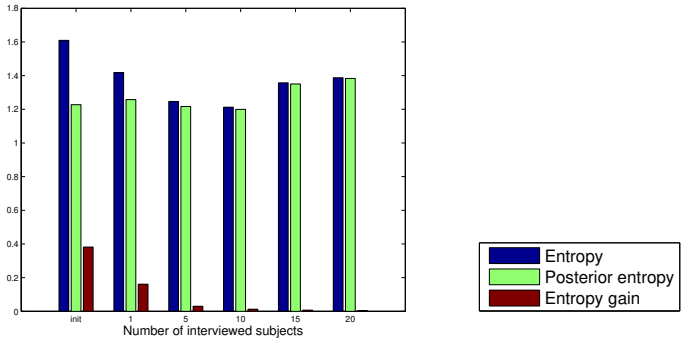
(b) Entropy evolution of the "punch" gesture.



(c) Entropy evolution of the "look out" gesture.



(d) Entropy evolution of the "thumbs up" gesture.



(e) Entropy evolution of the "thumbs down" gesture.

Figure 85: Temporal evolution of the entropy quantities sorted by robot movement, as the human survey is carried out and the questionnaire answers are fed into the active learning system.

Table 23: Demographic data of people surveyed.

Subject	Sex	Age	Technology expert?
1	M	34	Yes
2	F	38	No
3	M	37	Yes
4	F	35	No
5	F	30	No
6	M	23	Yes
7	F	26	No
8	M	30	Yes
9	F	24	No
10	F	25	No
11	M	23	Yes
12	F	34	No
13	M	29	Yes
14	F	28	No
15	M	31	Yes
16	F	28	Yes
17	F	25	No
18	M	32	Yes
19	M	31	No
20	M	31	Yes

In Sec. B.3, our experiments show that people perceive *head* movement attitudes as intended (i.e., similarly to the programmer's ground truth), but that it is difficult to achieve this kind of effective communication by using *arm* motions only (without facial expressions): they yield different responses and users are generally confused about their interpretation.

In terms of future work, the following aspects can be investigated:

- robot gesture design: enrich the corpus of possible gestures (including other limbs, e.g., legs); instead of manually programming robot joint trajectories, acquire them with kinesthetic teaching in compliance mode (i.e., the robot designer manually grabs robot parts and manipulates them);

- machine learning: different initializations of score matches (e.g., uniform versus expert prior knowledge); different optimization strategies other than the maximum-entropy criterion (e.g., artificial neural networks, reinforcement learning); exploit the gesture–parameter–value formulation of robot movements from Sec. B.2.2, in order to identify optimal parameters and values for robot gestures.

BIBLIOGRAPHY

[AG17] Paulo Abelha and Frank Guerin. "Learning How a Tool Affords by Simulating 3D Models from the Web". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2017, pp. 4923–4929. DOI: 10.1109/IROS.2017.8206372 (cit. on pp. 80, 81).

[Ant+16] Alexandre Antunes, Lorenzo Jamone, Giovanni Saponaro, Alexandre Bernardino, and Rodrigo Ventura. "From Human Instructions to Robot Actions: Formulation of Goals, Affordances and Probabilistic Planning". In: *IEEE International Conference on Robotics and Automation*. 2016, pp. 5449–5454. DOI: 10.1109/ICRA.2016.7487757 (cit. on pp. 15, 107).

[Ant+17] Alexandre Antunes, Giovanni Saponaro, Anthony Morse, Lorenzo Jamone, José Santos-Victor, and Angelo Cangelosi. "Learn, Plan, Remember: A Developmental Robot Architecture for Task Solving". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017, pp. 283–289. DOI: 10.1109/DEVLRN.2017.8329819 (cit. on pp. 15, 107).

[AR11] Jake K. Aggarwal and Michael S. Ryoo. "Human Activity Analysis: A Review". In: *ACM Computing Surveys* 43.3 (Apr. 2011), pp. 1–43. DOI: 10.1145/1922649.1922653 (cit. on pp. 34, 50, 143).

[Ara+12] Takaya Araki, Tomoaki Nakamura, Takayuki Nagai, Shogo Nagasaka, Tadahiro Taniguchi, and Naoto Iwahashi. "Online Learning of Concepts and Words Using Multimodal LDA and Hierarchical Pitman–Yor Language Model". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 1623–1630. DOI: 10.1109/IROS.2012.6385812 (cit. on p. 57).

[ASB10] Brenna D. Argall, Eric L. Sauser, and Aude G. Billard. "Tactile Guidance for Policy Refinement and Reuse". In: *IEEE International Conference on Developmental and Learning*. 2010, pp. 7–12. DOI: 10.1109/DEVLRN.2010.5578872 (cit. on p. 164).

[Asf+19] Tamim Asfour, Rüdiger Dillmann, Nikolaus Vahrenkamp, Martin Do, Mirko Wächter, Christian Mandery, Peter Kaiser, Manfred Kröhnert, and Markus Grotz. "The Karlsruhe ARMAR Humanoid Robot Family". In: *Humanoid Robotics:*

A Reference. Springer, 2019, pp. 337–368. DOI: 10.1007/978-94-007-6046-2_23 (cit. on p. 39).

[BA97] Giovanni Berlucchi and Salvatore Aglioti. "The body in the brain: neural bases of corporeal awareness". In: *Trends in Neurosciences* 20 (12 1997), pp. 560–564. DOI: 10.1007/s00221-009-1970-7 (cit. on p. 92).

[BB93] Emily W. Bushnell and J. Paul Boudreau. "Motor Development and the Mind: The Potential Role of Motor Abilities as a Determinant of Aspects of Perceptual Development". In: *Child Development* 64.4 (1993), pp. 1005–1021. DOI: 10.2307/1131323 (cit. on p. 77).

[Bec80] Benjamin B. Beck. *Animal Tool Behavior: The Use and Manufacture of Tools by Animals*. Garland STPM Press, 1980. ISBN: 978-0824071684 (cit. on p. 75).

[Bee+16] Michael Beetz, Raja Chatila, Joachim Hertzberg, and Federico Pecora. "AI Reasoning Methods for Robotics". In: *Springer Handbook of Robotics*. Springer, 2016, pp. 329–356. DOI: 10.1007/978-3-319-32552-1_14 (cit. on p. 110).

[Bei07] Ricardo Beira. "Mechanical Design of an Anthropomorphic Robot Head". Master Thesis in Design Engineering. Lisbon, Portugal: Instituto Superior Técnico, Dec. 2007 (cit. on p. 39).

[Bia13] Nadia Bianchi-Berthouze. "Understanding the Role of Body Movement in Player Engagement". In: *Human–Computer Interaction* 28.1 (2013), pp. 40–75. DOI: 10.1080/07370024.2012.688468 (cit. on p. 163).

[Bil+16] Erik A. Billing, Henrik Svensson, Robert Lowe, and Tom Ziemke. "Finding Your Way from the Bed to the Kitchen: Reenacting and Recombining Sensorimotor Episodes Learned from Human Demonstration". In: *Frontiers in Robotics and AI* 3 (2016), p. 9. DOI: 10.3389/frobt.2016.00009 (cit. on p. 145).

[Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. ISBN: 978-0387310732 (cit. on pp. 19–25, 28, 35, 85, 86, 161).

[BLL11] Concha Bielza, Guangdi Li, and Pedro Larrañaga. "Multi-Dimensional Classification with Bayesian Networks". In: *International Journal of Approximate Reasoning* 52 (2011), pp. 705–727. DOI: 10.1016/j.ijar.2011.01.007 (cit. on p. 27).

[Bou+12] Jean David Boucher, Ugo Pattacini, Amelie Lelong, Gerard Bailly, Frederic Elisei, Sascha Fagel, Peter Ford Dominey, and Jocelyne Ventre-Dominey. "I reach faster when I see you look: Gaze effects in human–human and human–robot face-to-face cooperation". In: *Frontiers in Neurorobotics* 6.May (2012), pp. 1–11. DOI: 10.3389/fnbot.2012.00003 (cit. on p. 164).

[Bre02] Cynthia Breazeal. *Designing Sociable Robots*. MIT Press, 2002. ISBN: 978-0262025102 (cit. on p. 49).

[BRZ06] Celia A. Brownell, Geetha B. Ramani, and Stephanie Zerwas. "Becoming a social partner with peers: Cooperation and social understanding in one- and two-year-olds". In: *Child Development* 77.4 (2006), pp. 803–821. DOI: 10.1111/j.1467-8624.2006.t01-1-.x-i1 (cit. on p. 52).

[BWB08] Andrea Bauer, Dirk Wollherr, and Martin Buss. "Human–Robot Collaboration: A Survey". In: *International Journal of Humanoid Robotics* 05.01 (2008), pp. 47–66. DOI: 10.1142/S0219843608001303 (cit. on pp. 72, 143).

[BWW15] Pablo Barros, Cornelius Weber, and Stefan Wermter. "Emotional expression recognition with a cross-channel convolutional neural network for human–robot interaction". In: *IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 582–587. DOI: 10.1109/HUMANOIDS.2015.7363421 (cit. on p. 162).

[Cac+17] Riccardo Caccavale, Matteo Saveriano, Giuseppe Fontanelli, Fanny Ficuciello, Dongheui Lee, and Alberto Finzi. "Imitation Learning and Attentional Supervision of Dual-Arm Structured Tasks". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017 (cit. on p. 111).

[Cam+96] Lee W. Campbell, David A. Becker, Ali Azarbayejani, Aaron F. Bobick, and Alex Pentland. "Invariant Features for 3-D Gesture Recognition". In: *IEEE Conference on Automatic Face and Gesture Recognition*. 1996, p. 157. DOI: 10.1109/AFGR.1996.557258 (cit. on p. 144).

[CFT16] Vivian Chu, Tesca Fitzgerald, and Andrea L. Thomaz. "Learning Object Affordances by Leveraging the Combination of Human-Guidance and Self-Exploration". In: *ACM/IEEE International Conference on Human–Robot Interaction*. 2016, pp. 221–228. DOI: 10.1109/HRI.2016.7451755 (cit. on pp. 36, 37).

[CH92] Gregory F. Cooper and Edward Herskovitz. "A Bayesian Method for the Induction of Probabilistic Networks from Data". In: *Machine Learning* 9.4 (1992), pp. 309–347. DOI: 10.1007/BF00994110 (cit. on p. 27).

[Che03] Anthony Chemero. "An Outline of a Theory of Affordances". In: *Ecological Psychology* 15.2 (2003), pp. 181–195. DOI: 10.1207/S15326969EC01502_5 (cit. on p. 7).

[CM00] Linda L. Chao and Alex Martin. "Representation of Manipulable Man-Made Objects in the Dorsal Stream". In: *Neuroimage* 12.4 (2000), pp. 478–484. DOI: 10.1006/nimg.2000.0635 (cit. on p. 10).

[CM11] David L. Chen and Raymond J. Mooney. "Learning to Interpret Natural Language Navigation Instructions from Observations". In: *AAAI Conference on Artificial Intelligence*. 2011, pp. 859–865 (cit. on p. 112).

[COK15] Hande Celikkanat, Güren Orhan, and Sinan Kalkan. "A Probabilistic Concept Web on a Humanoid Robot". In: *IEEE Transactions on Autonomous Mental Development* 7.2 (2015), pp. 92–106. DOI: 10.1109/TAMD.2015.2418678 (cit. on p. 34).

[CS03] Silvia Coradeschi and Alessandro Saffiotti. "An introduction to the anchoring problem". In: *Robotics and Autonomous Systems* 43.2 (2003), pp. 85–96. DOI: 10.1016/S0921-8890(03)00021-6 (cit. on pp. 112, 117).

[CS15] Angelo Cangelosi and Matthew Schlesinger. *Developmental Robotics: From Babies to Robots*. MIT Press, 2015. ISBN: 978-0262028011 (cit. on p. 6).

[CT07] Anthony Chemero and Michael T. Turvey. "Gibsonian Affordances for Roboticists". In: *Adaptive Behavior* 15.4 (2007), pp. 473–480. DOI: 10.1177/1059712307085098 (cit. on p. 7).

[Deh+16a] Atabak Dehban, Lorenzo Jamone, Adam R. Kampff, and José Santos-Victor. "A Moderately Large Size Dataset to Learn Visual Affordances of Objects and Tools Using iCub Humanoid Robot". In: *European Conference on Computer Vision*. Workshop on Action and Anticipation for Visual Learning. 2016 (cit. on pp. 13, 47, 99, 101–103).

[Deh+16b] Atabak Dehban, Lorenzo Jamone, Adam R. Kampff, and José Santos-Victor. "Denoising Auto-encoders for Learning of Objects and Tools Affordances in Continuous Space". In: *IEEE International Conference on Robotics and Automation*. 2016, pp. 4866–4871. DOI: 10.1109/ICRA.2016.7487691 (cit. on pp. 13, 47, 141).

[Deh+17] Atabak Dehban, Lorenzo Jamone, Adam R. Kampff, and José Santos-Victor. "A Deep Probabilistic Framework for Heterogeneous Self-Supervised Learning of Affordances". In: *IEEE-RAS International Conference on Humanoid Robots*. 2017, pp. 476–483. DOI: 10.1109/HUMANOIDS.2017.8246915 (cit. on pp. 13, 47, 141).

[DLS13] Anca D. Dragan, Kenton C. T. Lee, and Siddhartha S. Srinivasa. "Legibility and Predictability of Robot Motion". In: *ACM/IEEE International Conference on Human–Robot Interaction*. 2013, pp. 301–308. DOI: 10.1109/HRI.2013.6483603 (cit. on p. 144).

[Dra+15] Anca D. Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S. Srinivasa. "Effects of Robot Motion on Human–Robot Collaboration". In: *ACM/IEEE International Conference on Human–Robot Interaction*. 2015, pp. 51–58. DOI: 10.1145/2696454.2696473 (cit. on pp. 72, 144).

[DS14] Anca D. Dragan and Siddhartha Srinivasa. "Integrating human observer inferences into robot motion planning". In: *Autonomous Robots* 37.4 (2014), pp. 351–368. DOI: 10.1007/s10514-014-9408-x (cit. on p. 144).

[Dua+18] Nuno Duarte, Jovica Tasevski, Moreno Coco, Mirko Raković, and José Santos-Victor. "Action Anticipation: Reading the Intentions of Humans and Robots". In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4132–4139. DOI: 10.1109/LRA.2018.2861569 (cit. on p. 142).

[Dzi+09] Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. "What to do and how to do it: Translating Natural Language Directives into Temporal and Dynamic Logic Representation for Goal Management and Action Execution". In: *IEEE International Conference on Robotics and Automation*. 2009. DOI: 10.1109/ROBOT.2009.5152776 (cit. on p. 112).

[EFE72] Paul Ekman, Wallace V. Friesen, and Phoebe Ellsworth. *Emotion in the Human Face: Guidelines for Research and an Integration of Findings*. Pergamon, 1972. ISBN: 978-0080166438 (cit. on pp. 161, 164).

[Elf+13] Jos Elfring, Sjoerd van den Dries, M. J. G. van de Molengraft, and Maarten Steinbuch. "Semantic world modeling using probabilistic multiple hypothesis anchoring". In: *Robotics and Autonomous Systems* 61.2 (2013), pp. 95–105. DOI: 10.1016/j.robot.2012.11.005 (cit. on pp. 112, 117).

[ETF16] Manfred Eppe, Sean Trott, and Jerome Feldman. "Exploiting Deep Semantics and Compositionality of Natural Language for Human–Robot Interaction". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2016, pp. 731–738. DOI: 10.1109/IROS.2016.7759133 (cit. on p. 112).

[Fan+14] Sean Ryan Fanello, Ugo Pattacini, Ilaria Gori, Vadim Tikhanoff, Marco Randazzo, Alessandro Roncone, Francesca Odone, and Giorgio Metta. "3D Stereo Estimation and Fully Automated Learning of Eye–Hand Coordination in Humanoid Robots". In: *IEEE-RAS International Conference on Humanoid Robots*. 2014, pp. 1028–1035. DOI: 10.1109/HUMANOIDS.2014.7041491 (cit. on p. 142).

[Fit+03] Paul Fitzpatrick, Giorgio Metta, Lorenzo Natale, Sajit Rao, and Giulio Sandini. "Learning About Objects Through Action: Initial Steps Towards Artificial Cognition". In: *IEEE International Conference on Robotics and Automation*. 2003, pp. 3140–3145. DOI: 10.1109/ROBOT.2003.1242073 (cit. on p. 30).

[Fit+14] Paul Fitzpatrick, Elena Ceseracciu, Daniele E. Domenichelli, Ali Paikan, Giorgio Metta, and Lorenzo Natale. "A middle way for robotics middleware". In: *Journal of Software Engineering for Robotics* 5.2 (Sept. 2014), pp. 42–49. DOI: 10.6092/JOSER_2014_05_02_p42 (cit. on p. 39).

[Fog+05] Leonardo Fogassi, Pier Francesco Ferrari, Benno Gesierich, Stefano Rozzi, Fabian Chersi, and Giacomo Rizzolatti. "Parietal Lobe: From Action Organization to Intention Understanding". In: *Science* 308.5722 (2005), pp. 662–667. DOI: 10.1126/science.1106138 (cit. on p. 13).

[FRO14] Jacqueline Fagard, Lauriane Rat-Fischer, and J. Kevin O'Regan. "The emergence of use of a rake-like tool: a longitudinal study in human infants". In: *Frontiers in Psychology* 5 (2014). DOI: 10.3389/fpsyg.2014.00491 (cit. on pp. 8, 77, 79).

[FS17] Arvid Fahlström Myrman and Giampiero Salvi. "Partitioning of Posteriorgrams Using Siamese Models for Unsupervised Acoustic Modelling". In: *Workshop on Grounding Language Understanding*. 2017, pp. 27–31. DOI: 10.21437/GLU.2017-6 (cit. on p. 73).

[Gao+18] Qiaozi Gao, Shaohua Yang, Joyce Chai, and Lucy Vanderwende. "What Action Causes This? Towards Naive Physical Action–Effect Prediction". In: *Annual Meeting of the Association for Computational Linguistics*. 2018, pp. 934–945 (cit. on p. 57).

[Gaz+07] Valeria Gazzola, Giacomo Rizzolatti, Bruno Wicker, and Christian Keysers. "The anthropomorphic brain: the mirror neuron system responds to human and robotic actions". In: *Neuroimage* 35.4 (2007), pp. 1674–1684. DOI: 10.1016/j.neuroimage.2007.02.003 (cit. on pp. 13, 145).

[GC03] Paolo Giudici and Robert Castelo. "Improving Markov Chain Monte Carlo Model Search for Data Mining". In: *Machine Learning* 50.1-2 (2003), pp. 127–158. DOI: 10.1023/A:1020202028934 (cit. on p. 27).

[Gib03] Eleanor J. Gibson. "The World Is So Full of a Number of Things: On Specification and Perceptual Learning". In: *Ecological Psychology* 15.4 (2003), pp. 283–287. DOI: 10.1207/s15326969eco1504_3 (cit. on p. 1).

[Gib14] James J. Gibson. *The Ecological Approach to Visual Perception: Classic Edition*. Originally published in 1979 by Houghton Mifflin Harcourt. Psychology Press, 2014. ISBN: 978-1848725782 (cit. on pp. 1, 7).

[Gib94] Eleanor J. Gibson. *An Odyssey in Learning and Perception*. MIT Press, 1994. ISBN: 978-0262571036 (cit. on p. 77).

[GM92] Melvyn A. Goodale and A. David Milner. "Separate visual pathways for perception and action". In: *Trends in Neurosciences* 15.1 (1992), pp. 20–25. DOI: 10.1016/0166-2236(92)90344-8 (cit. on p. 10).

[Gol09] Robert P. Goldman. "A Semantics for HTN Methods". In: *International Conference on Automated Planning and Scheduling*. 2009, pp. 146–153 (cit. on p. 114).

[Gon+14a] Afonso Gonçalves, João Abrantes, Giovanni Saponaro, Lorenzo Jamone, and Alexandre Bernardino. "Learning Intermediate Object Affordances: Towards the Development of a Tool Concept". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2014, pp. 482–488. DOI: 10.1109/DEVLRN.2014.6983027 (cit. on pp. 14, 76, 82, 87, 93).

[Gon+14b] Afonso Gonçalves, Giovanni Saponaro, Lorenzo Jamone, and Alexandre Bernardino. "Learning Visual Affordances of Objects and Tools through Autonomous Robot Exploration". In: *IEEE International Conference on Autonomous Robot Systems and Competitions*. 2014, pp. 128–133. DOI: 10.1109/ICARSC.2014.6849774 (cit. on pp. 14, 76).

[Har90] Stevan Harnad. "The Symbol Grounding Problem". In: *Physica D: Nonlinear Phenomena* 42.1-3 (1990), pp. 335–346. DOI: 10.1016/0167-2789(90)90087-6 (cit. on pp. 57, 112).

[HD96] Cecil Huang and Adnan Darwiche. "Inference in Belief Networks: A Procedural Guide". In: *International Journal of Approximate Reasoning* 15.3 (1996), pp. 225–263. DOI: 10.1016/S0888-613X(96)00069-2 (cit. on p. 28).

[HDH11] Pascal Haazebroek, Saskia van Dantzig, and Bernhard Hommel. "A computational model of perception and action for cognitive robotics". In: *Cognitive Processing* 12.4 (2011), pp. 355–365. DOI: 10.1007/s10339-011-0408-x (cit. on pp. 109, 110).

[HGC95] David Heckerman, Dan Geiger, and David M. Chickering. "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data". In: *Machine Learning* 20.3 (1995), pp. 197–243. DOI: 10.1023/A:1022623210503 (cit. on pp. 27, 32).

[Hin+12] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97. DOI: 10.1109/MSP.2012.2205597 (cit. on p. 153).

[Hof04] Claes von Hofsten. "An action perspective on motor development". In: *Trends in Cognitive Sciences* 8 (6 2004), pp. 266–272. DOI: 10.1016/j.tics.2004.04.002 (cit. on p. 92).

[Iwa07] Naoto Iwahashi. "Robots that learn language: A developmental approach to situated human–robot conversations". In: *Human Robot Interaction*. InTech, 2007. Chap. 5. DOI: 10.5772/5188 (cit. on p. 52).

[Jam+16] Lorenzo Jamone, Emre Ugur, Angelo Cangelosi, Luciano Fadiga, Alexandre Bernardino, Justus Piater, and José Santos-Victor. "Affordances in Psychology, Neuroscience and Robotics: A Survey". In: *IEEE Transactions on Cognitive and Developmental Systems* (2016). DOI: 10.1109/TCDS.2016.2594134 (cit. on pp. 7, 30, 80).

[Jam10] Davide K. James. "Fetal learning: A critical review". In: *Infant and Child Cevelopment* 19 (2010), pp. 45–54. DOI: 10.1002/icd.653 (cit. on p. 77).

[Jen96] Finn V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1996. ISBN: 978-0387915029 (cit. on p. 21).

[JI13] Raghvendra Jain and Tetsunari Inamura. "Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools". In: *Artificial Life and Robotics* 18.1-2 (2013), pp. 95–103. DOI: 10.1007/s10015-013-0105-1 (cit. on pp. 80, 81).

[JKS13] Yun Jiang, Hema Koppula, and Ashutosh Saxena. "Hallucinated Humans as the Hidden Context for Labeling 3D Scenes". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2993–3000. DOI: 10.1109/CVPR.2013.385 (cit. on p. 30).

[Jos00] Rhawn Joseph. "Fetal Brain Behavior and Cognitive Development". In: *Developmental Review* 20 (1 2000), pp. 81–98. DOI: 10.1006/drev.1999.0486 (cit. on p. 92).

[KAA11] Cem Keskin, Oya Aran, and Lale Akarun. "Hand Gesture Analysis". In: *Computer Analysis of Human Behavior*. Springer, 2011, pp. 125–149. DOI: 10.1007/978-0-85729-994-9_6 (cit. on p. 144).

[KAC17] Sotaro Kita, Martha W. Alibali, and Mingyuan Chu. "How Do Gestures Influence Thinking and Speaking? The Gesture-for-Conceptualization Hypothesis". In: *Psychological Review* 124.3 (2017), pp. 245–266. DOI: 10.1037/rev0000059 (cit. on p. 144).

[Kan+00] Eric R. Kandel, James H. Schwartz, Thomas M. Jessell, Steven A. Siegelbaum, and A. James Hudspeth. *Principles of Neural Science*. Fifth edition. McGraw-Hill, 2000. ISBN: 978-0071390118 (cit. on p. 12).

[Kan+03] Takayuki Kanda, Hiroshi Ishiguro, Michita Imai, and Tetsuo Ono. "Body Movement Analysis of Human–Robot Interaction". In: *International Joint Conference on Artificial Intelligence*. 2003, pp. 177–182 (cit. on p. 144).

[KB13] Andrea Kleinsmith and Nadia Bianchi-Berthouze. "Affective Body Expression Perception and Recognition: A Survey". In: *IEEE Transactions on Affective Computing* 4.1 (2013), pp. 15–33. DOI: 10.1109/T-AFFC.2012.16 (cit. on p. 163).

[KGS13] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. "Learning Human Activities and Object Affordances from RGB-D Videos". In: *International Journal of Robotics Research* 32.8 (2013), pp. 951–970. DOI: 10.1177/0278364913478446 (cit. on p. 57).

[Kil+09] James M. Kilner, Alice Neal, Nikolaus Weiskopf, Karl J. Friston, and Chris D. Frith. "Evidence of Mirror Neurons in Human Inferior Frontal Gyrus". In: *Journal of Neu-*

roscience 29.32 (2009), pp. 10153–10159. DOI: 10.1523/JNEUROSCI.2668-09.2009 (cit. on p. 13).

[KKL14] George Konidaris, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. "Constructing Symbolic Representations for High-Level Planning". In: *AAAI Conference on Artificial Intelligence*. 2014 (cit. on p. 115).

[KKL18] George Konidaris, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. "From Skills to Symbols: Learning Symbolic Representations for Abstract High-Level Planning". In: *Journal of Artificial Intelligence Research* 61 (2018), pp. 215–289. DOI: 10.1613/jair.5575 (cit. on p. 115).

[KL11] Leslie Pack Kaelbling and Tomás Lozano-Pérez. "Hierarchical Task and Motion Planning in the Now". In: *IEEE International Conference on Robotics and Automation*. 2011, pp. 1470–1477. DOI: 10.1109/ICRA.2011.5980391 (cit. on p. 114).

[KNY02] Hideki Kozima, Cocoro Nakagawa, and Hiroyuki Yano. "Emergence of imitation mediated by objects". In: *International Conference on Epigenetic Robotics*. 2002, pp. 59–61 (cit. on p. 30).

[Köh17] Wolfgang Köhler. *The Mentality of Apes*. Originally published in 1925 by Brace & Company Inc. Routledge, 2017. ISBN: 978-1351294959 (cit. on pp. 79, 80).

[Krü+11] Norbert Krüger, Christopher Geib, Justus Piater, Ronald Petrick, Mark Steedman, Florentin Wörgötter, Aleš Ude, Tamim Asfour, Dirk Kraft, Damir Omrčen, Alejandro Agostini, and Rüdiger Dillmann. "Object–Action Complexes: Grounded Abstractions of Sensory–Motor Processes". In: *Robotics and Autonomous Systems* 59.10 (2011), pp. 740–757. DOI: 10.1016/j.robot.2011.05.009 (cit. on p. 113).

[KS16] Hema S. Koppula and Ashutosh Saxena. "Anticipating Human Activities Using Object Affordances for Reactive Robotic Response". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (2016), pp. 14–29. DOI: 10.1109/TPAMI.2015.2430335 (cit. on p. 36).

[KYL17] Sangwook Kim, Zhibin Yu, and Minho Lee. "Understanding human intention by connecting perception and action learning in artificial agents". In: *Neural Networks* 92 (2017), pp. 29–38. DOI: 10.1016/j.neunet.2017.01.009 (cit. on p. 145).

[Lak+17] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. "Building Machines That Learn and Think Like People". In: *Behavioral and Brain*

Sciences 40 (2017). DOI: 10.1017/S0140525X16001837 (cit. on p. 57).

[Lal+13] Stéphane Lallée, Katharina Hamann, Jasmin Steinwender, Felix Warneken, Uriel Martinez, Hector Barron-Gonzales, Ugo Pattacini, Ilaria Gori, Maxime Petit, Giorgio Metta, Paul Verschure, and Peter Ford Dominey. "Cooperative Human Robot Interaction Systems: IV. Communication of Shared Plans with Naïve Humans using Gaze and Speech". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 129–136. DOI: 10.1109/IROS.2013.6696343 (cit. on p. 30).

[LC11] Jamy Li and Mark Chignell. "Communication of Emotion in Social Robots through Simple Head and Arm Movements". In: *International Journal of Social Robotics* 3.2 (2011), pp. 125–142. DOI: 10.1007/s12369-010-0071-x (cit. on pp. 162, 164, 165).

[Lem+12] Séverin Lemaignan, Raquel Ros, E. Akin Sisbot, Rachid Alami, and Michael Beetz. "Grounding the Interaction: Anchoring Situated Discourse in Everyday Human–Robot Interaction". In: *International Journal of Social Robotics* 4.2 (2012), pp. 181–199. DOI: 10.1007/s12369-011-0123-x (cit. on pp. 112, 117).

[Lem+17] Séverin Lemaignan, Mathieu Warnier, E. Akin Sisbot, Aurélie Clodic, and Rachid Alami. "Artificial cognition for social human–robot interaction: An implementation". In: *Artificial Intelligence* 247 (2017), pp. 45–69. DOI: 10.1016/j.artint.2016.07.002 (cit. on p. 110).

[Lev+18] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. "Learning hand–eye coordination for robotic grasping with deep learning and large-scale data collection". In: *International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436. DOI: 10.1177/0278364917710318 (cit. on p. 141).

[LG13] Michele A. Lobo and James C. Galloway. "The onset of reaching significantly impacts how infants explore both objects and their bodies". In: *Infant Behavior and Development* 36.1 (2013), pp. 14–24. DOI: 10.1016/j.infbeh.2012.09.003 (cit. on pp. 77, 79).

[LK14] Tomás Lozano-Pérez and Leslie Pack Kaelbling. "A constraint-based method for solving sequential manipulation planning problems". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014, pp. 3684–3691. DOI: 10.1109/IROS.2014.6943079 (cit. on p. 114).

[Llo82] Stuart Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489 (cit. on p. 31).

[LMM07] Manuel Lopes, Francisco S. Melo, and Luis Montesano. "Affordance-based imitation learning in robots". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007, pp. 1015–1021. DOI: 10.1109/IROS.2007.4399517 (cit. on p. 30).

[Loc00] Jeffrey J. Lockman. "A Perception–Action Perspective on Tool Use Development". In: *Child Development* 71.1 (2000), pp. 137–144. DOI: 10.1111/1467-8624.00127 (cit. on pp. 77, 79, 84).

[Lop+04] Manuel Lopes, Ricardo Beira, Miguel Praça, and José Santos-Victor. "An Anthropomorphic Robot Torso for Imitation: Design and Experiments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2004, pp. 661–667. DOI: 10.1109/IROS.2004.1389428 (cit. on p. 31).

[Lop+09] Manuel Lopes, Francisco S. Melo, Ben Kenward, and José Santos-Victor. "A computational model of social-learning mechanisms". In: *Adaptive Behavior* 17.6 (2009), pp. 467–483. DOI: 10.1177/1059712309342757 (cit. on p. 145).

[Low99] David G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *IEEE International Conference on Computer Vision*. Vol. 2. 1999, pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410 (cit. on p. 36).

[Loz+87] Tomás Lozano-Pérez, Joseph L. Jones, Emmanuel Mazer, Patrick A. O'Donnell, W. Eric L. Grimson, Pierre Tournassoud, and Alain Lanusse. "Handey: A Robot System that Recognizes, Plans, and Manipulates". In: *IEEE International Conference on Robotics and Automation*. 1987, pp. 843–849. DOI: 10.1109/ROBOT.1987.1087847 (cit. on p. 114).

[LS05] Manuel Lopes and José Santos-Victor. "Visual Learning by Imitation with Motor Representations". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 35.3 (2005), pp. 438–449. DOI: 10.1109/TSMCB.2005.846654 (cit. on p. 30).

[LS07] Manuel Lopes and José Santos-Victor. "A Developmental Roadmap for Learning by Imitation in Robots". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37.2 (Apr. 2007), pp. 308–321. DOI: 10.1109/TSMCB.2006.886949 (cit. on p. 30).

[LS88] Steffen L. Lauritzen and David J. Spiegelhalter. "Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems". In: *Journal of the Royal Statistical Society, Series B: Methodological* 50.2 (1988), pp. 157–224. DOI: 10.2307/2345762 (cit. on p. 28).

[LT10] Tobias Lang and Marc Toussaint. "Planning with Noisy Probabilistic Relational Rules". In: *Journal of Artificial Intelligence Research* 39 (2010), pp. 1–49. DOI: 10.1613/jair.3093 (cit. on pp. 114, 115, 121).

[Lun+03] Max Lungarella, Giorgio Metta, Rolf Pfeifer, and Giulio Sandini. "Developmental robotics: a survey". In: *Connection Science* 15.4 (2003), pp. 151–190. DOI: 10.1080/09540090310001655110 (cit. on p. 6).

[MA07] Sushmita Mitra and Tinku Acharya. "Gesture Recognition: A Survey". In: 37.3 (2007), pp. 311–324. DOI: 10.1109/TSMCC.2007.893280 (cit. on p. 144).

[Mat+12] Cynthia Matuszek, Nicholas FitzGerald, Luke Zettlemoyer, Bo Liefeng, and Dieter Fox. "A Joint Model of Language and Perception for Grounded Attribute Learning". In: *International Conference on Machine Learning*. 2012, pp. 1671–1678. ISBN: 978-1450312851 (cit. on p. 112).

[Mat+13] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. "Learning to Parse Natural Language Commands to a Robot Control System". In: *Experimental Robotics*. Vol. 88. 2013, pp. 515–529. DOI: 10.1007/978-3-319-00065-7 (cit. on p. 112).

[Mat+14] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. "Learning from Unscripted Deictic Gesture and Language for Human–Robot Interactions". In: *AAAI Conference on Artificial Intelligence*. 2014, pp. 2556–2563 (cit. on p. 57).

[MC00] Ulrich Müller and Jeremy I. M. Carpendale. "The role of social interaction in Piaget's theory: language for social cooperation and social cooperation for language". In: *New Ideas in Psychology* 18.2 (2000), pp. 139–156. DOI: 10.1016/S0732-118X(00)00004-0 (cit. on p. 50).

[MC16] Anthony F. Morse and Angelo Cangelosi. "Why Are There Developmental Stages in Language Learning? A Developmental Robotics Model of Language Development". In: *Cognitive Science* 41 (2016), pp. 32–51. DOI: 10.1111/cogs.12390 (cit. on pp. 13, 47, 57).

[MC99] Lynn S. Messing and Ruth Campbell. *Gesture, Speech, and Sign*. Oxford University Press, 1999. ISBN: 978-0198524519 (cit. on pp. 51, 144).

[McN96] David McNeill. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, 1996. ISBN: 978-0226561349 (cit. on pp. 51, 144).

[Met+10] Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, Alexandre Bernardino, and Luis Montesano. "The iCub humanoid robot: An open-systems platform for research in cognitive development". In: *Neural Networks* 23.8 (2010), pp. 1125–1134. DOI: 10.1016/j.neunet.2010.08.010 (cit. on p. 39).

[MFN06] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. "YARP: Yet Another Robot Platform". In: *International Journal on Advanced Robotics Systems* 3.1 (2006), pp. 43–48. DOI: 10.5772/5761 (cit. on p. 39).

[Mis+16] Dipendra K. Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. "Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions". In: *International Journal of Robotics Research* 35.1-3 (2016), pp. 281–300. DOI: 10.1177/0278364915602060 (cit. on p. 112).

[Mol+18] Bogdan Moldovan, Plinio Moreno, Davide Nitti, José Santos-Victor, and Luc De Raedt. "Relational Affordance for Multiple-Object Manipulation". In: *Autonomous Robots* 42.1 (2018), pp. 19–44. DOI: 10.1007/s10514-017-9637-x (cit. on pp. 80, 82, 113).

[Mon+08] Luis Montesano, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. "Learning Object Affordances: From Sensory–Motor Maps to Imitation". In: *IEEE Transactions on Robotics* 24.1 (2008), pp. 15–26. DOI: 10.1109/TRO.2007.914848 (cit. on pp. 4, 5, 21, 28, 30, 31, 33, 39, 82, 83, 139).

[Mon+10] Luis Montesano, Manuel Lopes, Francisco S. Melo, Alexandre Bernardino, and José Santos-Victor. "A Computational Model of Object Affordances". In: *Advances in Cognitive Systems*. IET Publishers, 2010. Chap. 5, pp. 87–126. DOI: 10.1049/PBCE071E_ch5 (cit. on p. 30).

[Mou+18] Clément Moulin-Frier, Tobias Fischer, Maxime Petit, Grégoire Pointeau, Jordi-Ysard Puigbo, Ugo Pattacini, Sock Ching Low, Daniel Camilleri, Phuong Nguyen, Matej Hoffmann, Hyung Jin Chang, Martina Zambelli, Anne-Laure

Mealier, Andreas Damianou, Giorgio Metta, Tony J. Prescott, Yiannis Demiris, Peter Ford Dominey, and Paul F. M. J. Verschure. "DAC-h3: A Proactive Robot Cognitive Architecture to Acquire and Express Knowledge About the World and the Self". In: *IEEE Transactions on Cognitive and Developmental Systems* 10.4 (2018), pp. 1005–1022. DOI: 10.1109/TCDS.2017.2754143 (cit. on pp. 110–112).

[MP16] Dimitris Mavroeidis and Katerina Pastra. *The PRAXICON database, version 1.0*. Tech. rep. CSRI, 2016. URL: https://github.com/CSRI/PraxiconDB (cit. on pp. 35, 112, 116).

[MS10] Alicia P. Melis and Dirk Semmann. "How is human cooperation different?" In: *Philosophical Transactions of the Royal Society, Series B: Biological Sciences* 365.1553 (2010), pp. 2663–2674. DOI: 10.1098/rstb.2010.0157 (cit. on p. 52).

[MT08] Reinhard Moratz and Thora Tenbrink. "Affordance-Based Human–Robot Interaction". In: *Towards Affordance-Based Robot Control*. Springer, 2008, pp. 63–76. DOI: 10.1007/978-3-540-77915-5_5 (cit. on pp. 30, 33).

[MTN18] Tanis Mar, Vadim Tikhanoff, and Lorenzo Natale. "What Can I Do With This Tool? Self-Supervised Learning of Tool Affordances From Their 3-D Geometry". In: *IEEE Transactions on Cognitive and Developmental Systems* 10.3 (2018), pp. 595–610. DOI: 10.1109/TCDS.2017.2717041 (cit. on pp. 80, 81, 142).

[Mur12] Kevin Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. ISBN: 978-0262018029 (cit. on pp. 89, 150).

[MYA95] David Madigan, Jeremy York, and Denis Allard. "Bayesian Graphical Models for Discrete Data". In: *International Statistical Review* 63.2 (1995), pp. 215–232. DOI: 10.2307/1403615 (cit. on p. 27).

[Nat+19] Lorenzo Natale, Chiara Bartolozzi, Francesco Nori, Giulio Sandini, and Giorgio Metta. "iCub". In: *Humanoid Robotics: A Reference*. Springer, 2019, pp. 291–323. DOI: 10.1007/978-94-007-6046-2_21 (cit. on p. 39).

[ND02] Chrystopher L. Nehaniv and Kerstin Dautenhahn. "The Correspondence Problem". In: *Imitation in Animals and Artifacts*. MIT Press, 2002. Chap. 2. DOI: 978-0262527750 (cit. on p. 145).

[NNI09] Tomoaki Nakamura, Takayuki Nagai, and Naoto Iwahashi. "Grounding of Word Meanings in Multimodal Concepts Using LDA". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009, pp. 3943–3948. DOI: 10.1109/IROS.2009.5354736 (cit. on p. 57).

[Nor02] Joel Norman. "Two visual systems and two theories of perception: An attempt to reconcile the constructivist and ecological approaches". In: *Behavioral and Brain Sciences* 25.1 (2002), pp. 73–96. DOI: 10.1017/S0140525X0200002X (cit. on p. 10).

[Nou98] Illah R. Nourbakhsh. "Using Abstraction to Interleave Planning and Execution". In: *Third Biannual World Automation Congress*. 1998, pp. 66–72 (cit. on p. 114).

[OPM02] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), pp. 971–987. DOI: 10.1109/TPAMI.2002.1017623 (cit. on p. 46).

[Osó+10] Pedro Osório, Alexandre Bernardino, Ruben Martinez-Cantin, and José Santos-Victor. "Gaussian Mixture Models for Affordance Learning using Bayesian Networks". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 4432–4437. DOI: 10.1109/IROS.2010.5650297 (cit. on p. 32).

[PA12] Katerina Pastra and Yiannis Aloimonos. "The minimalist grammar of action". In: *Philosophical Transactions of the Royal Society, Series B: Biological Sciences* 367.1585 (2012), pp. 103–117. DOI: 10.1098/rstb.2011.0123 (cit. on p. 34).

[Pac+17] Daniela Pacella, Michela Ponticorvo, Onofrio Gigliotta, and Orazio Miglino. "Basic emotions and adaptation. A computational and evolutionary model". In: *PLoS one* 12.11 (2017), e0187463. DOI: 10.1371/journal.pone.0187463 (cit. on p. 162).

[PAD11] Alice Mado Proverbio, Roberta Adorni, and Guido Edoardo D'Aniello. "250 ms to code for action affordance during observation of manipulable objects". In: *Neuropsychologia* 49.9 (2011), pp. 2711–2717. DOI: 10.1016/j.neuropsychologia.2011.05.019 (cit. on p. 10).

[Pas+16] Giulia Pasquale, Carlo Ciliberto, Lorenzo Rosasco, and Lorenzo Natale. "Object Identification from Few Examples by Improving the Invariance of a Deep Convolutional Neural Network". In: *IEEE/RSJ International Confer-*

ence on Intelligent Robots and Systems. 2016. DOI: 10.1109/IROS.2016.7759720 (cit. on p. 117).

[Pas08] Katerina Pastra. "PRAXICON: the development of a grounding resource". In: *International Workshop on Human–Computer Interaction*. 2008 (cit. on pp. 35, 112, 116).

[Pat+10] Ugo Pattacini, Francesco Nori, Lorenzo Natale, Giorgio Metta, and Giulio Sandini. "An Experimental Evaluation of a Novel Minimum-Jerk Cartesian Controller for Humanoid Robots". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010. DOI: 10.1109/IROS.2010.5650851 (cit. on pp. 42, 119, 165).

[PB06] Rolf Pfeifer and Josh Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT press, 2006. ISBN: 978-0262162395 (cit. on p. 6).

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988. ISBN: 978-1558604797 (cit. on pp. 21, 22, 26, 123, 161).

[Pia62] Jean Piaget. *Play, Dreams & Imitation in Childhood*. W. W. Norton & Company, 1962. ISBN: 978-0393001716 (cit. on p. 79).

[PPD06] Rong Pan, Yun Peng, and Zhongli Ding. "Belief Update in Bayesian Networks Using Uncertain Evidence". In: *IEEE International Conference on Tools with Artificial Intelligence*. 2006, pp. 441–444. DOI: 10.1109/ICTAI.2006.39 (cit. on p. 60).

[Pra+16] Erwin Prassler, Mario E. Munich, Paolo Pirjanian, and Kazuhiro Kosuge. "Domestic Robotics". In: *Springer Handbook of Robotics*. Springer, 2016, pp. 1729–1758. DOI: 10.1007/978-3-319-32552-1_65 (cit. on p. 107).

[PRM10] Frédéric Py, Kanna Rajan, and Conor McGann. "A Systematic Agent Framework for Situated Autonomous Systems". In: *International Conference on Autonomous Agents and Multi-Agent Systems*. 2010, pp. 583–590 (cit. on p. 110).

[Pul05] Friedemann Pulvermüller. "Brain mechanisms linking language and action". In: *Nature Reviews Neuroscience* 6.7 (2005), pp. 576–582. DOI: 10.1038/nrn1706 (cit. on p. 34).

[PZK07] Hanna M. Pasula, Luke S. Zettlemoyer, and Leslie Pack Kaelbling. "Learning Symbolic Models of Stochastic Domains". In: *Journal of Artificial Intelligence Research* 29 (2007), pp. 309–352. DOI: 10.1613/jair.2113 (cit. on p. 114).

[Qui+09] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: An Open-Source Robot Operating System". In: *IEEE International Conference on Robotics and Automation*. Vol. 3. Workshop on Open-Source Software 3.2. 2009, p. 5 (cit. on p. 39).

[RA15] Siddharth S. Rautaray and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: a survey". In: *Artificial Intelligence Review* 43.1 (2015), pp. 1–54. DOI: 10.1007/s10462-012-9356-9 (cit. on p. 143).

[Rab89] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286. DOI: 10.1109/5.18626 (cit. on pp. 146, 147, 154).

[Raf+16] Stéphane Raffard, Catherine Bortolon, Mahdi Khoramshahi, Robin N. Salesse, Marianna Burca, Ludovic Marin, Benoit G. Bardy, Aude Billard, Valérie Macioce, and Delphine Capdevielle. "Humanoid robots versus humans: How is emotional valence of facial expressions recognized by individuals with schizophrenia? An exploratory study". In: *Schizophrenia research* 176.2-3 (2016), pp. 506–513. DOI: 10.1016/j.schres.2016.06.001 (cit. on p. 162).

[RBC15] Karinne Ramirez-Amaro, Michael Beetz, and Gordon Cheng. "Transferring skills to humanoid robots by extracting semantic representations from observations of human activities". In: *Artificial Intelligence* 247 (2015), pp. 95–118. DOI: 10.1016/j.artint.2015.08.009 (cit. on p. 111).

[RC04] Giacomo Rizzolatti and Laila Craighero. "The mirror-neuron system". In: *Annual Review of Neuroscience* 27 (2004), pp. 169–192. DOI: 10.1146/annurev.neuro.27.070203.144230 (cit. on p. 12).

[RFG01] Giacomo Rizzolatti, Leonardo Fogassi, and Vittorio Gallese. "Neurophysiological mechanisms underlying the understanding and imitation of action". In: *Nature Reviews Neuroscience* 2 (2001), pp. 661–670. DOI: 10.1038/35090060 (cit. on p. 13).

[RM04] Narender Ramnani and R. Christopher Miall. "A system in the human brain for predicting the actions of others". In: *Nature Neuroscience* 7.1 (2004), pp. 85–90. DOI: 10.1038/nn1168 (cit. on p. 52).

[RN09] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Third edition. Pearson, 2009. ISBN: 978-0136042594 (cit. on pp. 2, 21).

[Rob77] Robert W. Robinson. "Counting unlabeled acyclic digraphs". In: *Combinatorial Mathematics V*. Springer, 1977, pp. 28–43. DOI: 10.1007/BFb0069178 (cit. on p. 26).

[Ron+16] Alessandro Roncone, Ugo Pattacini, Giorgio Metta, and Lorenzo Natale. "A Cartesian 6-DoF Gaze Controller for Humanoid Robots". In: *Robotics: Science and Systems*. 2016. DOI: 10.15607/RSS.2016.XII.022 (cit. on pp. 42, 165).

[Ros09] David A. Rosenbaum. *Human Motor Control*. Second edition. London, UK: Academic Press, 2009. ISBN: 978-0123742261 (cit. on p. 77).

[Ros77] Deborah Rosenblatt. "Developmental trends in infant play". In: *Biology of Play*. JB Lippincott, 1977. Chap. 4, pp. 33–44. DOI: 10.1177/1525740103024004020 1_4 (cit. on p. 77).

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. "The Earth Mover's Distance as a Metric for Image Retrieval". In: *International Journal of Computer Vision* 40.2 (2000), pp. 99–121. DOI: 10.1023/A:1026543900054 (cit. on p. 97).

[SA12] Emrah Akin Sisbot and Rachid Alami. "A Human-Aware Manipulation Planner". In: *IEEE Transactions on Robotics* 28.5 (Oct. 2012), pp. 1045–1057. DOI: 10.1109/TRO.2012.2196303 (cit. on p. 110).

[Sal+12] Giampiero Salvi, Luis Montesano, Alexandre Bernardino, and Jose Santos-Victor. "Language Bootstrapping: Learning Word Meanings From Perception–Action Association". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42.3 (2012), pp. 660–671. DOI: 10.1109/TSMCB.2011.2172420 (cit. on pp. 5, 33, 39, 49, 53, 54, 56, 59, 62, 69, 140, 146).

[San+17] Adam Santoro, David Raposo, David G.T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. "A simple neural network module for relational reasoning". In: *Conference on Neural Information Processing Systems*. 2017, pp. 4974–4983 (cit. on p. 57).

[Sap+17a] Giovanni Saponaro, Lorenzo Jamone, Alexandre Bernardino, and Giampiero Salvi. "Interactive Robot Learning of Gestures, Language and Affordances". In: *Workshop on Grounding Language Understanding*. Satellite of Interspeech. 2017, pp. 83–87. DOI: 10.21437/GLU.2017-17 (cit. on pp. 14, 50, 60).

[Sap+17b] Giovanni Saponaro, Pedro Vicente, Atabak Dehban, Lorenzo Jamone, Alexandre Bernardino, and José Santos-Victor. "Learning at the Ends: From Hand to Tool Affordances in Humanoid Robots". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2017, pp. 331–337. DOI: 10.1109/DEVLRN.2017.8329826 (cit. on pp. 14, 76, 82, 93, 94).

[Sap+18] Giovanni Saponaro, Alexandre Antunes, Rodrigo Ventura, Lorenzo Jamone, and Alexandre Bernardino. "Combining Affordance Perception and Probabilistic Planning for Robust Problem Solving in a Cognitive Robot". In: *Autonomous Robots* (2018). Under review (cit. on pp. 15, 107).

[Sap+19] Giovanni Saponaro, Lorenzo Jamone, Alexandre Bernardino, and Giampiero Salvi. "Beyond the Self: Using Grounded Affordances to Interpret and Describe Others' Actions". In: *IEEE Transactions on Cognitive and Developmental Systems* (2019). DOI: 10.1109/TCDS.2018.2882140 (cit. on pp. 14, 50).

[SB11] Giovanni Saponaro and Alexandre Bernardino. "Generation of Meaningful Robot Expressions with Active Learning". In: *ACM/IEEE International Conference on Human–Robot Interaction*. Late Breaking Report. 2011, pp. 243–244. DOI: 10.1145/1957656.1957752 (cit. on pp. 15, 161).

[Sch+18] Paul Schydlo, Mirko Raković, Lorenzo Jamone, and José Santos-Victor. "Anticipation in Human-Robot Cooperation: A Recurrent Neural Network Approach for Multiple Action Sequences Prediction". In: *IEEE International Conference on Robotics and Automation*. 2018, pp. 5909–5914. DOI: 10.1109/ICRA.2018.8460924 (cit. on p. 142).

[Sch06] Stefan Schaal. "Dynamic Movement Primitives - A Framework for Motor Control in Humans and Humanoid Robotics". In: *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280. DOI: 10.1007/4-431-31381-8_23 (cit. on p. 141).

[Sci+15] Alessandra Sciutti, Caterina Ansuini, Cristina Becchio, and Giulio Sandini. "Investigating the ability to read others' intentions using humanoid robots". In: *Frontiers in Psychology* 6.September (2015), pp. 1–6. DOI: 10.3389/fpsyg.2015.01362 (cit. on p. 36).

[SD10] Ágnes Szokolszky and Éva Devánszky. "The Development of Spoon-Use in the Daily Routine of Infants: A Naturalistic Observation Study". In: *Studies in Perception and Action IX: Fourteenth International Conference on Perception and Action*. 2010, pp. 74–78 (cit. on pp. 77, 79).

[SH08] Robert St. Amant and Thomas E. Horton. "Revisiting the definition of animal tool use". In: *Animal Behaviour* 75.4 (2008), pp. 1199–1208. DOI: 10.1016/j.anbehav.2007.09.028 (cit. on p. 75).

[Sip12] Michael Sipser. *Introduction to the Theory of Computation*. Third edition. Cengage Learning, 2012. ISBN: 978-1133187813 (cit. on p. 62).

[SK16] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Second edition. Springer, 2016. ISBN: 978-3319325507 (cit. on p. 2).

[SLH12] Guido Schillaci, Bruno Lara, and Verena V. Hafner. "Internal Simulations for Behaviour Selection and Recognition". In: *International Workshop on Human Behavior Understanding*. 2012, pp. 148–160. DOI: 10.1007/978-3-642-34014-7_13 (cit. on p. 145).

[Smi+14] Linda B. Smith, Sandra Street, Susan S. Jones, and Karin H. James. "Using the axis of elongation to align shapes: Developmental changes between 18 and 24 months of age". In: *Journal of Experimental Child Psychology* 123 (2014), pp. 15–35. DOI: 10.1016/j.jecp.2014.01.009 (cit. on p. 79).

[Sri+14] Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. "Combined Task and Motion Planning Through an Extensible Planner-Independent Interface Layer". In: *IEEE International Conference on Robotics and Automation*. 2014, pp. 639–646. DOI: 10.1109/ICRA.2014.6906922 (cit. on p. 114).

[SS07] Jivko Sinapov and Alexander Stoytchev. "Learning and Generalization of Behavior-Grounded Tool Affordances". In: *IEEE International Conference on Developmental and Learning*. 2007, pp. 19–24. DOI: 10.1109/DEVLRN.2007.4354064 (cit. on pp. 79–81).

[SSB13] Giovanni Saponaro, Giampiero Salvi, and Alexandre Bernardino. "Robot Anticipation of Human Intentions through Continuous Gesture Recognition". In: *International Conference on Collaboration Technologies and Systems*. International Workshop on Collaborative Robots and Human–Robot Interaction. 2013, pp. 218–225. DOI: 10.1109/CTS.2013.6567232 (cit. on pp. 15, 143, 146).

[Ste03] Luc Steels. "Evolving grounded communication for robots". In: *Trends in Cognitive Sciences* 7.7 (2003), pp. 308–312. DOI: 10.1016/S1364-6613(03)00129-3 (cit. on p. 52).

[Sto05] Alexander Stoytchev. "Behavior-Grounded Representation of Tool Affordances". In: *IEEE International Conference on Robotics and Automation*. 2005, pp. 3060–3065. DOI: 10.1109/ROBOT.2005.1570580 (cit. on pp. 79–81).

[Sto08] Alexander Stoytchev. "Learning the Affordances of Tools using a Behavior-Grounded Approach". In: *Towards Affordance-Based Robot Control*. Springer, 2008, pp. 140–158. DOI: 10.1007/978-3-540-77915-5_10 (cit. on pp. 79–81).

[Str+16] Francesca Stramandinoli, Vadim Tikhanoff, Ugo Pattacini, and Francesco Nori. "Grounding Speech Utterances in Robotics Affordances: An Embodied Statistical Language Model". In: *IEEE International Conference on Development and Learning and on Epigenetic Robotics*. 2016, pp. 79–86. DOI: 10.1109/DEVLRN.2016.7846794 (cit. on p. 57).

[SW09] Abhik Shah and Peter Woolf. "Python Environment for Bayesian Learning: Inferring the Structure of Bayesian Networks from Knowledge and Data". In: *Journal of Machine Learning Research* 10 (Feb. 2009), pp. 159–162 (cit. on p. 27).

[Tan+16] Tadahiro Taniguchi, Takayuki Nagai, Tomoaki Nakamura, Naoto Iwahashi, Tetsuya Ogata, and Hideki Asoh. "Symbol Emergence in Robotics: A Survey". In: *Advanced Robotics* 30.11-12 (2016), pp. 706–728. DOI: 10.1080/01691864.2016.1164622 (cit. on p. 112).

[TCL07] Michael Tomasello, Malinda Carpenter, and Ulf Liszkowski. "A New Look at Infant Pointing". In: *Child Development* 78.3 (2007), pp. 705–722. DOI: 10.1111/j.1467-8624.2007.01025.x (cit. on p. 51).

[Tel+11a] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. "Approaching the Symbol Grounding Problem with Probabilistic Graphical Models". In: *AI Magazine* 32.4 (2011), pp. 64–77. ISSN: 0738-4602 (cit. on p. 112).

[Tel+11b] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation". In: *AAAI Conference on Artificial Intelligence*. 2011, pp. 1507–1514 (cit. on p. 112).

[Thi+13] Serge Thill, Daniele Caligiore, Anna M. Borghi, Tom Ziemke, and Gianluca Baldassarre. "Theories and computational models of affordance and mirror systems: An integrative review". In: *Neuroscience and Biobehavioral Reviews* 37.3

(2013), pp. 491–521. DOI: 10.1016/j.neubiorev.2013.01.012 (cit. on p. 30).

[Tik+08] Vadim Tikhanoff, Paul Fitzpatrick, Giorgio Metta, Lorenzo Natale, Francesco Nori, and Angelo Cangelosi. "An Open Source Simulator for Cognitive Robotics Research: The Prototype of the iCub Humanoid Robot Simulator". In: *Workshop on Performance Metrics for Intelligent Systems*. 2008. DOI: 10.1145/1774674.1774684 (cit. on p. 93).

[Tik+13] Vadim Tikhanoff, Ugo Pattacini, Lorenzo Natale, and Giorgio Metta. "Exploring affordances and tool use on the iCub". In: *IEEE-RAS International Conference on Humanoid Robots*. 2013. DOI: 10.1109/HUMANOIDS.2013.7029967 (cit. on pp. 79, 81, 119).

[TK00] Simon Tong and Daphne Koller. "Active Learning for Parameter Estimation in Bayesian Networks". In: *Neural Information Processing Systems*. Vol. 13. 2000, pp. 647–653 (cit. on pp. 161, 172, 173).

[Tun+07] Eugene Tunik, Nichola J. Rice, Antonia Hamilton, and Scott T. Grafton. "Beyond grasping: Representation of action in human anterior intraparietal sulcus". In: *NeuroImage* 36 (2007), T77–T86. DOI: 10.1016/j.neuroimage.2007.03.026 (cit. on p. 10).

[Tur50] Alan M. Turing. "Computing machinery and intelligence". In: *Mind* 59.236 (1950), pp. 433–460. DOI: 10.1093/mind/LIX.236.433 (cit. on p. 6).

[Ugu+15a] Emre Ugur, Yukie Nagai, Hande Celikkanat, and Erhan Oztop. "Parental scaffolding as a bootstrapping mechanism for learning grasp affordances and imitation skills". In: *Robotica* 33.5 (2015), pp. 1163–1180. DOI: 10.1017/S0263574714002148 (cit. on p. 145).

[Ugu+15b] Emre Ugur, Yukie Nagai, Erol Sahin, and Erhan Oztop. "Staged Development of Robot Skills: Behavior Formation, Affordance Learning and Imitation with Motionese". In: *IEEE Transactions on Autonomous Mental Development* 7.2 (2015), pp. 119–139. DOI: 10.1109/TAMD.2015.2426192 (cit. on p. 145).

[UP15a] Emre Ugur and Justus Piater. "Bottom-Up Learning of Object Categories, Action Effects and Logical Rules: From Continuous Manipulative Exploration to Symbolic Planning". In: *IEEE International Conference on Robotics and Automation*. 2015, pp. 2627–2633. DOI: 10.1109/ICRA.2015.7139553 (cit. on p. 113).

[UP15b] Emre Ugur and Justus Piater. "Refining discovered symbols with multi-step interaction experience". In: *IEEE-RAS International Conference on Humanoid Robots*. 2015, pp. 1007–1012. DOI: 10.1109/HUMANOIDS.2015.7363477 (cit. on p. 113).

[VHF16] David Vernon, Claes von Hofsten, and Luciano Fadiga. "Desiderata for developmental cognitive architectures". In: *Biologically Inspired Cognitive Architectures* 18 (2016), pp. 116–127. DOI: 10.1016/j.bica.2016.10.004 (cit. on p. 110).

[VJB16] Pedro Vicente, Lorenzo Jamone, and Alexandre Bernardino. "Robotic Hand Pose Estimation Based on Stereo Vision and GPU-Enabled Internal Graphical Simulation". In: *Journal of Intelligent & Robotic Systems* 83.3-4 (2016), pp. 339–358. DOI: 10.1007/s10846-016-0376-6 (cit. on p. 92).

[VS12] Niklas Vanhainen and Giampiero Salvi. "Word Discovery with Beta Process Factor Analysis". In: *International Conference on Speech Communication and Technology*. 2012, pp. 798–802 (cit. on p. 73).

[VS14] Niklas Vanhainen and Giampiero Salvi. "Pattern Discovery in Continuous Speech Using Block Diagonal Infinite HMM". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2014, pp. 3719–3723. DOI: 10.1109/ICASSP.2014.6854296 (cit. on p. 73).

[Wan+13] Zhikun Wang, Katharina Mülling, Marc Peter Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. "Probabilistic Movement Modeling for Intention Inference in Human–Robot Interaction". In: *International Journal of Robotics Research* 32.7 (Apr. 2013), pp. 841–858. DOI: 10.1177/0278364913478447 (cit. on p. 35).

[WH99] Ying Wu and Thomas S Huang. "Vision-Based Gesture Recognition: A Review". In: *Gesture-Based Communication in Human–Computer Interaction*. Springer, 1999, pp. 103–115. DOI: 10.1007/3-540-46616-9_10 (cit. on p. 144).

[WHS05] Alexander B. Wood, Thomas E. Horton, and Robert St. Amant. "Effective Tool Use in a Habile Agent". In: *IEEE Systems and Information Engineering Design Symposium*. 2005, pp. 75–81. DOI: 10.1109/SIEDS.2005.193241 (cit. on pp. 79, 80).

[WMR10] Jason Wolfe, Bhaskara Marthi, and Stuart Russell. "Combined Task and Motion Planning for Mobile Manipulation". In: *International Conference on Automated Planning and Scheduling*. 2010 (cit. on p. 114).

[WRT00] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. "A Gesture Based Interface for Human–Robot Interaction". In: *Autonomous Robots* 9.2 (2000), pp. 151–173. DOI: 10.1023/A:1008918401478 (cit. on p. 143).

[Wu+15] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. "Watch-n-Patch: Unsupervised Understanding of Actions and Relations". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4362–4370. DOI: 10.1109/CVPR.2015.7299065 (cit. on p. 35).

[Wu+16] Chenxia Wu, Jiemi Zhang, Bart Selman, Silvio Savarese, and Ashutosh Saxena. "Watch-Bot: Unsupervised Learning for Reminding Humans of Forgotten Actions". In: *IEEE International Conference on Robotics and Automation*. 2016, pp. 2479–2486. DOI: 10.1109/ICRA.2016.7487401 (cit. on p. 36).

[WW11] Daniel Wigdor and Dennis Wixon. *Brave NUI World: Designing Natural User Interfaces for Touch and Gestures*. Elsevier, 2011. ISBN: 978-0123822314 (cit. on p. 143).

[Xu+09] Jiang Xu, Patrick J. Gannon, Karen Emmorey, Jason F. Smith, and Allen R. Braun. "Symbolic Gestures and Spoken Language Are Processed by a Common Neural System". In: *Proceedings of the National Academy of Sciences of the United States of America* 106.49 (2009), pp. 20664–20669. DOI: 10.1073/pnas.0909197106 (cit. on p. 145).

[You+06] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, 2006 (cit. on pp. 64, 153).

[YPL07] Hee-Deok Yang, A-Yeon Park, and Seong-Whan Lee. "Gesture Spotting and Recognition for Human–Robot Interaction". In: *IEEE Transactions on Robotics* 23.2 (Apr. 2007), pp. 256–270. DOI: 10.1109/TRO.2006.889491 (cit. on p. 143).

[Yu+09] Chen Yu, Linda B. Smith, Hongwei Shen, Alfredo F. Pereira, and Thomas Smith. "Active Information Selection: Visual Attention Through the Hands". In: *IEEE Transactions on Autonomous Mental Development* 1.2 (Aug. 2009), pp. 141–151. DOI: 10.1109/TAMD.2009.2031513 (cit. on p. 79).

[Zec+17] Philipp Zech, Simon Haller, Safoura Rezapour Lakani, Barry Ridge, Emre Ugur, and Justus Piater. "Computational Models of Affordance in Robotics: A Taxonomy and Systematic Classification". In: *Adaptive Behavior* 25.5 (2017),

pp. 235–271. DOI: 10.1177/1059712317726357 (cit. on pp. 30, 43).

[Zec+19] Philipp Zech, Erwan Renaudo, Simon Haller, Xiang Zhang, and Justus Piater. "Action representations in robotics: A taxonomy and systematic classification". In: *International Journal of Robotics Research* 38.5 (2019), pp. 518–562. DOI: 10.1177/0278364919835020 (cit. on p. 141).

[ZL04] Dengsheng Zhang and Guojun Lu. "Review of Shape Representation and Description Techniques". In: *Pattern Recognition* 37.1 (Jan. 2004), pp. 1–19. DOI: 10.1016/j.patcog.2003.07.008 (cit. on p. 46).