

USING LEARNED AFFORDANCES FOR ROBOTIC BEHAVIOR
DEVELOPMENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET REMZİ DOĞAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

“USING LEARNED AFFORDANCES FOR ROBOTIC BEHAVIOR DEVELOPMENT

submitted by **MEHMET REMZİ DOĞAR** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Volkan Atalay
Head of Department, **Computer Engineering**

Asst. Prof. Dr. Erol Şahin
Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Faruk Polat
Computer Engineering, METU

Assoc. Prof. Dr. Ferda Nur Alpaslan
Computer Engineering, METU

Assoc. Prof. Dr. Göktürk Üçoluk
Computer Engineering, METU

Asst. Prof. Dr. Erol Şahin
Computer Engineering, METU

Asst. Prof. Dr. Bilge Say
Informatics Institute, METU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Mehmet Remzi Dođar

Signature :

ABSTRACT

USING LEARNED AFFORDANCES FOR ROBOTIC BEHAVIOR DEVELOPMENT

Doğar, Mehmet Remzi

M.Sc., Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Erol Şahin

September 2007, 66 pages

“Developmental robotics” proposes that, instead of trying to build a robot that shows intelligence once and for all, what one must do is to build robots that can develop [47]. A robot should go through cognitive development just like an animal baby does. These robots should be equipped with behaviors that are simple but enough to bootstrap the system. Then, as the robot interacts with its environment, it should display increasingly complex behaviors. Studies in developmental psychology [63, 28] and neurophysiology [56, 27, 3] provide support for the view that, the animals start with innate simple behaviors, and develop more complex behaviors through the differentiation, sequencing, and combination of these primitive behaviors. In this thesis, we propose such a development scheme for a mobile robot. J.J. Gibson’s concept of “affordances” [33] provides the basis of this development scheme, and we use a formalization of affordances [13] to make the robot learn about the dynamics of its interactions with its environment. We show that an autonomous robot can start with pre-coded primitive behaviors, and as it executes its behaviors randomly in an environment, it can learn the *affordance relations* between the environment and its behaviors. We then present two ways of using these learned structures, in achieving more complex, voluntary behaviors. In the first case, the robot still uses its pre-coded primitive behaviors only, but the *sequencing* of these are such that new more complex behaviors emerge. In the second case, the robot uses its pre-coded primitive behaviors to create *new* behaviors.

Keywords: Robotic Behavior Development, Affordances, Motor Learning, Behavior Learning, Autonomous Robotics

ÖZ

ÖĞRENİLMİŞ SAĞLARLIKLARI KULLANARAK ROBOT DAVRANIŞ GELİŞİMİ

Doğar, Mehmet Remzi

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Erol Şahin

Eylül 2007, 66 sayfa

“Gelişimsel Robotik”, ilk haliyle ve tek seferde zeki davranışlar gösteren robotlar yapmaya çalışmak yerine, gelişebilen robotlar yapmayı savunur [47]. Bir hayvan yavrusunun yapacağı gibi, bir robot da bilişsel bir gelişim geçirmelidir. Bu robotlar basit ama gelişimi başlatmaya da yeterli davranışlarla donatılmış olmalıdırlar. Daha sonra, robot çevresiyle etkileştikçe, giderek daha karmaşık davranışlar sergilemelidir. Gelişim psikolojisi [63, 28] ve nörofizyoloji [56, 27, 3] alanındaki çalışmalar, hayvanların doğuştan gelen basit davranışlar ile başladıkları, ve daha karmaşık davranışları bu ilkel davranışların farklılaşması, sıralanması veya birleştirilmesi yoluyla elde ettikleri görüşünü desteklemektedirler. Bu tezde, bir gezer robota uygulanmak üzere böyle bir gelişim biçimi öneriyoruz. J.J. Gibson’un “sağlarlık” kavramı [33] bu gelişim biçiminin temelini oluşturuyor; ve biz de, robotun çevreyle olan etkileşiminin dinamiklerini öğrenmesinde sağlarlık için yapılmış bir formalizasyonu [13] kullanıyoruz. Bir otonom robotun önceden kodlanmış ilkel davranışlardan başlayarak, bu davranışları belirli bir ortamda rasgele çalıştırdıkça, çevresi ve kendi davranışları arasındaki *sağlarlık ilişkilerini* öğrenebildiğini gösteriyoruz. Sonra, öğrenilmiş olan bu yapıları daha karmaşık ve istemli davranışlar gösterebilmekte kullanmak için iki yöntem sunuyoruz. İlk yöntemde robot, hala sadece önceden kodlanmış olan ilkel davranışlarını kullanıyor, ama bu ilkel davranışların *sıralanması* yeni ve daha karmaşık bir davranışı ortaya çıkaracak şekilde oluyor. İkinci yöntemde ise robot, önceden kodlanmış ilkel davranışlarını *yeni* davranışlar keşfetmek için kullanıyor.

Anahtar Kelimeler: Robot Davranış Gelişimi, Sağlarlık, Motor Öğrenme, Davranış Öğrenme,
Otonom Robotbilim

ACKNOWLEDGMENTS

This work has been possible with the support of many people. I'd like to thank all of them.

Specifically, I'd like to thank my supervisor Erol Şahin, for his guidance during my M.Sc. study and during the writing of this thesis. I'd like to thank him for teaching me so much, and for founding the KOVAN Research Lab. and striving to make it such a convenient place for us, the young scientists, to develop our knowledge and skills.

I'd like to thank all my friends at the KOVAN Research Lab. for their support. I did most of the work presented here in close collaboration with Maya Çakmak, and would like to thank her for her contributions. I'd like to thank Emre Uğur for setting the base (with his M.Sc. thesis) on which we built our theses. I'd like to thank Hande Çelikkanat, Fatih Gökçe, Ali Emre Turgut, Levent Bayındır, Onur Soysal, Erkin Bahçeci, and Yaman Çakmakçı for making working in the lab such a pleasurable experience with their friendship and joy.

During my M.Sc. study I was supported by *The Scientific & Technological Research Council of Turkey* (TÜBİTAK) under the *National Scholarship Programme for MSc Students*. This work was also partially funded by the MACS project, a Future and Emerging Technologies project (IST-FET) of the European Community, under the grant IST-2003-2.3.2.4.

To my family.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	vi
ACKNOWLEDGMENTS	viii
DEDICATION	ix
TABLE OF CONTENTS	x
LIST OF FIGURES	xii
LIST OF TABLES	xiv
CHAPTER	
1 INTRODUCTION	1
2 AFFORDANCES	4
2.1 Ecological approach to perception and “affordances”	4
2.2 Affordance-related research	6
2.2.1 Psychology	6
2.2.2 Neurophysiology and neuropsychology	8
2.2.3 Design	9
2.3 Affordance-related work in robotics	10
3 FORMALIZING AFFORDANCES	13
3.1 Prior formalizations of affordances	13
3.1.1 Turvey’s formalization	13
3.1.2 Stoffregen’s formalization	14
3.1.3 Chemero’s formalization	15
3.1.4 Steedman’s formalization	16
3.2 A new formalism for affordances	17
3.2.1 Three perspectives of affordances	17

3.2.2	A new affordance formalization	19
4	BEHAVIOR DEVELOPMENT	22
4.1	Behavior development in psychology	22
4.2	Motor control and learning in neuroscience	24
4.3	Behavior learning and development in robotics	26
5	EXPERIMENTAL FRAMEWORK	28
5.1	Robotic and simulation platform	28
5.2	Primitive behaviors	29
5.3	Interaction environment	30
5.4	Perception and representation of entities and effects	30
6	USING AFFORDANCES FOR ROBOTIC BEHAVIOR DEVELOPMENT	32
6.1	Methodology	33
6.2	Learning affordance relations	34
6.2.1	Data collection: Interacting with the world	34
6.2.2	Forming affordance relations	35
6.3	Achieving goal-directed behavior	38
6.3.1	Achieving goal-directed behavior through the sequential use of primitive behaviors	38
6.3.2	Achieving goal-directed behavior through the simultaneous use of primitive behaviors	45
7	DISCUSSIONS AND CONCLUSION	54
	REFERENCES	58

LIST OF FIGURES

FIGURES

Figure 3.1	Three perspectives to view affordances.	18
Figure 5.1	The KURT3D, and a snapshot from MACSim.	29
Figure 5.2	Phases of perception.	31
Figure 6.1	Representation of the entity and the effect.	35
Figure 6.2	Interpretation of effect classes obtained with unsupervised clustering for the primitive behavior <i>move-forward</i>	37
Figure 6.3	Relevant grids in the range image representation for three possible primitive behaviors.	38
Figure 6.4	Behavior selection.	39
Figure 6.5	Three different behaviors achieved using the same three primitive be- haviors and their learned <i>affordance relations</i>	41
Figure 6.6	Snapshot from the arena where the performance of the behavior was measured.	42
Figure 6.7	Performance for different number of training interactions.	43
Figure 6.8	Three cases of different goal-directed behaviors.	45
Figure 6.9	Robot's reaction to a situation where the object is placed 20° to the right of the robot.	49
Figure 6.10	Robot's reaction to a situation where the object is placed 45° to the right of the robot.	50
Figure 6.11	Robot's reaction to a situation where the object is placed directly ahead of the robot.	51
Figure 6.12	Robot's reaction to different situations using the two different strategies of using only primitive behaviors, and using behavioral generalization .	52

Figure 6.13 Robot dribbling a ball using the learned affordance relations. 53

LIST OF TABLES

TABLES

Table 5.1	The motor parameter values passed by the primitive behaviors.	29
Table 6.1	The motor parameter values passed by the primitive behaviors for behavioral generalization.	48

CHAPTER 1

INTRODUCTION

“Developmental robotics” is a new approach to “building” intelligent machines [47]. It proposes that, instead of trying to build a robot that shows intelligence once and for all, what one must do is to build robots that can develop. An artificial agent (a robot) should go through cognitive (and sometimes also physical) development just like an animal baby does. These robots should be equipped with sensors, actuators, and behaviors that are simple but enough to bootstrap the system. Then, as the robot interacts with its environment, it should go through developmental stages during which it displays increasingly complex behavior, develops its own symbol system, conceptual world, and learns to use its actuators and sensors more effectively.

Think of a baby that tries to reach out and make grabbing movements in every random direction when it is first born. These reaching-out and grabbing actions are simple innate behaviors that bootstrap the system in order to make the baby interact with its environment, and develop as a result of these multiple interactions [28]. In time, the baby will discover the situations in which the grabbing behavior has some real predictable effect, and it will be able to use these behaviors in a controlled and goal-directed manner, instead of making useless grabbing attempts in random directions.

A developing robot should also start with simple unintentional behaviors. Using these behaviors it should be able to interact with its environment and gather information about this interaction: information about the environment, its own body, and the things that it can consistently change in the environment using its body. Then, with increasing experience, it should be able to execute its behaviors voluntarily to achieve specific goals. It should also be able to demonstrate novel behaviors, or enriched versions of its primitive behaviors, as a result of this development. In the end, this development should have the effect of adapting the robot’s behaviors to its environment, and gradually increasing the robot’s performance.

Objectives and motivation

The objective of this thesis is to propose and implement a development scheme for an autonomous mobile robot. That is, starting from a set of simple pre-coded behaviors/actions, through interaction and experience, we aim to realize a transition from these unintentional behaviors to intentional voluntary behaviors for the robot. This development should also result in demonstration of novel/enriched behaviors that are different from the pre-coded existing behaviors.

On a more conceptual level, our objective is to contribute to the view which suggests that robots, just like human beings and other animals, should go through a developmental process, where they shape their “intelligence” through their own experience.

In MACS project ¹ we aimed to build a robot control architecture that learns, perceives, and uses *affordances*. The *affordance* concept, borrowed from psychology, provides us with a tool to think about the problems of robotics in terms of agent-environment interactions. In proposing the behavioral development of a robot through its interactions with the world, we also place the *affordance* concept at the core of our study. Using affordance representations and our formalization of the concept, the robot learned generic relations about its behaviors and its interaction with the world.

Relation to other work and contributions

This is the third in a series of theses, that used *affordances* to attack different problems in robotics.

In [81], Uğur made a robot learn the “traversability” affordance in an environment. In this study, the features in the environment that specify if a specific behavior of the robot will succeed or not was learned by the robot. Using these learned structures, the robot was then able to traverse in an environment successfully, perceiving the affordances of the objects.

In a second study [7], Çakmak used the learned affordance relations for the problem of *planning*. Based upon the learning structure of Uğur’s, Çakmak used an extended version of the affordance formalization [13] where the robot learned the effects of its own behaviors along with the features in the environment that support these behaviors and effects. Çakmak then mapped these learned structures to the conventional planning structures of *pre-condition*, *action*, and *post-condition*, and investigated how good the robot performs in terms of planning.

¹<http://www.macs-eu.org/>

In this thesis, we will investigate the problem of *behavior development*, using the learned affordance structures. Different from the previous work, we will investigate how new behaviors can emerge through the learning of affordance structures and the usage of pre-coded behaviors.

We still do not know the exact mechanisms of behavioral development in animals/humans (please see Chapter 4 for a discussion). But one way one can achieve “new” behaviors through innate primitive behaviors is through the use of these primitive behaviors in a *sequential* and *simultaneous* manner. In the development scheme we propose, our robot will start with primitive behaviors, and during its interactions with the environment, it will learn generic *affordance relations*, which will then be used in the *sequential* and *simultaneous* execution of the primitive behaviors, so that our robot displays a more “intelligent” new behavior. By *sequential* execution we mean the execution of the primitive behaviors one after another. In the case of *sequential* execution, the robot will be executing only one of its primitive behaviors in an instant, but the sequencing of these behaviors will demonstrate a higher-level behavior. By *simultaneous* execution we mean the execution of multiple of the primitive behaviors at the same instant, and a blending of them. In the end we will see that, our robot is able to discover behaviors that it has never seen or experimented with before, and that are different from the primitive behaviors that it has been trained with. The benefit is that, these “new” behaviors will improve the robot’s performance in situations where the primitive behaviors are not good enough.

In the next chapter we will first introduce our key concept, “affordances”, and give some brief information about the affordance-related studies in robotics. Then we will present various formalizations for the concept of affordances. Next, the studies related to “behavioral development” in psychology, neuroscience, and also in robotics will be presented. We will discuss these studies and see what they can offer to our study. Then we will present the experimental framework for our study. The next chapter will present our development scheme, how we implemented this on our robot, and our results. Lastly, we will conclude with a discussion of the results.

CHAPTER 2

AFFORDANCES

Animals perceive the world around them. They do this to act accordingly in that world. In this perceptual process they use the information coming through their sensors, and this information guides the animal's actions. Perceiving the world and acting accordingly is key to an animal's survival.

But what is in this perceptual information? Animals use perception for selecting and guiding their actions; but does the perceptual information *directly* specify which actions to execute to the animal? Or is it the case that the perceptual information includes only a *valueless* description of the world outside which is then subject to the animal's interpretation, and only as a result of this interpretation an action can be selected?

2.1 Ecological approach to perception and “affordances”

J.J. Gibson, an influential psychologist of the 20th century and founder of the school of *ecological psychology*, believed that the perceptual information includes the action information [33]. According to this view, we perceive the environment in terms of *affordances*. *Affordances* are action possibilities in the environment that are directly specified to the agent acting in it through some variables in the perceptual data. Since these perceptual variables are action possibilities they are directly relevant for the agent's behaviors in that environment. Some simple examples can be given: a rigid, horizontal surface *affords* walking on it; a small stone *affords* throwing; an opening wider than one's shoulder width *affords* passing-through it; etc.

Beginning from his early studies Gibson identified certain “meanings” inherent in the environment that helped the perceiver to act accordingly [40]. These meanings were specified through certain optical variables. For instance, in his studies on the aircraft pilots at the

American army during the Second World War, he exemplified such an optical variable: *the optical center of expansion*. Gibson argued that the point in the visual field of a pilot, from which the picture seems to flow out, was an indicator of the direction of the plane. Gibson claimed that this point and its position relative to the horizon was an important optical variable that is *meaningful* for the behavior of the pilot. The point was that, there are certain optical variables that are directly meaningful for the behavior in a way that requires no more mental inference or calculation. It is sufficient just to pick the optical variable. In our case, it is sufficient for the pilot to perceive the difference between the optical center of expansion and the horizon, for a safe flight. This line of thinking later evolved into what we today called “affordances”.

Although Gibson used the word “affordance” as early as 1966 [32], he proposed a definition for the term only in his last book in 1979:

The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill. The verb to afford is found in the dictionary, but the noun affordance is not. I have made it up. I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment. (J.J. Gibson, 1979/1986, p. 127)

According to the theory of affordances, for an animal there is no need to make further processing of its perception in selecting the proper actions in its perceived environment. To perceive means to know which actions are possible. What is in perception is not valueless/meaningless descriptive data. Rather, it *directly* specifies which actions to execute, what the environment *affords*, without a need for further interpretation. This is known as *direct perception*. Quoting from J.J. Gibson himself:

“The perceiving of an affordance is not a process of perceiving a value-free physical object to which meaning is somehow added in a way that no one has been able to agree upon; it is a process of perceiving a value-rich ecological object.” (J.J. Gibson, 1979/1986, p. 140)

This approach to perception implies a very tight perception-action coupling. Classical approach to perception separates the selection of action processes from those of the perceptual processes. In this classical view, first perceptual data are retrieved, using these data a mental model of the actual world is built, and then according to this mental model proper

actions are taken. On the contrary, J.J. Gibson believed that perceptual and motor processes are highly intertwined. In accordance, perceptual data and action data are also coupled.

J.J. Gibson’s approach also implies a tight ecological coupling between the animal and its environment, hence the name *ecological psychology*. If perceptual information was simply a description of the world outside, it would point only one way: the environment. But in ecological approach, the perceptual information includes something about the animal too; the action. In this sense what is in perception points both ways: to the environment and to the animal. As a result, one should examine the animal-environment ecological system as a whole, and should not isolate either of them. In this sense, J.J. Gibson was very critical of laboratory experiments in psychology of perception, since the subjects (animals/humans) in these experiments were highly isolated from their natural environments.

2.2 Affordance-related research

After J.J. Gibson introduced the term, *affordances* has been the subject of many scientific studies. While most of these were studies of experimental psychology, other scientific disciplines such as neuroscience, design, and robotics have also shown interest in the subject.

2.2.1 Psychology

In psychology, the affordance related research mainly continued under the school of *ecological psychology*. These were studies that investigated the mechanisms of affordance perception; they tried to understand how affordances appear and disappear in the environment, what are the relations of affordances to the physical world behind, and how these relate to the animal’s own body.

A pioneer work in this aspect is [82], where Warren studied the perception of the affordance of stair-climbing. In this work, the mechanisms underlying the perception of this affordance, and the natural basis of the affordance categories “climbable-unclimbable” were investigated.

In the study, using a biomechanical model, Warren first found the critical stair height (in terms of its ratio to leg-length) needed to be able to climb a stair. After this, in the experiments he conducted with human subjects, he showed that the transition between the perceptual categories of “climbable” and “unclimbable” was happening at the same ratio (of stair-height/leg-length) predicted by the biomechanical model. He named the points where the transitions of these perceptual affordance categories took place *critical points*. *Critical*

points have π values that are essentially ratios between an environmental measure and a bodily measure. In this study it was the ratio between the stair-height and the leg-length of the subjects.

Warren did not just find the *critical points*, but he also found the *optimal points* for the action using his biomechanical model. *Optimal points* correspond to the values of the ratio between the stair-height and leg-length, such that the action (here, climbing) can be executed with minimum energy consumption. Again in the experiments conducted using human subjects, he showed that the *optimal points* computed using the biomechanical model correspond to the “preferred stair-height” judgments of the human subjects.

In summary, in this study [82] Warren investigated the natural basis of the perceptual affordance categories, and expressed these in terms of *critical points* and *optimal points*. He claimed that these critical and optimal points (ratios) are constants within a species, and perception of affordances becomes possible through the use of these values. This method of using *critical point* and *optimal point* values in expressing affordances has also been used in other subsequent studies, e.g. [83, 51], and opened the way to express affordances using numerical values.

Mechanisms of affordance perception have been investigated in many similar studies for different actions. In [51], Mark points that the perception of surfaces that afford “sitting on” is related to the ratio of the surface height to the subject’s leg-length. Similarly, in [83] Warren and Whang show that for walking through apertures, the aperture-to-shoulder-width ratio gives the critical and optimal points for the affordance.

In summary, these experiments aimed to show that animals (mostly human) can perceive whether a specific action is *do-able* or *not-do-able* in an environment. This implies that, what we perceive is not necessarily objects (e.g. stairs, doors, chairs), but the action possibilities (e.g. climbable, passable, sitable) offered by the environment. These experiments are characterized by two main points: taking the ratio of an environmental measure and a bodily measure of the human subject; and, based on the value of this ratio, making a binary judgment about whether a specific action is do-able or not.

The first point indicates how the experimenters interpreted affordances. Since affordances were roughly defined as the properties of the environment taken relative to the organism acting in it, the goal was to show that the ratio between an environmental measure and a bodily measure of the organism have consequences for behavior. This ratio must also be perceivable, so that the organism is aware of this measure which, in a way, determines the success of its behavior. Thus, this relativity of environmental properties was incorporated into

the experiments simply as a division operation between two metrics, one of the environment and one of the organism. From a conceptual point of view, this is a crude simplification of the relation between the properties of the organism and the environment that comprise an affordance, but for the particular actions and setups used in the experiments, it seemed sufficient.

2.2.2 Neurophysiology and neuropsychology

A set of findings of neurophysiological and neuropsychological research that is also associated with the idea of affordances came from studies on *mirror* and *canonical neurons* which were discovered in the pre-motor cortex of the monkey brain. During experiments with monkeys [64] (later similar findings were also found for human subjects [21]), mirror neurons fired both when the monkey was grasping an object, and when the monkey was watching somebody else do the grasping. These findings implied that the same neurons were used both ways: for the execution of an action as output of the system, and also for perceiving that action as an input to the system [25]. Their discovery supports the view that says action and perception are closely related. These neurons, which are located in the premotor cortex of the monkey brain, are thought to be responsible for the motor activation of actions like grasping and holding.

Rizzolatti and Gentilucci [65] discovered that canonical neurons, normally considered to be motor neurons for grasping actions, would fire when the subject does not execute a grasping action, but only sees a graspable object. Their activity on such a purely perceptive task that included an object that affords that particular action the motor neurons were responsible for, indicated that they may be related to the concept of affordance. The resulting conclusions are interestingly similar to those of the ecological approach:

“This process, in neurophysiological terms, implies that the same neuron must be able not only to code motor acts, but also to respond to the visual features triggering them. ...3D objects, are identified and differentiated not in relation to their mere physical appearance, but in relation to the effect of the interaction with an acting agent.” ([24])

Similar results were found when subjects were doing a highly cognitive task such as ‘imagining’ that they are doing a grasping action [34]. Again the motor neurons were activated, showing that the neural circuitry that is used for motor activations are also utilized during a cognitive task such as ‘imagining’.

In [37], Humphreys showed that, when presented with a tool, some patients, who lacked the ability to name the tool, had no problem in gesturing the appropriate movement for using it. According to Humphreys, this suggested a direct link from the visual input to the motor actions that is independent from more abstract representations of the object, e.g. its name. In another study that Humphreys presented, two groups were shown object pictures, non-object pictures and words. One of the groups was asked to determine if some actions were applicable to what had been presented. The other control group was asked to make size judgments. The brain activities in both groups were compared using functional brain imaging. It was observed that a specific region of the brain was activated more in the first group who were to make action judgments. It was also seen that this specific region was activated more when the subjects were presented with pictures of the objects rather with than the name. This showed that action related regions of the brain were activated more when the visual input was supplied, rather than just naming it. All these findings suggest that there is a strong link between perception and action in terms of neuropsychological activity.

2.2.3 Design

The concept affordance has influenced other, seemingly unrelated, disciplines as well. One of these is the Design domain. The concept was introduced to the Design community by D. Norman's popular book, *Psychology of Everyday Things* (POET) [57]. In his book, D. Norman discussed the perceptual information that can make the user aware of an object's affordances. In this context, he defined affordances as follows:

“... affordance refers to the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used.”

Unlike J.J. Gibson however, D. Norman was interested in how “everyday things” can be designed such that the user can easily infer what they afford. He analyzed the design of existing everyday tools and interfaces, identifying design principles. In this respect, his discussion of affordances deviated from the Gibsonian definition of the term [52]. In [58], D. Norman writes:

“The designer cares more about what actions the user perceives to be possible than what is true”.

Since POET, the term affordance has been used in many ways in the Design community, some in the sense that D. Norman introduced, some being more loyal to J.J. Gibson’s definition, and others deviating from both of these and using the term in a totally new way [52].

In a later article [58], D. Norman, uncomfortable with the misuse of the term in the Design community, distinguished between “real affordances”, indicating the potentials in the environment independent from the user’s perception, and what he calls “perceived affordances” stating:

“When I get around to revising POET, I will make a global change, replacing all instances of the word ‘affordance’ with the phrase ‘perceived affordance’.”

2.3 Affordance-related work in robotics

The concept of affordances is highly related to autonomous robot control and it has influenced studies in this field. We believe that, for a proper discussion of the relationship of the affordance concept to robot control, the similarity of the arguments of J.J. Gibson’s theory and reactive/behavior-based robotics should first be noted [1].

The concept of affordances and behavior-based robotics emerged in very similar ways as opposing suggestions to the then dominant paradigms in their fields. J.J. Gibson constructed his theory based on criticism of the then dominant theory of perception and cognition, which favored modeling and inference. Likewise, behavior-based robotics was motivated by criticism of the then dominant robotic architectures, which favored modeling and inference. This parallelism between the two fields suggests that they are applications of the same line of thinking to different domains (p. 244, [1]; [18]).

Opposing modeling and inference, J.J. Gibson defended a more direct relationship between the organism and the environment and suggested that a model of the environment and costly inferential processes were not needed. In a similar vein, behavior-based robotics advocated a tight coupling between perception and action. Brooks, claiming that “the world is its own best model”, suggested an approach that eliminated all modeling and internal representation [5].

J.J. Gibson suggested that only the relevant information is picked up from the environment, saying “perception is economical”. In robotics a behavior is a sensory-motor mapping which can often be simplified to a function from certain sensors to certain actuators. In this sense, the perceptual part of a behavior can be said to implement *direct perception* by

extracting only the relevant information from the environment for action, without relying on modeling or inference. Such a minimality is also in agreement with the economical perception concept of the affordance theory.

As discussed above, most of the concepts within affordance theory are inherently included in reactive robotics. The behaviors should be minimally designed for the task, taking into account the niche of the robot's working environment and the task itself. This is in agreement with the arguments of ecological psychology. Some roboticists have already been explicitly using ideas on affordances in designing behavior-based robots. For example, Murphy [54] suggested that robotic design can benefit from ideas in the theory of affordances such that complex perceptual modeling can be eliminated without loss in capabilities. She studied three case studies and drew attention to the importance of the ecological niche in the design of behaviors. Likewise, Duchon et al. [18] benefited from J.J. Gibson's ideas on direct perception and optic flow in the design of behaviors and coined the term *ecological robotics* for the practice of applying ecological principles to the design of mobile robots.

The use of affordances within autonomous robotics is mostly confined to behavior-based control of the robots, and its use in deliberation remains a rather unexplored area. This is not a coincidence, but a consequence of the shortfalls in J.J. Gibson's theory. The reactive approach could not scale up to complex tasks in robotics, in the same way that the theory of affordances in its original form was unable to explain some aspects of perception and cognition.

Recently a number of robotic studies focused on the learning of affordances in robots. In these studies, affordance learning is referred to as the learning of the consequences of a certain action in a given situation [22, 74, 73], or learning of the invariant properties of environments that afford a certain behavior [49].

MacDorman [49], extracted invariant features of different affordance categories. In his study, the invariant features are defined as image signatures that do not vary among the same affordance category but vary among different affordance categories. However, his affordance categories were defined in terms of internal indicators, such as tasty, poisonous, and were not directly related to the actions.

In a recent study [61], an artificial agent is used to represent the state of the world internally as behavioral *affordances* and *goals*. For each action in its repertoire, the agent has *outcome predictors* that correspond to preconditions for the action, and *outcome indicators* that correspond to post-conditions for the action. These predictors and indicators are used to represent the internal state of the agent.

Stoytchev [74, 73] studied learning for the so-called ‘binding affordances’ and ‘tool affordances’, where learning binding affordances corresponds to discovering the behavior sequences that result in the robot arm binding to different kinds of objects, whereas learning tool affordances corresponds to discovering tool-behavior pairs that give the desired effects. In this study the representation of objects is said to be grounded in the behavioral repertoire of the robot, in the sense that the robot knows what it can do with an object using each behavior. However, in this study, object identification was done by assigning unique colors to each object, hence leaving no way of building associations between the distinctive features of the objects and their affordances. Therefore, a generalization which would make the robot respond properly to novel objects was not possible.

In [22], Fitzpatrick et al. studied the learning of object affordances in a robotic domain. They proposed that a robot can learn what it can do with an object only by acting on it, ‘playing’ with it, and observing the effects in the environment. For this aim, they used four different actions of a robot arm, namely, *pull in*, *side tap*, *push away*, *back tap*, on four different objects. (bottle, cube, toy car, ball). After applying each of the actions on each of the objects several times, the robot learned about the roll-ability¹ affordance of these objects, by observing the changes in the environment during the application of the actions. Then, when it needs to roll an object, it uses this knowledge. However, similar to Stoytchev’s study, Fitzpatrick et al. did not establish any association between the visual features of the objects and their affordances, giving no room for generalization of the affordance knowledge to novel objects.

In both Stoytchev’s works [74, 73] and Fitzpatrick et al.’s work [22], the objects were differentiated using their colors only. Therefore, when the learning had been done with a red toy car, if the robot was presented with the very same car but this time a blue one, it would have no idea about the rolling affordance of it. This again shows the importance of learning the distinctive features of objects, along with the effects that can be created by acting on these objects, and building the association between these two learned sets, so that the robot can make predictions about the effects it can create in the environment when it is presented with new objects that it had not seen before.

¹What the robot actually learns about objects is the most probable rolling direction of the objects with respect to their principal axis. Hence, after the learning phase, the robot knows that the bottle rolls perpendicular to its principal axis, and the toy car rolls parallel to its principal axis.

CHAPTER 3

FORMALIZING AFFORDANCES

The discussions around the concept of *affordances*, discussions on what *affordances* are, and what kind of qualities they possess continued after J.J. Gibson died. One of the major reasons was that Gibson himself defined term differently in different places, revised the definition of the term constantly, and his understanding of the term also evolved over time [13]. Consequently, the need aroused to formalize the concept, and to collect and organize the properties associated with affordances in a coherent and formal structure. In this chapter we will present such attempts. In the next section we will present the previous formalization attempts. Then we will present our proposal for formalizing affordances, first discussed in [13].

3.1 Prior formalizations of affordances

After J.J. Gibson, there has been a number of studies [77, 67, 35, 85, 69, 72, 10, 53] attempting to clarify the meaning behind the term affordances and to create a common understanding on which discussions can be based. We will now review four of the proposed formalisms.

3.1.1 Turvey's formalization

One of the earliest attempts to formalize affordances came from Turvey [77]. In his formalism, Turvey defined an affordance as a *disposition*. Here, a disposition is a property of a thing that is a potential, a possibility. These potentials become *actualized* if they combine with their complements (e.g. “solubility” of the salt is its disposition, and if it combines with its complement, which is water’s property of “being able to solve”, then they get actualized, resulting in the salt getting “dissolved”). Therefore, dispositions are defined in pairs, and when two complement dispositions meet in space and time, they get actualized. Basing

his views on this account of dispositions, Turvey defined affordances as dispositions of the environment, and defined their complement dispositions as the “effectivities” of the organism. He provided this definition:

“An affordance is a particular kind of disposition, one whose complement is a dispositional property of an organism.”(M.T. Turvey, 1992, p. 179)

Later in his discussion, Turvey formalized this definition as follows:

“Let W_{pq} (e.g., a person-climbing-stairs system) = $j(X_p, Z_q)$ be composed of different things Z (person) and X (stairs). Let p be a property of X and q be a property of Z . Then p is said to be an affordance of X and q the effectivity of Z (i.e. the complement of p), if and only if there is a third property r such that:

- $W_{pq} = j(X_p, Z_q)$ possesses r . [where $j(\cdot)$ is the juxtaposition function that joins X_p and Z_q .]
- $W_{pq} = j(X_p, Z_q)$ possesses neither p nor q .
- Neither Z nor X possesses r .”(M.T. Turvey, 1992, p. 180)

Here, when the physical structure that renders the stairs climb-able (X_p), and the effectivity of the agent (W_q) that makes it able to climb come together ($j(\cdot)$), a new dynamics -the action of climbing- (r) arise.

In this formalism, although the actualization of affordances requires an interaction of an agent on the environment to produce a new dynamics, Turvey explicitly attached affordances to the environment that the organism is acting in.

3.1.2 Stoffregen’s formalization

A criticism of Turvey’s formalism came from Stoffregen [72]. According to Stoffregen, there are two main views about affordances. The first view places affordances in the environment alone, while the second view places affordances in the organism-environment system as a whole. Stoffregen adopts the latter view and argues that affordances *can not* be defined as properties of the environment only, as Turvey did. In this line of view, Stoffregen [72] described affordances as:

“Affordances are properties of the animal-environment system, that is, that they are emergent properties that do not inhere in either the environment or the animal.”(T.A. Stoffregen, 2003, p. 115)

He claimed that attaching affordances to the environment was problematic for their specification to the organism. The reason was that if affordances belong to the environment only, and if what the organism perceives are affordances, then the organism perceives things that are only about the environment but not about itself. If this is the case, then the agent has to do further perceptual processing to infer what is available *for him*. However, this goes against the basic notion of *direct perception*.

Based on these criticisms, Stoffregen modified Turvey’s definition to propose a new one to resolve these problems. He presented it in the following way:

“Let W_{pq} (e.g., a person-climbing-stairs system) = (X_p, Z_q) be composed of different things Z (e.g., person) and X (e.g., stairs). Let p be a property of X and q be a property of Z . The relation between p and q , p/q , defines a higher order property (i.e., a property of the animal–environment system), h . Then h is said to be an affordance of W_{pq} if and only if

- $W_{pq} = (X_p, Z_q)$ possesses h .
- Neither Z nor X possesses h .”(T.A. Stoffregen, 2003, p. 123)

Here, affordances are defined as “properties of the animal-environment system”, rather than as properties of the environment only.

3.1.3 Chemero’s formalization

Chemero [10] also criticized Turvey’s view which placed affordances in the environment regarding them as environmental properties. Partially in agreement with Stoffregen’s proposal, Chemero suggested that:

“Affordances, are relations between the abilities of organisms and features of the environment.”(A. Chemero, 2003, p. 189)

This definition refines Stoffregen’s proposal in a number of ways. First, it states that affordances are “relations within the animal-environment system”, rather than “properties of the animal-environment system”. Second, it also notes that this relation exists between the “abilities of the organism” and the “features of the environment”, as compared with a property (of the system) being generated through the interaction between the “property of the organism” and the “property of the environment”.

Formally Chemero proposed that an affordance is a relation that can be represented in the form of:

Affords- ϕ (feature, ability), where ϕ is the afforded behavior.

Here the term “ability” stands for the functional properties of the organisms that are shaped through the evolutionary history of the species or the developmental history of the individual. In that respect, they are different from simple body-scale measures (e.g. the leg-length), but correspond to more general capabilities of the organism. One of the main differences between the two similar formalisms of Stoffregen and Chemero, which both define affordances at the organism-environment scale, is that while Stoffregen’s definition of affordance does not include the behavior exploiting the affordance, Chemero’s definition does include it.

3.1.4 Steedman’s formalization

Independent of discussions in the ecological psychology literature, there have also been other attempts of formalization of affordances. One of these came from Steedman [69] who used Linear Dynamic Event Calculus to reach a formalization of affordances. Steedman’s formalization skips the perceptual aspect of affordances (e.g. the invariants of the environment that help the agent perceive the affordances, and the nature of these invariants and the relation of them to the bodily properties of the agent etc.), but instead, it focuses on developing a representation where object schemas are defined in relation to the events and actions that they are involved in. For instance, Steedman suggests that a door is linked with the actions of ‘pushing’ and ‘going-through’, and the pre-conditions and consequences of applying these actions to the door. The different actions that are associated with a particular kind of object constitute the *Affordance-set* of that object schema, and this set can be populated via learning. More formally, in Steedman’s formalization, an object schema is a function mapping objects of that kind into second-order functions from their affordances to their results¹. Thus, an object instance specifies what actions can be applied to it, under which conditions and what consequences it yields. This makes the formalization also suitable for planning, for which Steedman argues that reactive/forward-chaining planning is the best candidate. Steedman’s formalization is, as far as we know, the first attempt to develop a formalization of affordances that allows logical/computational manipulation and planning. Steedman also believes this structure of affordances to have implications for the linguistic capability of humans.

¹Steedman’s actual formalization requires at least a basic presentation of Linear Dynamic Event Calculus and Lambda Calculus. Since we do not have the space for these here, we restrict ourselves to the prose definition. For a complete account of this formalization, see [69].

To summarize, it can be said that Stoffregen’s and Chemero’s formalizations, by defining affordances as a relation on the scale of organism-environment system, differ from Turvey’s formalization which defines affordances as environmental properties. But there are also differences between Chemero’s and Stoffregen’s definitions, one of them being the inclusion of behaviors in the definition of affordances in Chemero’s formalization. Steedman’s formalization differs from the other three formalizations by providing an explicit link to action possibilities offered by the environment, and by proposing the use of the concept in planning.

We believe that none of the reviewed formalisms can be used as a base to develop an affordance-based robot control architecture. In the next section, we will introduce three perspectives through which affordances can be discussed, to explain the source of confusion on the discussions.

3.2 A new formalism for affordances

Although there are many prior formalizations of the *affordance* concept, in [13] we proposed a new formalization that will form the base of our study. We do not use the prior formalizations because they lack certain aspects that are critical in building a robotic system that uses affordances. Most of the prior formalizations are interpretations of the concept from a psychological-philosophical perspective, and do not propose -or discuss- the concept from the perspective of building agents that use affordances. This requires thinking about the concept again, keeping in mind that what we need is an interpretation of the concept which can provide us a representation inside the agent that will help the agent perceive the affordances in the environment. In other words, we need a formalization that can make it possible to interpret the concept from the perspective of the agent, since what we are trying to do is building an agent that makes use of affordances.

3.2.1 Three perspectives of affordances

There is not an agreement on the definition of the term *affordance*, and different sources define it differently. These definitions are sometimes conflicting, creating a confusion.

One important disagreement on the definition of affordances is on where to attach them. Some sources define affordances as properties of the environment, that an agent perceives and uses. Other sources define affordances as emergent properties of the animal-environment system, and claim that it can not be defined as an environmental property only. In the literature one can find different confusing statements where affordances are mentioned relative

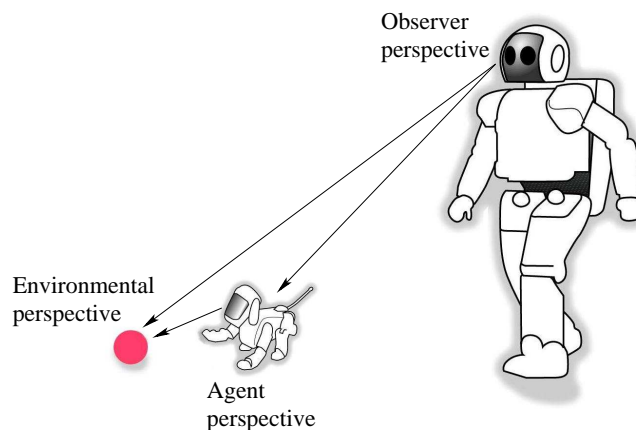


Figure 3.1: Three perspectives to view affordances. In this hypothetical scene (adapted from Erich Rome’s slide depicting a similar scene), the (robot) dog is interacting with a ball, and this interaction is being observed by a human(oid) who is invisible to the dog. (Image taken from [13])

to an environmental object (“an apple affords eating”), to an agent (“there is walkability affordance for the person”), or to the agent-environment system (“the stair affords climbing for the person”)

These different statements are confusing, yet not wrong. That is because we believe that there are three different perspectives to view affordances: the *agent’s perspective*, the *environmental perspective*, and the *observer’s perspective*. In these statements making explicit the perspective that the writer is using will be enough to solve the confusion.

The three perspectives to view affordances can be explained using the scene in Figure 3.1 [13]. In this scene, the robot dog is interacting with a ball, and this interaction is being observed by a humanoid who is invisible to the dog. Here, the dog has the *agent’s perspective* in perceiving the affordances of the ball, the ball has the *environmental perspective* in interacting with the agent, and the human has the *observer’s perspective* in observing the interaction possibilities between the dog and the ball.

From the *agent’s perspective* affordances are the behaviors (and their relevant consequences) offered to that specific agent by its immediate environment, that is, viewing affordances from the eyes of the animal living in that environment and actually using those affordances. In Figure 3.1 for example, the robot dog, using the agent’s perspective would see that it can “push” the ball. In our daily lives also we make use of this view of the affordances. This is also the view that is most relevant for the study of building robots that use

affordances.

From the *environmental perspective*, affordances are all the behaviors (and their relevant consequences) offered by that environment (or generally by a specific object in that environment) to the agents living in that environment. In Figure 3.1, the ball affords “pushing” and “hiding” to the dog, “throwing” to the human, etc. This is the most useful perspective to define an environment from a general perspective, and relative to the living organisms in that environment. In that sense one can claim that the *environmental perspective* of affordances has strong connections with the term *habitat*.

From the *observer’s perspective* affordances are the behaviors (and their relevant consequences) offered by the environment to the agent, in an agent-environment system. This is for example the perspective that a scientist uses when investigating an ecological system, with the environment and the organisms living in the environment taken as a whole. Also in Figure 3.1, the human observer sees that there is “push-ability” affordance in the dog-ball system. One can claim that, this view can also be used when one individual in a society is observing a different individual and its interaction with the environment. This opens the way of using *observer’s perspective* to view affordances, in explaining learning by imitation².

In the following discussion, we will continue discussing only the *agent’s perspective* of affordances, since we believe that it gives us the view of affordances such that it can actually be used by a robot. You can find a more detailed discussion and definition of the other *perspectives* in [13].

3.2.2 A new affordance formalization

In this new formalization [13], Sahin et al. define affordances as a relation between the *agent* and the *environment*, in line with Chemero’s formalization. Affordances are results of a relation between certain properties of an agent and certain properties of its environment. In the formalism the properties of the agent are included as the *behavior*, and the properties of the environment are included as the *entity*. A third component, the *effect*, representing the result of the interaction between the behavior of the robot with the entity in the environment, is also included in the relation. The explicit inclusion of *effect* component is another point where the formalism differs from the previous ones.

The formalism is based on *relation instances* of the form (effect, (behavior, entity)). This can be read as, when the agent applies a certain *behavior* on a certain *entity* in the environment, a certain *effect* is observed. Here an *entity* should not be confused with the

²In this thesis, we will not go any further to investigate the use of affordances in learning by imitation.

term *object*. An *entity* is a collection of features in the environment, that is not necessarily connected to an object. Of course, the set of features can correspond to an object, but it can also correspond to a part of an object, a relation between multiple objects, or features that can not be connected to any object. The *entity* represents the state of the environment (and the agent’s own state) as perceived by the agent. The *behavior* component represents the actualization of the action of the agent, including the bodily features of the agent and the physical dynamics of the interaction. The *effect* component represents the actual change created in the environment, as a result of the application of the *behavior* on the *entity* as perceived by the agent. Quoting from [13], “a certain *behavior* applied on a certain *entity* should produce a certain *effect*, e.g. a certain perceivable change in the environment, or in the state of the agent. For instance, the *lift-ability* affordance implicitly assumes that, when the *lift behavior* is applied to a *stone*, it produces the effect *lifted*, meaning that the *stone’s* position, as perceived by the agent, is elevated.”

Formally, in [13], a first definition for affordances is given as:

An affordance is an acquired relation between a certain *effect* and a (*entity*, *behavior*) tuple, such that when the agent applies the *behavior* on the *entity*, the *effect* is generated.

Such relation instances of the form (effect, (behavior, entity)) are acquired through the interaction of the agent with its environment. But they represent a single interaction of the agent with its environment. So they are not ready to be used in making generic predictions. But affordances must have predictive, anticipatory qualities.

Building on top of such *relation instances* we can form generic *affordance relations* that represent facts about the agent’s interaction with the world, rather than being a record of single interaction. Supposing that an agent interacts with its environment and collects many relation instances of the form (effect, (entity, behavior)), it can then merge these relation instances to extract generic affordance relations, generic rules about the nature of the interaction of the agent with its environment.

This can be done through the discovery of invariant properties in the collected *entities*, *behaviors*, and *effects*, and merging them in *equivalence classes*. Below, we will present the different kinds of equivalence classes, how they are formed, and what they represent.

Every behavior has some kind of an effect. While the effect created in the environment during a certain application of a behavior is never exactly the same with the effects of other applications of the behavior, some of them can be said to be equivalent. Lifting a “blue-

can” and a “black-can” creates different effects in the environment as the agent perceives it because of the different colors of the objects, but the effect “lifted”, is internal to both of the effects, and hence they are equivalent with regards to the “lifting” action. Discovering and merging the equivalent effects to build *effect equivalence classes* is a first step in building other equivalence classes and affordance relations.

Two entities are equivalent, if, when a behavior is applied on them the interaction results in *equivalent effects*. Keeping with the example above, if the “blue-can” and the “black-can” entities produces *equivalent effects* when the “lift” behavior is applied, then they are *equivalent entities* in terms of the “lifting” behavior. We can collect and merge such equivalent entities in *entity equivalence classes*. Merging different entities into *entity equivalence classes* is a process of narrowing-down to the set of features that are invariant in all the entities in that equivalence class. For example, for the two equivalent entities of a “blue-can” and a “black-can”, the common features can be represented as “*-can”; which basically says to the robot: “If it is a can, it affords lifting up, no matter what color it is.”

Two behaviors are equivalent, if, when they are applied to *equivalent entities*, they result in *equivalent effects*. This can be two different behaviors that produce the same result in that context; or this equivalence can be understood in terms of specifying a possible range for parameter values of the same behavior. As an example for the former, the “lift-with-left-hand” behavior and the “lift-with-right-hand” can be seen as equivalent behaviors in lifting a “*-can”. As an example for the latter, the parameters that specify the pressure that the hand of the robot will apply to the “can” may be different in different *relation instances*, but the exact value may not matter if it is in a certain range. Such parametrized instances of the same behavior can also be merged in the same *behavior equivalence class*.

The last equivalence class we can form is the *affordance equivalence class*. We can merge the (entity, behavior) couples that produces equivalent effects into an *affordance equivalence class*. An example for this case can be (“ground”, “walk”) and (“river”, “swim”) couples which both have the effect “traversed”.

Based on this account of *equivalence classes*, Sahin et al. [13] revise their definition as:

Affordance (agent perspective): An affordance is an acquired relation between a certain $\langle effect \rangle$ and a certain $\langle (entity, behavior) \rangle$ tuple such that when the agent applies a $(entity, behavior)$ within $\langle (entity, behavior) \rangle$, an *effect* within $\langle effect \rangle$ is generated.

CHAPTER 4

BEHAVIOR DEVELOPMENT

How behavior develops in humans and other animals have been the subject of many scientific studies. In psychology, theories of development have been proposed in this area. At the level of the sensors, central nervous system, and motor neurons, neuroscience has investigated behavior control, motor skills, and motor development. In robotics also, there have been efforts to make robots learn and develop behaviors. In this chapter, we will look at the literature on behavior development in these three scientific areas.

4.1 Behavior development in psychology

In psychology, under the subdiscipline of *developmental psychology*, Piaget is one of the most influential figures, with his *theory of cognitive development*. With his theory, Piaget investigated the character and development of human knowledge. He tried to understand how we acquire this knowledge, how *concepts* develop in our cognitive world, how these are related to behavioral development, and how certain high-level operations like logic develops.

According to Piaget the human cognitive development consists of successive stages [62]. Each stage constitutes the base on which the next stage will be built on. The first stages which start with sensorimotor schemata paves the way to conscious and voluntary behavior, and then to the acquisition of concepts and laws about the external world, and to formal logical thought. According to Piaget, in this transition from one stage to a higher stage, two basic processes take part: *assimilation* and *accommodation* [62]. These are two processes that complement each other. *Assimilation* emphasizes that the knowledge about external world experiences are acquired into the human cognitive system according to the already existing schemata in the system. In this sense, Piaget's theory is epistemologically a constructivist theory. That means, the knowledge about the external world is not acquired *as it is*, and

the perceptual information is not the only force that shapes our cognitive structures; but knowledge is acquired in accordance with - and through - the already existing structures. On the other hand, *accommodation* expresses how these structures change through interaction with the world, and experience. Because these already existing structures are also not the only factor that shape human cognition of course; experiences take part in shaping human cognition too. So, in a sense, *accommodation* is the opposing process to *assimilation*: it is the process that tries to make the existing cognitive structures coherent with the external world.

For Piaget, the development of behavior also occurs in this framework [63]. The newborn baby, who has innate reflexes, executes and tries these reflexes and primitive behaviors, trying to accommodate them to the environment. As the baby experiments with these behaviors, it can then use these as voluntary behaviors. To quote from Gelman [26]

For Piaget, neonates spontaneously practice their reflexes, the effect being the differentiation of inborn reflexes into different sensory-motor schemes. Active use of these yields integrated action schemes, and thus novel ways to act on the environment. (...) (For Piaget), an infant's initial knowledge is limited to innate reflexes and is combined with an inclination to actively use and adapt these as a result of repeated interactions with objects. This eventually leads to the development of intercoordinated schemes and movement to action-based representations that take the infant from an out-of-sight, out-of-mind stage to internalized representations (...).

E.J. Gibson was also interested in child development, and she was the first one to investigate affordances in the context of development [76]. She studied the mechanisms of the *learning of affordances* and used the ecological approach to study child development.

E.J. Gibson defined learning as a perceptual process and named her theory of learning "perceptual learning". She based the process of learning on perception, not on mentally constructed representations of the world. She argued that learning is neither the construction of representations from smaller pieces, nor the association of a response to a stimulus. Instead, she claimed, learning is "discovering *distinctive* features and *invariant* properties of things and events" [28] or "discovering the information that specifies an affordance" [30]. Learning is not "enriching the input", but discovering the critical perceptual information in that input. She named this process of discovery *differentiation*, and defined it as "narrowing down from a vast manifold of (perceptual) information to the minimal, optimal information

that specifies the affordance of an event, object, or layout” [30]. E.J. Gibson suggested that babies use exploratory activities, such as mouthing, listening, reaching and shaking, to gain this perceptual data, and that these activities bring about “information about changes in the world that the action produces” [28]. As development proceeds, exploratory activities become performatory and controlled, executed with a goal.

4.2 Motor control and learning in neuroscience

In executing a motor behavior, the central nervous system commands the muscles through the motor neurons. It is a very complex process including driving multiple muscles in a synchronized way, in the correct timing and order. To achieve this, the central nervous system must map the motor goals (for example the final position of the arm, the direction or velocity of the movement) to neuron signals controlling the muscles. This is a difficult problem, since it constitutes a mapping from a small number of variables to a large number of variables that drive multiple muscles [3]. The current studies on motor control and learning, essentially investigates how to solve this problem, and how to learn this mapping.

There are studies proposing that the central nervous system solves this problem using inverse kinematics. These are mathematical methods to determine the parameters that drive the muscles during the complete motion, based on the final desired position and orientation of the joints and muscles. But this point of view entails the determination of the parameters of the complete motion from the very start, so it is not flexible and can not explain momentary reactions during motor behavior execution [3]. Another approach to the problem of motor control suggests that the motion of the joints can be expressed in terms of transitions between “equilibrium positions” that are determined by the central nervous system. These “equilibrium positions” corresponds to positions where the forces in the opposing muscles that drive a joint are equal [4].

Another approach that tries to explain how complex patterns of motor behavior emerge says that, these complex patterns are actually the result of combining more simple primitive actions [3]. For example, in [56] Mussa-Ivaldi et al. found that when they stimulated separate modules in the spinal cord of a frog one-by-one, they correspond to a limited number of force patterns and motor actions. But when they stimulated two modules simultaneously, the resulting force pattern corresponded to the vector summation of the individual force patterns of each individual module. Through this, they showed that using a linear combination of a set of simple pre-coded force patterns, it was possible to generate a different complex motion.

Mussa-Ivaldi et al. interprets this as [56]: “this linear behavior is quite remarkable and provides a strong support to the view that the central nervous system may generate a wide repertoire of motor behaviors through the vectorial superposition of a few motor primitives stored within the neural circuits (...)”. According to Bizzi [3], “the limited-force pattern may be viewed as representing an elementary alphabet from which, through superimposition, a vast number of movements could be fashioned”.

Another study that supports this position is the influential work of “population coding” by Georgopoulos et al. [27]. Through experiments they conducted on rhesus monkeys Georgopoulos et al. found that the arm movements of the monkey can be predicted using the activation values of a population of neurons in the monkey brain. In this population of neurons, it was seen that each individual neuron has a preferred direction, and when it fires it makes the monkey arm move towards that direction. But when multiple of these neurons fire together, it was seen that the resulting direction of the monkey arm was a weighted sum of each individual neuron’s preferred direction. Moreover, these weights were given by the activation values of each neuron. So, the more a specific neuron fires, the closer is the direction of the monkey’s movement to the preferred direction of that neuron. That means each neuron contributes to the resulting direction, and the contribution is proportional to the activation value of that neuron. And the monkey’s movement is equal to this weighted vectorial sum.

We have seen that both in the studies of *developmental psychology* and in the studies of *motor control and learning* in neuroscience, the idea of starting from pre-coded primitive behaviors, and through training and development, achieving more complex behaviors is accepted as a possibility. If we combine the approach of developmental psychologists Piaget and E.J. Gibson (which says that a baby starts from innate primitive reflexes and enriches them through experience until they become voluntary action) with the approach of neuroscience (which says that complex patterns of motor behavior can be explained using combination of simple pre-coded behaviors), then we believe that this presents a very good research potential for robotic behavior development. In this kind of research, one should investigate how robots equipped with simple pre-coded(innate) behaviors can develop to achieve more complex behaviors through the usage of these simple behaviors. This actually constitutes the grounds where this thesis aims to make its contribution. But before going into the details of how we realized such a development, in the next section we will review the other robotics studies conducted in this area.

4.3 Behavior learning and development in robotics

In robotics, there have been increasing interest in behavioral development and learning in recent years. There are studies that makes a robot learn behavior parametrization [46], learn to use behaviors purposively [2, 22], and demonstrate stages of development through the usage of a fixed set of behaviors [60]. In this section we will present some of these studies.

In [44] Lee et al. use case-based reasoning in selecting parameters for their behaviors, for goal-directed navigation. In this study the robot has a “case-library”, where each case is indexed by environmental features and outputs a set of behavioral parameters. According to the values coming from its sensors, the robot tries to find a best candidate among the cases in the case library, that reflects the current environmental situation the most. After finding the best-matching case, the behavioral parameters of that case are used. In [44], the “case library” is created manually, which requires extensive experimentation and tedious work. In [46], Likhachev et al. extends this work by making the robot populate its case library through its own experience. In this work the robot starts with an empty case library. It creates new cases as it perceives set of environmental features that are not similar to the cases already existing in the library. But since the proper output behavior parameters of these new cases are not known by the robot at the beginning, they act rather randomly. By the help of an explicit performance evaluator, the performance of each case is computed, and a gradient-ascent search is made over the output behavior parameters of these cases. As the robot experiences more in the environment, the cases converge to the correct behavior parameters. Note that the external performance evaluator plays a crucial role in finding the proper parameters, providing reinforcement signals.

Another study that uses reinforcement learning is Asada et al.’s work on “purposive behavior acquisition” [2]. In this study, the robot has a fixed set of behaviors, and using these navigational behaviors it aims to shoot a ball into a goal. At the beginning the robot does not know when to execute which behavior in scoring goals; that is, it does not have any idea what its behaviors are good for. But through a reinforcement learning process, the robot learns using its behaviors purposively. Therefore, after training, the robot manages to select the correct behaviors in different situations, so that it gets closer to scoring goals.

In robotics, there are also studies that aims to mimic developmental stages that animals go through. In [60], Oudeyer et al. made a robot show different phases of cognitive development. In what they called “playground experiments”, Oudeyer et al. placed a robot-dog in a playground that included various simple toys. In this environment, by executing some

primitive behaviors randomly, the robot learned the dynamics and relation between its behaviors, and the events in the environment. When Oudeyer et al. also provided an external motivation to the robot to show interest in situations which are “neither too predictable nor too unpredictable”, the robot autonomously went through a developmental sequence. During this development, the robot’s complexity of activities increased at each stage.

In [14], Aarno et al. present a perceptual framework, using which they aim to make a robot learn about objects by interacting with them, like an human infant.

It is evident from all these studies that, there is an increasing interest in robotic behavior development and learning. Rather than the hand-coding of the behaviors of a robot by a human programmer, these new studies aims to make robots develop and learn their own behaviors.

CHAPTER 5

EXPERIMENTAL FRAMEWORK

In this chapter we will present the experimental framework of our study, before going into the details of our development scheme in the next chapter. Here, we will first present the robotics and simulation platform that we have used in our study. Then we will present the primitive behaviors implemented on our robot and give the details of the interaction environment where our robot will be trained. Lastly, we will present the details of the perceptual processes using which our robot gathers information from the environment. We will also see how our robot forms the *entity* and *effect* structures which are at the core our affordance formalization.

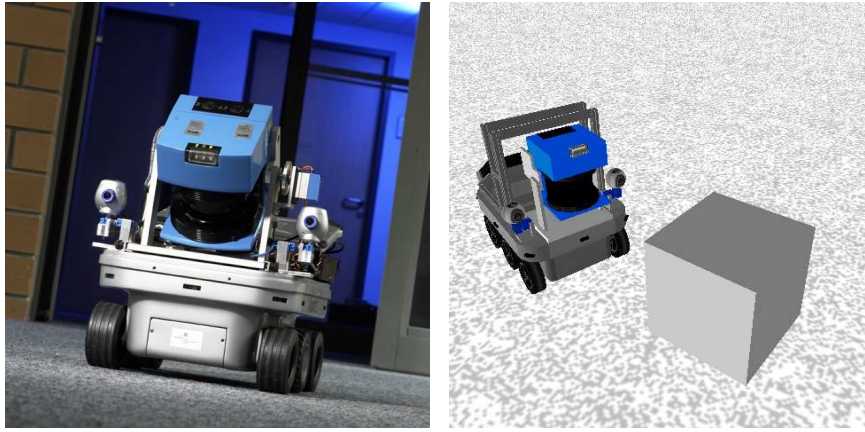
5.1 Robotic and simulation platform

The robotic platform KURT3D¹ and its simulated counterpart are used in this study. KURT3D is a differential drive mobile robot equipped with a 3D laser range scanner (based on the SICK LMS 200 2D laser scanner). Using this laser scanner, the robot is able to create a range image of its environment. The maximum horizontal range for the scanner is 180°, and the maximum vertical range is also 180°. At the finest resolution a range value for every quarter degrees can be delivered by the scanner, both in vertical direction and horizontal direction. That means, at the maximum resolution and at the widest vertical and horizontal range, the range image has the size 720 × 720. Such an image can be delivered in approximately 45 seconds. Of course, the resolution, vertical range, and horizontal range values can be set to lower values by the user. The robot also has encoders in its wheel system.

The simulator we used for KURT3D is MACSim [80] implemented using ODE (Open Dynamics Engine²) library. The KURT3D is modeled inside the virtual world, together

¹URL: <http://www.ais.fraunhofer.de/ARC/kurt3D/>

²URL:<http://ode.org/>



(a) KURT3D

(b) MACSim

Figure 5.1: (a) The KURT3D. (b) A snapshot from MACSim showing the KURT3D robot facing a box-shaped object.

Table 5.1: The motor parameter values passed by the primitive behaviors.

<i>Behavior</i>	<i>LeftWheel</i>	<i>RightWheel</i>	<i>Duration</i>
TURN_LEFT	-0.25 m/s	+0.25 m/s	0.7 sec
MOVE_FORWARD	+0.25 m/s	+0.25 m/s	1.7 sec
TURN_RIGHT	+0.25 m/s	-0.25 m/s	0.7 sec





with its sensors and actuators. The simulator also supports creation of several types of objects in the virtual world of the robot, so that the robot can interact with them. You can see a snapshot from the simulator in Figure 5.1.

5.2 Primitive behaviors

We implemented and used three primitive behaviors on the robot for our experiments. These are *move-forward*, *turn-left*, and *turn-right* behaviors. The *move-forward* behavior drives the robot straight ahead that places the robot approximately 40cm away from its initial position, if the move is not obstructed by any obstacles. The *turn-left*, and *turn-right* behaviors turns the robot in place for approximately 50°. These behaviors are implemented by driving the right and left wheels of the robot at a constant speed for a constant duration. The values we used to drive the motors can be seen in Table 5.1.

5.3 Interaction environment

In our learning phase each trial is performed with a single object in the environment. The objects are placed randomly within a proximity of $1m$ to the robot, in the frontal area spanning 180° . The interaction environment contains four types of simple objects:

- rectangular boxes (),
- spherical objects (),
- cylindrical objects, either
 - in upright position (), or
 - lying on the ground (),

An example interaction environment can be seen in Figure 5.1 where a box is placed in front of the robot.

After learning, the environment that the developed behaviors are tried in is not same environment, and it is cluttered with many objects. We will present an example of this environment in Section 6.3.1.

5.4 Perception and representation of entities and effects

The robot perceives its environment mainly through its 3D scanner. It uses the range images from the scanner to extract a set of features which consists the robot's perception of the environment. This feature-extraction process was first used in [81] by Uğur. Here, we use the same process to extract shape and distance related features from the range image.

The feature set is obtained in three steps as shown in Figure 5.2. The robot makes a full resolution scan of 720×720 . First, the image is down-scaled to a resolution of 360×360 pixels. Then, it is split into grids of size 12×12 pixels. This means that there are 900 such grids (since $(360/12)^2 = 900$), in total. Then, for each grid, distance and shape related features are extracted. The distance related features are the distance of the closest point, distance of the furthest point, and the mean distance of all the points within a grid. The shape related features are computed from the normal vectors in the grid. A normal vector for each point in a grid is computed using the range values. Then the direction of each normal vector is recorded in two base-dimensions, φ and θ , in latitude and longitude. Two angular histograms are computed for each of these dimensions. The histograms are sliced into 18

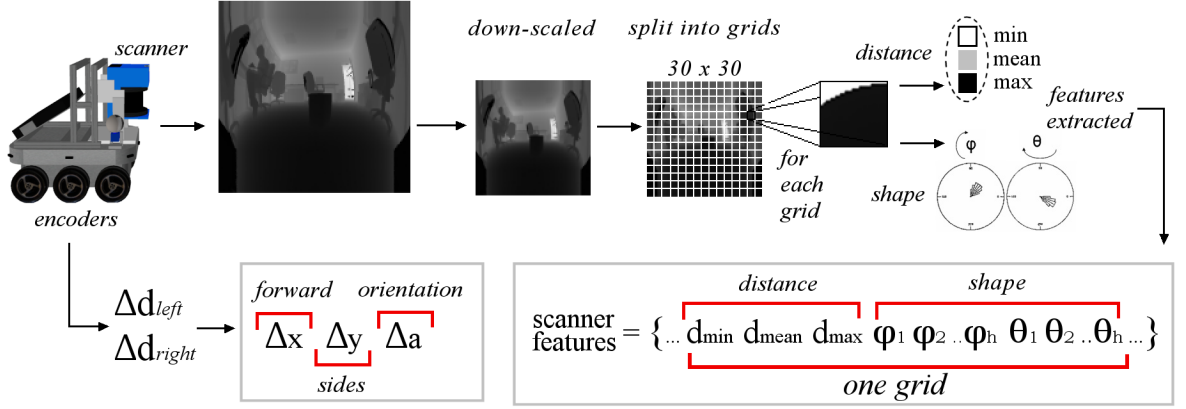


Figure 5.2: Phases of perception. Distance and shape features are extracted from the scanner range image. Also three displacement values are extracted from the encoders. (Figure taken from [17])

intervals of 20° each. And the frequency values in each of these slices of the histograms are used as the shape related features. Since there are two channels of 18 values each, there are 36 shape related feature for each grid. Adding the three distance related features of a grid, there are 39 features to represent a single grid. We mentioned that there are 900 such grids. So the total number of features to describe the scene becomes $900 \times 39 = 35100$. In addition to the scanner features, values from the wheel-encoders are also recorded.

In our formalization *entity* represents the state of the environment as perceived by the agent before performing a behavior. In this study it is represented with the scanner features obtained before the execution of a primitive behavior by the robot.

In our formalization *effect* represents the perceivable change in the environment or in the state of the agent, produced by performing a behavior. In this study, the *effect* is represented with the vectorial difference between the scanner features obtained after and before the execution of a primitive behavior of the robot, together with 3 more features extracted from the encoder values that correspond to the change of the robot's position in the forward and left-right directions, and the change in its orientation. (Figure 6.1)

CHAPTER 6

USING AFFORDANCES FOR ROBOTIC BEHAVIOR DEVELOPMENT

In this section we propose a behavioral development scheme on a robot. We will make use of the theory and formalization of affordances in achieving such a development. The generic affordance relations that are acquired by the robot will provide the basic building blocks of the development scheme we are going to propose.

How can we realize behavioral development for a robot? The robot will start with a set of primitive behaviors. These will be simple pre-coded behaviors. Initially, the robot will not have any more knowledge about its behaviors than to execute them blindly. Starting from the set of primitive behaviors, the robot will have no initial knowledge about when to apply these behaviors, and what kind of effects they create once they are applied. The robot first has to learn the possible effects it can create in the environment using these behaviors. It should also learn when to apply which behavior to create a specific change in the environment. Discovering the uses of its primitive behaviors, the robot can then utilize them in developing more complex behavior patterns.

In creating new more intelligent behavior from the primitive behaviors, the primitive behaviors (that the robot has learned about, and has done its training with) can be used in two ways as the result of development: sequentially, and simultaneously. In sequential usage, the primitive behaviors can be used as they are: there will be one single primitive behavior active at an instant, but the cumulative effect of the execution of these behaviors will form a goal-directed intelligent behavior on a wider time-scale. In the second way of using the primitive behaviors in behavior development, the simultaneous way of using the primitive behaviors, the primitive behaviors can be combined, such that, at an instant it is not any of the primitive behaviors that is executing, but a new behavior that has never

been seen or demonstrated by the robot before, yet is used by it intelligently to create effects in the environment that are more in accordance with its goals than any of the primitive behaviors. This is the second way of using the learned primitive behaviors that we will try to demonstrate.

In the following, we will present the implementation of this development scheme. In the next section a brief overview of the behavioral development stages as implemented will be given. Then we will see how the robot collects the *relation instances* by interacting with its environment, and how it merges them in generic affordance relations. After this we will present the usage of the learned affordance relations in developing goal-directed behaviors. As mentioned before, we will demonstrate two ways of doing this: sequential usage of the primitive behaviors, and simultaneous usage of the primitive behaviors.

6.1 Methodology

In this section we will present a brief overview of the processes involved in using the formalization of affordances described in the previous sections, in making a robot learn about its own primitive behaviors and use them goal-directedly. The underlying process consists of two phases: learning affordance relations, and achieving goal-directed behavior.

Learning affordance relations The phase of learning affordance relations in turn includes two steps: data collection (interaction with the world), and forming affordance relations using these collected data. In the data collection step the robot executes its primitive behaviors one at a time, in a certain environment. It perceives and records the environment before executing a behavior, and after executing it. This corresponds to collecting $(effect, (entity, behavior))$ instances. After executing all of its behaviors in many different environments, the robot has a collection many such relation instances. This collection is then used in forming the *affordance relations*. This requires forming *entity equivalence classes* and *effect equivalence classes* from the relation instances of a specific behavior, and connecting them in an *affordance relation*. Practically, these *affordance relations* are trained classifiers that can then be used to create goal-directed behavior.

Achieving goal-directed behavior In this phase the robot uses the learned *affordance relations* to achieve more intelligent behaviors. Perceiving the current environment provides a description of the *entity*. Using this *entity* and the learned affordance relations $(\langle effect \rangle, \langle (entity, behavior) \rangle)$, the robot can then execute the *behavior(s)* which will

result in the *effect* that will make the robot achieve its goal. We will present two approaches in developing goal-directed behaviors. In the first approach, the robot will use its primitive behaviors in a sequential manner to achieve goal-directed behavior. Practically this means feeding in the *entity* representation to the trained classifiers, of which there is one for each action. Then the predictions of each classifier, which are *effect-prototypes*, are compared to see which effect will be more useful in achieving the goal. The action, whose classifier predicted a better *effect*, is executed. In the second approach, the robot will use its primitive behaviors simultaneously to achieve goal-directed behavior. Practically, this will again correspond to feeding in the current *entity* representation to the trained classifiers, of which there is one for each action. Then the predictions of each classifier, which are *effect-prototypes*, are compared with the goal representation to see how similar each *behavior's effect* prediction is to the desired *effect*. The similarity values will then be used as weights for the behavioral parameters, in blending the primitive behaviors so that a new behavior emerges.

6.2 Learning affordance relations

In learning generic affordance relations, the robot first interacts with its environment, collects data about the interaction, and then applies certain machine learning processes over the data to form generic affordance relations.

6.2.1 Data collection: Interacting with the world

In the interaction phase, the robot collects affordance *relation instances*. Perceived entity and effect instances are linked together with the primitive behavior that was executed to produce the effect. The three constitute a *relation instance*. Figure 6.1 depicts the extraction of these instances.

Here, for each trial, the robot perceives the situation before applying a primitive behavior and stores it as the *entity*. It then performs one of its primitive behaviors. After that, it perceives the environment again, and extracts and stores the *effect* as described in 5.4.

In this study a total number of 3000 trials for each primitive behavior were performed in the simulator during the interaction phase. The details of the interaction environment was presented in 5.3.

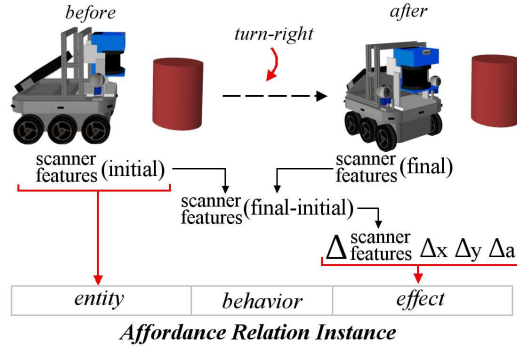


Figure 6.1: Representation of the entity and the effect. Distance and shape features extracted from the scanner image, taken before the execution of a primitive behavior, constitute the *entity*. The difference between the features extracted after the execution of the behavior and features extracted before the execution of the behavior constitute the representation of *effect*, together with the displacement values extracted from the encoders (see Figure 5.2).

6.2.2 Forming affordance relations

The aim of this phase is to derive *affordance relations* from the set of *relation instances* collected in the interaction phase, through the formation of equivalence classes. Within the set of *relation instances* of a behavior, similar effects are grouped together to get more general description of different kinds of effects that behavior can create. This is achieved through the unsupervised clustering of the effect instances. This corresponds to obtaining *effect equivalence classes*. After clustering, each *effect class* is assigned an *effect-id* and the *effect prototype* of the class is calculated.

Knowing the different kinds of *effects* that a behavior can create, the robot should then discover the *distinctive* features and *invariant* properties of the environments in which these effects are created. This corresponds to obtaining *entity equivalence classes*. This consists of two substeps. First, the robot selects the features describing the *entity* which are relevant in determining if a situation will result in one *effect* or another. This is achieved by applying a feature selection algorithm over the *entities*, using the corresponding *effect-ids* as their categories. Next, the robot learns the mapping from the *entities* to the *effect-ids*, for the execution of a behavior. This is achieved by training classifiers with the collected *affordance relation instances*. A separate classifier is trained for each behavior, using the *entity* (which now includes only the selected relevant features) as the input, and the corresponding *effect-id* of each instance as the target category.

In the rest of this section, we provide the details of these three steps in the learning phase.

Forming effect equivalence classes with clustering

A primitive behavior, when applied in different situations, creates different kinds of effects in the environment. Recognizing these different kind of effects is necessary if the robot is going to use the behaviors goal-directedly.

For this purpose, for each behavior, the 3000 *effect* data collected in the interaction phase were clustered using the k-means algorithm. The k parameter was experimentally set to 10. The k-means algorithm was applied with normalized distances to avoid the domination of scanner originated features over encoder originated features and shape related features over distance related features.

Figure 6.2 gives an interpretation of the results of clustering. After clustering, every *effect class* is assigned an *effect-id*. The *effect prototype* of a class is the mean of the individual *effects* in that class. The set of prototypes characterizes the different kinds of effects each behavior produces.

Selecting relevant features

The robot only needs the subset of features describing the *entity* which are important in determining if a situation will result in one *effect* or another. For this aim, we selected the relevant features in the *entity*, using the corresponding *effect-ids* as their labels. Selection of relevant features is done using the ReliefF algorithm, originally proposed by Kira and Rendell [43]. This method aims to estimate the weight of each feature in a feature set, based on its impact on the target category of the samples. In ReliefF, the weight of any feature is increased, if it has similar values for the samples in the same category, and if it has different values for the samples in different categories.

To speed-up this feature-selection process, instead of using the complete set of interaction samples, 50 samples from every class were randomly selected. We used the data-mining software WEKA [86] as an implementation of ReliefF.

In Figure 6.3, the grids corresponding to the relevant features for each behavior are given. It can be observed that the grids to which selected attributes belong, differ for each behavior.

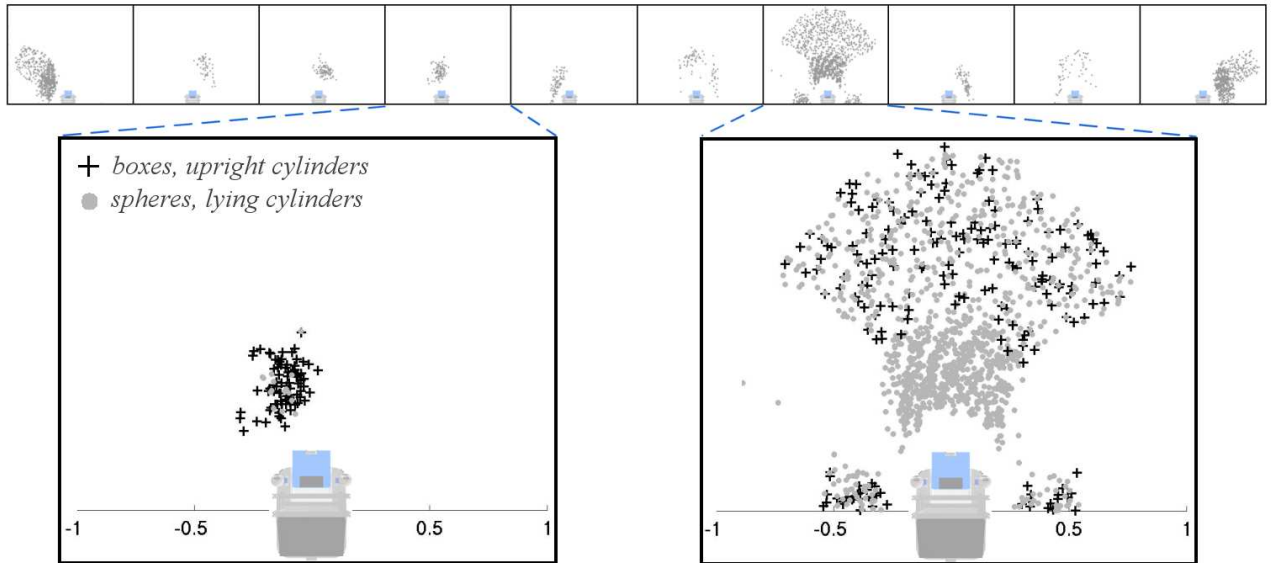


Figure 6.2: Interpretation of effect classes obtained with unsupervised clustering for the primitive behavior *move-forward*. The upper image contains the distribution of object positions in the interaction phase for the samples in the resulting 10 clusters. In the enlarged pictures the types of objects can also be observed. The left image corresponds to a cluster whose prototype effect has a small value for “change in the forward direction”. It can be observed that in the samples which belong to this cluster, the object was placed in front of the robot, and it was close to the robot such that that the robot would come in contact with the object during its forward motion. Moreover, the majority of these objects were boxes and upright cylinders, so that the robot’s motion would be blocked by the object. The right image, on the other hand, corresponds to a cluster whose prototype effect has a large change in the forward direction. This cluster contains interaction samples in which the object was either far enough, such that the robot would not get in contact with the object, or it was on the path of the robot’s motion but it was a sphere or a lying cylinder, so that it would be rolled away without blocking the motion. In the upper image, it can also be observed that clusters were formed according to the position of the object being roughly on the right or the left of the robot.

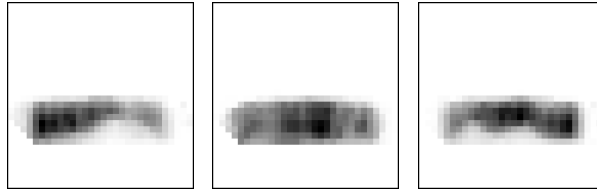


Figure 6.3: Relevant grids in the range image representation for three possible primitive behaviors: *turn-left*, *move-forward*, and *turn-right*. Darkness is an indication of relevance. It can be seen that only a small portion of all the grids are relevant for each behavior, and most of the grids are completely white, indicating no relevance. Also, for *turn-left* and *turn-right* actions, the grids on left and right, respectively, are more relevant.

Linking effects to entities

Support Vector Machines (SVMs) are trained to classify *entities* (which now include only the 2000 most relevant features selected in the previous step) into *effect* classes. We used the libSVM [8] library as an implementation of SVMs. For each behavior, an SVM was trained using the *entities* as the inputs, and the corresponding *effect-ids* of each instance as the target value. These SVM classifiers are then used in the execution phase, to predict what kind of *effect* a behavior will generate, given a perceptual representation (*entity*) of the current environment.

6.3 Achieving goal-directed behavior

In this section we will present the two ways of achieving goal-directed behavior, using the learned affordance relations: through the sequential use of primitive behaviors, and through the simultaneous use of primitive behaviors.

6.3.1 Achieving goal-directed behavior through the sequential use of primitive behaviors

In [17] we demonstrated how the learned affordance relations can be used in achieving goal-directed behaviors, through the utilization of the primitive behaviors sequentially. Here the robot uses the learned affordance relations in selecting the proper primitive behavior in a given context. Although the behaviors that are used by the robot are the same pre-coded primitive behaviors, since the robot now has the knowledge of what effects a behavior creates when applied in a certain context, it is able to use them to more intelligently. Therefore,

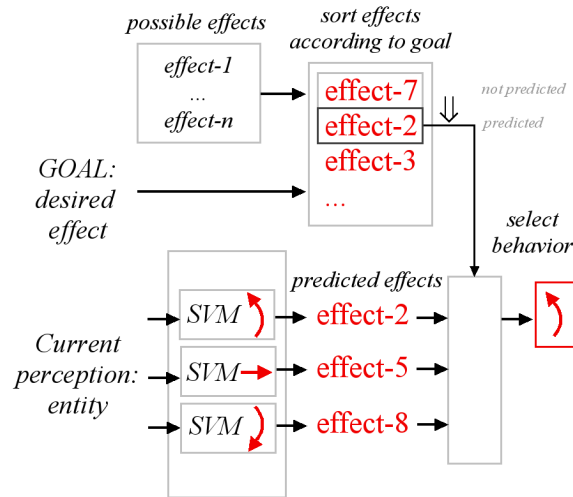


Figure 6.4: Behavior selection. The different possible *effects prototypes* are sorted according to the current desired effect. The current perception of the environment is supplied to the SVMs for each primitive behavior. The behavior, whose SVM predicts an *effect* that is higher in the sorted list, is executed.(Figure taken from [17])

although the behaviors are the same, when viewed on a wider time-scale the sequencing of the behaviors displays a certain goal-directed behavior. As a consequence many different goal-directed behaviors can be achieved using the same restricted set of primitive behaviors, and the same learned affordance relations. The robot learns the generic affordance relations once, but it can use them to achieve many different goal-directed behaviors.

In this section we will first explain how the behavior selection is done by the robot. Then we will present three examples for the goal-directed behaviors. The *traverse* behavior uses the “traversability” affordance in the environment for navigation. The *approach* behavior makes the robot go towards an object. The *avoid* behavior tries to avoid any contact with the objects to navigate in the environment.

Behavior selection

The robot uses the learned *affordance relations* to select the primitive behavior in achieving goal-directed behaviors. Given the perceptual representation of the current environment as an *entity*, the trained classifiers will predict an *effect-id* which indicates the effect class that the behavior, for which the classifier was trained, will produce in this environment. By comparing the *effect prototypes* of the predicted classes with its desired effect determined by its current goal, the robot can select the behavior that will produce the most useful effect in

achieving its goal. The behavior selection mechanism is shown in Figure 6.4.

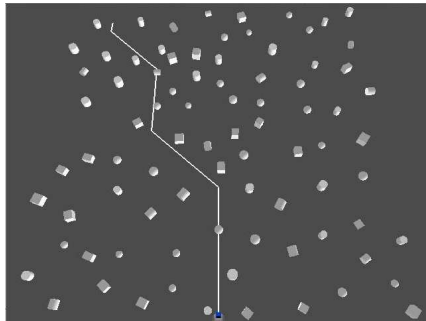
Specifying the current desired goal and sorting the effect prototypes according to this desired goal is what results in different behaviors. This goal specification and assigning priorities to the possible effects can be done in different ways. The difference between the current situation and the desired goal gives us a description of the *desired effect*. We can then sort the *effect prototypes* according to their similarity to this *desired effect*. Another possibility is to assign priorities to certain effect prototypes directly, by using a global evaluation criteria. The behaviors that will be demonstrated in the next section use such a method. In the next section, we will present these behaviors together with the criteria we used to evaluate the possible *effects prototypes* in achieving these behaviors.

Goal-directed Behaviors

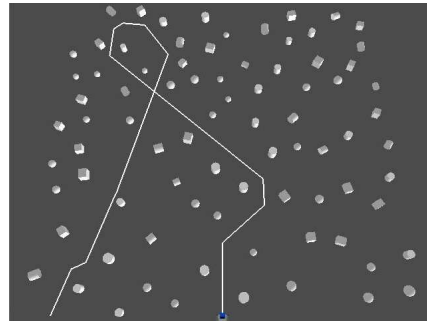
Traverse We define “traversing” as different from the conventional “wandering using obstacle avoidance”. In the obstacle-avoidance approach every object that is on the robot’s path is considered as an obstacle. In fact such a behavior is overly cautious compared to its intentions. For example if the object on its path is a ball that can be easily pushed away then there is no need for the robot to avoid it. Here the robot has to make the judgment of whether the situation affords “traversability” or not.

Given our robot and the objects in its environment, we can say that the environment is “traversable” for the robot, if there is an open-space, or a sphere (since it can be rolled away), or a cylinder lying on the ground in a proper orientation so that it can be rolled away. The other cases of when there is a box, or a cylinder in an orientation such that it can not be rolled away are the “non-traversable” cases.

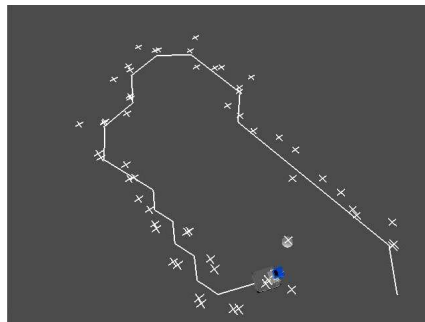
If we want our robot to explore the environment using traversability, it should be able to drive onto (by executing forward motion) traversable objects and open spaces but avoid (by *turn-left* or *turn-right*) non-traversable objects. This can be achieved by a specific ordering of the effect classes. In this case the most desired effect is the forward displacement of the robot but without being stopped by an object. This means that the highest priority should be given to the effect classes whose prototypes have a forward-displacement value greater than a threshold. Then must come the effect classes for the two turning motions *turn-right* and *turn-left*. Lastly, as the most undesired cases, the effect classes of the forward motion whose prototypes have a forward-displacement value smaller than the threshold should come, since this small value is an indication of the motion’s being stopped by an obstacle, thus a non-traversable case. In our implementation we set this threshold to a value just below 40cm,



(a) Traverse behavior



(b) Avoid behavior



(c) Approach behavior

Figure 6.5: Three different behaviors achieved using the same three primitive behaviors and their learned *affordance relations*. In (a), the robot wanders around perceiving the traversability affordance of the objects. When there is a sphere or a cylinder in a rollable orientation on its way, the robot rolls it away and continues forward-motion. When there is a box or a cylinder in non-rollable orientation on its way, the robot avoids it by turning left or right. In (b), the robot displays a more typical obstacle-avoidance behavior, where it avoids all the objects, whether it is rollable or not. In (c), an example path where the robot follows an object using its *approach* behavior is shown. The plus signs marks the places that objects appear. The line shows the robot's path.

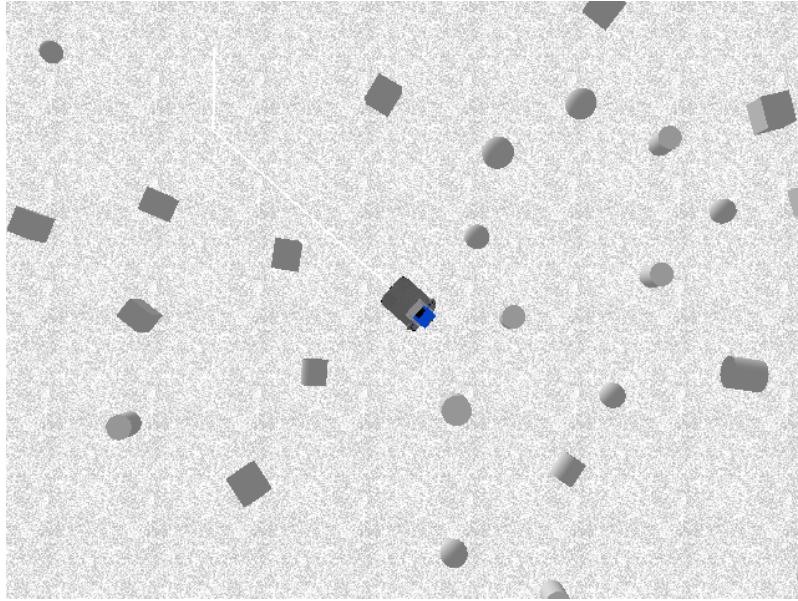


Figure 6.6: Snapshot from the arena where the performance of the behavior was measured. Here, only a small portion of the whole arena can be seen.

which is the distance the robot covers when it executes an unobstructed `MOVE_FORWARD` behavior.

We have tested the *traverse* behavior by placing the robot in an environment randomly filled with multiple traversable and non-traversable objects. The robot successfully explored the environment and also used the traversability affordance of the objects by rolling away the traversable objects on its way, and avoiding the non-traversable ones. One example path of the robot can be seen in Figure 6.5(a).

We also measured the performance of the traverse behavior for different number of interactions in training the robot and analyzed the effect of the increasing number of interactions (increasing experience) on the performance of the robot.

We measured the performance of the traversability behavior as follows: In the simulator, we placed the robot in a rectangular arena of $44m \times 48m$. The arena was cluttered with 1000 simple objects randomly. You can see a portion of a sample arena in Figure 6.6. The robot started its trial at the center of this arena and executed the traverse behavior. The actions the robot executed and the distance it covered was recorded at each step. The execution of a trial stopped when the robot got stuck to an obstacle when executing the *move-forward* behavior (it is not possible for the robot to get stuck when executing *turn-left* or *turn-right* since they simply make the robot turn in place). When the robot got stuck the total distance

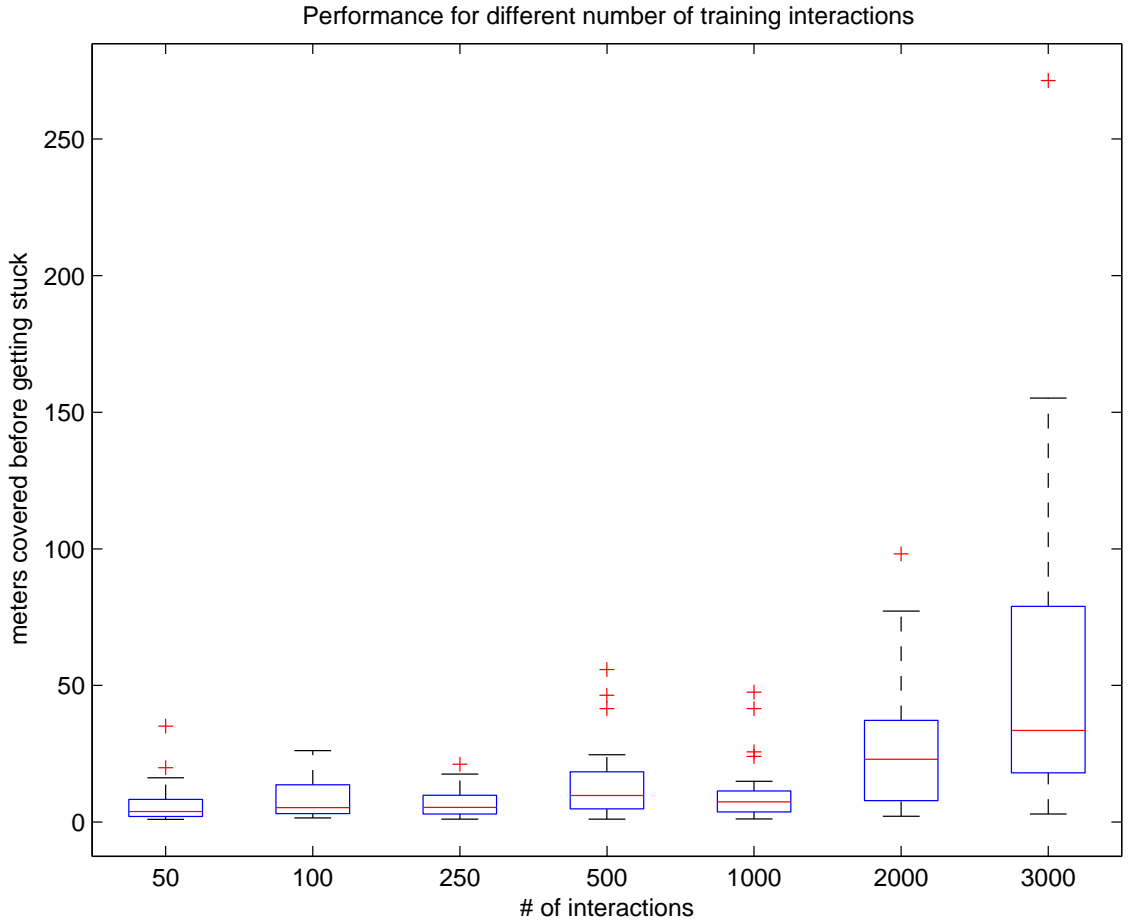


Figure 6.7: Performance for different number of training interactions. As the robot interacts with the environment more, the developed behavior performs better. When there are a few hundreds of interactions, the increase in the performance is not too clear, but when the robot reaches thousands of interactions it begins to perform remarkably better.

the robot has covered so far in that trial was recorded. If the robot got out of the arena borders without getting stuck, it was placed at the center again with a different distribution of objects in the arena, and the trial was continued until the robot got stuck.

We manipulated the number of training interactions the robot did with the environment to see its effect on the performance. For example when training the robot, we stopped at the 50th interaction, went through the learning process to form the affordance relations, and measured the performance of the robot. Then the robot continued to experience with the environment until 100th interaction, where we again measured the robot's performance. We did this at the 50th, 100th, 250th, 500th, 1000th, 2000th, and 3000th interactions. For

measuring the performance at each case we conducted 40 trials in the arena. One can say that the number of interactions corresponds to the “experience” of the robot during its development. So one expects an increasing performance with experience. In Figure 6.7 we can see that this is indeed the case. In this figure we can see an increase in the performance of the robot as its number of interactions increase. In fact, when there are a few hundreds of interactions, the increase in the performance is not too clear. But when the robot reaches thousands of interactions the robot begins to perform remarkably better.

Approach Approaching an object means going forward if the object is ahead, turning right if the object is on the right, and turning left if the object is on the left. In this view, the most desired effect would be to see an appearance, or approach, of objects in the middle portion of the 3D-scanning field. Recall that the 3D scan field is a 30×30 grid in our representation of the effect. We selected the horizontally middle portion of this grid. For every effect class, these grids holds the information about the change in the values of the features in the frontal region of the robot, when the corresponding behavior is executed. The priority of an effect class is assigned based on the sum of the change in the mean-distance features in these grids. Since the distance value is smaller when an object is close, the higher priorities are given given to those classes with the most negative value of this sum. This way the effect classes, which correspond to approaching or turning to an object so that it is ahead, becomes higher in the sorted effect list.

We have tried this *approach* behavior first by placing objects to random places in front of the robot. It was observed that the robot was able to make the correct decision of going ahead if the object is in the front, turning right if the object is on the right, and turning left if the object is on the left. Next we have simulated a slowly moving object in front of the robot, by placing the object on random positions in front of the robot as the robot made its moves. An example path of the object and the robot can be seen in Figure 6.5(c).

Avoid As a third behavior a more traditional approach to the traversability problem was employed. The rollability of certain objects was not taken into consideration and the robot tried to avoid contact with any object in the environment. To achieve this behavior the priority of an *effect* was assigned in exactly the opposite way as it was in the *approach* behavior. So the sorting of the effect classes was based on the sum of the change in the mean-distance features in the frontal region of the robot, which is the horizontally middle portion of the 30×30 grid in our representation of the effect. This way, the effect classes

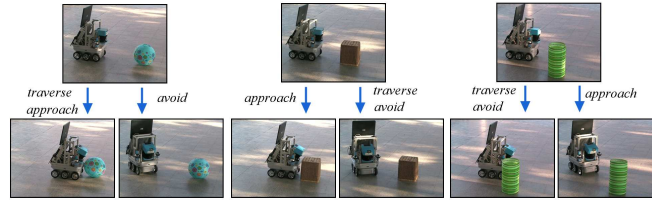


Figure 6.8: Three cases in which different goal-directed behaviors (*traverse*, *avoid*, *approach*) make use of different primitive behaviors (move-forward, turn-right, turn-left) in the same setting of the environment.

which correspond to turning away from an object that is ahead becomes higher in the sorted list of effects; and the effect classes, which correspond to approaching or turning to an object so that it is ahead, becomes lowest in the sorted list of effects. But this criteria was not enough to make the robot wander around, since it always tried to turn away from objects (by executing *turn-left* or *turn-right*) even if they were very far away, and never executed move-forward. So we disabled this sorting when there were no objects close in front of the robot, and made the robot execute the *move-forward* behavior in these cases. The path of the robot with this behavior is given in Figure 6.5(b).

The three goal-directed behaviors were also realized on a real robot. The trained controllers were transferred to a real KURT3D robot, and everyday objects like balls, trash bins, etc. were placed in front of the robot to test the behaviors. The robot was able to perceive the traversability of objects, so it rolled away the balls on its way, and avoided non-traversable objects like trash-bins. The robot was also able to display the *approach* and *avoid* behaviors as described in the previous sections. Figure 6.8 shows how the three goal-directed behaviors react in different environments ¹.

6.3.2 Achieving goal-directed behavior through the simultaneous use of primitive behaviors

In the previous section the robot was able to use a set of primitive behaviors such that when viewed on a wider time-scale the robot's behavior corresponded to goal-directed intelligent behaviors. But while we claimed that in achieving such kind of goal-directed behavior the robot made use of a generalization over the *effects* it can create, and a generalization over the features of the *entities* it interacts with, it can not be said that the robot made use

¹For a complete demonstration of the behaviors on the real robot, please see the video at <http://kovan.ceng.metu.edu.tr/~mehmet/iros07.mpeg>.

of a generalization over the *behaviors*. At any given moment the robot executed a single primitive behavior, and these primitive behaviors were the same behaviors that it also used during the training interactions, and they were programmed into the robot.

In this section we will try to achieve *behavioral generalization* (or more correctly, a generalization over the motor control parameters of the behaviors), so that, after training, the robot will not be constrained with the fixed set of pre-programmed behaviors but will be able to demonstrate novel behaviors. While the robot will still have a limited set of behaviors during training, after training it will react to situations with new behaviors, that are more effective than the primitive behaviors in creating the desired effect in the environment.

To be able to do that, the robot needs to make a generalization over the motor parameters that it uses for the behaviors, and relate these to the effects it can create with these parameters. Then, when it needs to create a specific effect in the environment, the robot can choose the correct set of parameters to create the effect.

We achieved this generalization using a weighted sum of the motor control parameters of the primitive behaviors, where the weights are determined according to the similarity of their effects to the desired effect. The inspiration for this approach comes from the work on *population coding* (See Section 4.2).

In the next section a general description of the method used for behavioral generalization will be given. Then, a concrete implementation of this method for a specific behavior will be presented.

Method

The method proposed uses motor parameter values of the primitive behaviors in determining a new value for that motor parameter. Each primitive behavior passes values to the motors as control parameters, and these values are fixed by the programmer for each behavior. For example, the TURN_LEFT behavior passes -0.25 m/s as the left-wheel-speed and +0.25 m/s as the right-wheel-speed, and the TURN_RIGHT behavior passes +0.25 m/s as the left-wheel-speed, and -0.25 m/s as the right-wheel-speed, etc.. During training only one primitive behavior is active, so the values that are passed to the motors are the ones of that behavior. After training these fixed values of the primitive behaviors are again used, but this time, at an instant, all the primitive behaviors contribute to the value that is actually passed to the motor. The weight of each primitive behavior's contribution is determined by the similarity of the predicted effect of that primitive behavior to the desired effect. As a result if the predicted effect of a primitive behavior is very similar to the desired effect, then

it contributes more to the values passed to the motors and vice versa.

Suppose that there are n primitive behaviors B_1, B_2, \dots, B_n , and each behavior B_i has a set of motor parameter values $v_{i1}, v_{i2}, \dots, v_{im}$ for each of the m motors M_1, M_2, \dots, M_m . Further suppose that D is the desired effect prototype, and p_1, p_2, \dots, p_n are the predicted effect-category prototypes in the current environment for each of the n behaviors. Also, let's say that there is a similarity function S that takes two effect prototypes as arguments and returns a value indicating the similarity between these two prototypes. Then, in an arbitrary environment, we can find the new value v'_j to be passed to motor M_j as:

$$v'_j = \sum_{i=1}^n \frac{S(D, p_i)}{\sum_{k=1}^n S(D, p_k)} * v_{ij} \quad (6.1)$$

That is, the resulting motor parameter value is the sum of each behavior's contribution for that parameter, and this contribution is proportional to the similarity of the predicted effect for that behavior to the desired effect. Note that, other than the learned affordance relations, we also need to define a similarity function that would indicate how similar a predicted effect is to the desired effect.

Implementation

An implementation of the proposed approach has been tried on the *approach* behavior. In the *approach* behavior presented in the previous chapter, the robot was using its primitive behavior *TURN_LEFT* when an object was on its left, *TURN_RIGHT* when an object was on its right, and *MOVE_FORWARD* when an object was ahead. While this approach succeeded in making the robot approach to objects, the robot was limited with the use of these three behaviors only. In this section we will present the results of applying the *behavioral generalization* method to the *approach* behavior. In this case, the robot is still trained with the same three behaviors, but after training it is able to demonstrate novel behaviors that enables it to create its desired effect more successfully than any of the primitive behaviors that it has been trained with.

The three primitive behaviors that the robot is trained with are *TURN_LEFT*, *MOVE_FORWARD*, and *TURN_RIGHT*. Table 6.1 shows the motors and the parameter values each behavior uses.

All three behaviors use the left wheel and right wheel motors with different parameter values. The robot is trained with these primitive behaviors as described in Sec. 6.2.

For the behavioral generalization approach, we also need to define a similarity function

Table 6.1: The motor parameter values passed by the primitive behaviors for behavioral generalization.

<i>Behavior</i>	<i>LeftWheel</i>	<i>RightWheel</i>	<i>Duration</i>
TURN_LEFT	-0.25 m/s	+0.25 m/s	0.7 sec
MOVE_FORWARD	+0.25 m/s	+0.25 m/s	0.7 sec
TURN_RIGHT	+0.25 m/s	-0.25 m/s	0.7 sec

that will return higher values for *effect-prototypes* that are similar to our desired effect. Since the goal of the behavior is approaching the object we used the same function we defined to assign priorities to *effect-prototypes* in Sec. 6.3.1. The robot should be approaching objects, so the most desired effect would be to see an appearance, or approach, of objects in the middle portion of the 3D-scanning field. And since the 3D scan field is a 30×30 grid in our representation of the effect, we selected the horizontally middle portion of this grid. For every effect class, these grids holds the information about the change in the values of the features in the frontal region of the robot, when the corresponding behavior is executed. So, the similarity value of an effect prototype is assigned based on the sum of the change in the mean-distance features in these grids. Since the distance value is smaller when an object is close, the higher similarity values are given to those classes with the most negative value of this sum. This way the effect prototypes, which correspond to approaching or turning to an object so that it is ahead, gets higher similarity values. Before using these similarity values as weights for the motor parameters, we mapped the lowest similarity value to 0 and normalized the values for each primitive behavior to the range $[0, 1]$.

The results of applying this behavioral generalization approach can be seen in Figure 6.9, Figure 6.10, and Figure 6.11. In these figures the robot’s reaction to different situations are shown, for the two strategies of using only primitive behaviors, and using behavioral generalization. It can be seen that the behavior generalization approach enables the robot to discover new behaviors different than the primitive behaviors, and these new behaviors improves the robot’s performance in situations where the primitive behaviors are not good enough. An exception is the case in Figure 6.11 where the object is placed directly ahead of the robot. In this case the primitive behavior MOVE_FORWARD approaches the object better than the behavior executed using the behavior generalization method. But note that the place where the object is placed in this case is exactly the right position for

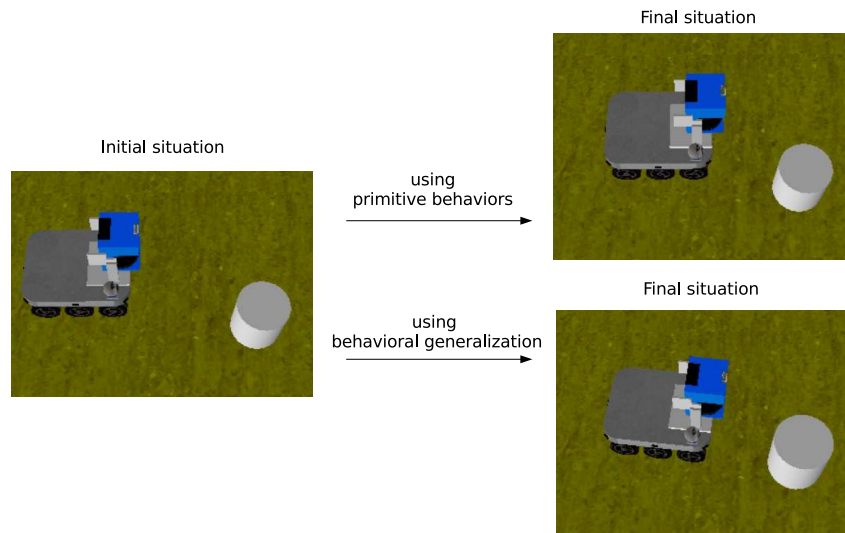


Figure 6.9: The object is placed 20° to the right of the robot, at a distance of 30cm . When using only the primitive behaviors to approach the object, the robot chooses to execute `MOVE_FORWARD` behavior. When using the behavioral generalization method, the robot makes a smoother motion towards the object which approaches the object more successfully. Actually this movement is a blending of the `MOVE_FORWARD` and `TURN_RIGHT` primitive behaviors, where the contribution of the `MOVE_FORWARD` behavior is more than `TURN_RIGHT` behavior.

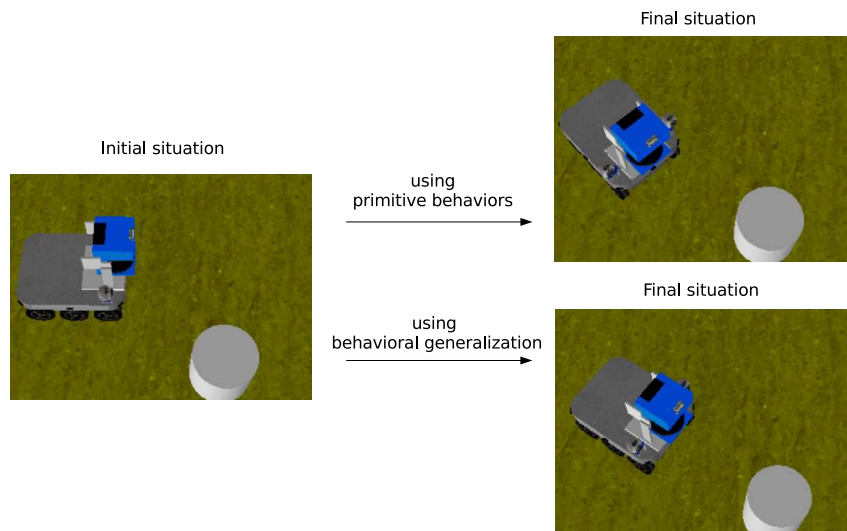


Figure 6.10: The object is placed 45° to the right of the robot, at a distance of 30cm . When it uses only the primitive behaviors, the robot chooses the `TURN_RIGHT` behavior to approach the object. When it uses the behavior generalization method the robot again makes a smoother motion towards the object which approaches the object more successfully. This movement is also a blending of the `MOVE_FORWARD` and `TURN_RIGHT` primitive behaviors, but different from the case in Figure 6.9, this time the contribution of the `TURN_RIGHT` behavior is more than the contribution of the `MOVE_FORWARD` behavior.

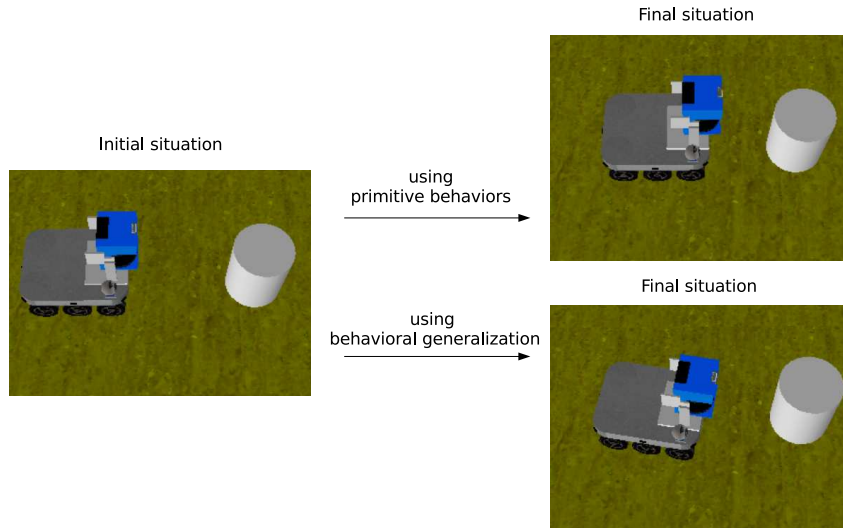


Figure 6.11: The object is placed directly ahead of the robot. This time the primitive behavior executed seems to be more successful than the behavioral generalization strategy, since the purity of the MOVE_FORWARD behavior seems to be tempered by some contribution from the TURN_RIGHT behavior.

MOVE_FORWARD behavior, and while the positions like these are limited with the number of primitive behaviors, there are infinitely many positions where the primitive behaviors will not suffice and behavioral generalization will perform better.

This can be seen more clearly in Figure 6.12. In this figure it can be seen that, when compared with using only the primitive behaviors, the behavioral generalization approach spans the same angular range in turning towards the object, but it does so in a more finer manner, spanning whole of the angular range. The trade-off is some lose of precision in the directions of the original primitive behaviors.

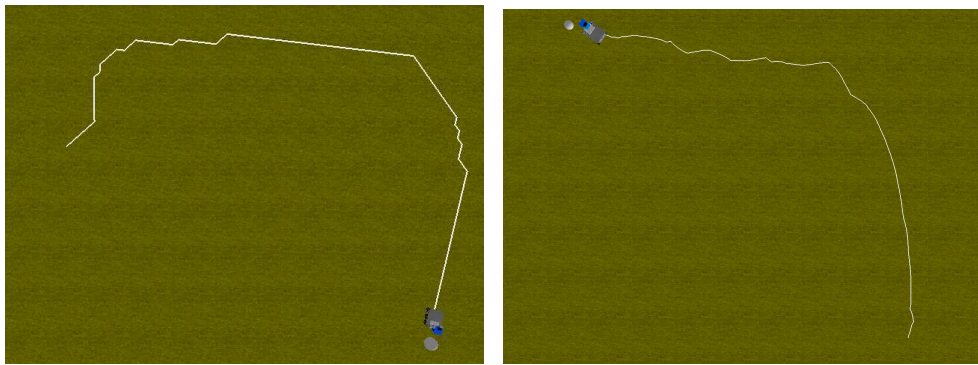
To see the behavior of the proposed approach on a more global scale, we placed a sphere in front of the robot let the robot execute its behaviors many times. What emerged was a behavior where the robot dribbles the sphere(ball) in front of it. The path of the robot when it rolls the ball can be seen in Figure 6.13.



(a)

(b)

Figure 6.12: Robot's reaction to different situations using the two different strategies of using only primitive behaviors, and using behavioral generalization. The arrows show the robot's position and heading direction after executing the behavior. The circles denote the object's position in each different case. If a circle and an arrow are of the same color, this means that when the object is in the location indicated by the circle, the robot's heading direction and position after executing the chosen behavior is indicated by the arrow of the same color. In (a) the robot uses only the primitive behaviors in approaching the object. Therefore, in the figure, there are only three arrows, representing the robot's position and heading direction after executing each of these three behaviors. It can be seen that the robot is able to approach the object and select the correct primitive behavior. But one can also notice that these primitive behaviors are very crude in turning towards the object. In (b) the robot uses the behavioral generalization strategy in turning towards the objects. In this figure there are eight arrows, corresponding to eight different reactions of the robot to different situations. Here again the robot is successful in turning towards the object, but this time it makes more detailed movements towards the objects showing an improvement over the case of using only the primitive behaviors.



(a)

(b)

Figure 6.13: Robot dribbling a ball using the learned affordance relations. In the simulator artificial perturbations are applied to the ball while it is rolling, so that it does not roll perfectly in one direction when the robot hits it. In (a) the path when robot uses only the three primitive behaviors. In (b) the behavioral generalization method is employed. It can be seen that this results in a smoother trajectory for the robot.

CHAPTER 7

DISCUSSIONS AND CONCLUSION

In this thesis we proposed a behavior development scheme for a mobile robot. We believe that, instead of trying to build intelligent machines from scratch, we should build robots that are equipped with simple capabilities that can bootstrap the system. Then, the robot should interact with its environment and collect its own experiences. As a result of this interaction the robot should develop capabilities and behaviors that were not coded into the initial system.

J.J. Gibson’s concept of “affordances” [33] provided the basis of our proposed development scheme. We used a formalization of affordances [13] to make the robot learn about the dynamics of its interactions with its environment. In this formalization, every interaction of the robot with the environment are represented as (effect, (entity, behavior)) triples. Collecting such *affordance relation instances* from the environment, our robot was then able to extract generic *affordance relations* pertaining to the relation between itself and the environment. Using these learned *affordance relations* our robot displayed higher-level behaviors.

In this thesis we implemented such a development scheme using three pre-coded behaviors: *move_forward*, *turn_left*, and *turn_right*. Using these primitive behaviors, our robot interacted with simple objects like boxes, cylinders, and spheres. Then, using the data it collected during its interactions with the environment, our robot formed *affordance relations*. In our implementation this practically corresponded to training SVMs that can predict the effects that will be created in the environment if a certain behavior is executed, in the current environment. Then, these trained SVMs were used by the robot to display more intelligent behaviors in the environment.

We tried two different methods in achieving more complex behaviors using the three simple pre-coded behaviors. As the first method we used the *sequential* execution of the

primitive behaviors. In this case, the robot uses its pre-coded primitive behaviors only, but the *sequencing* of these primitive behaviors were such that new more complex behaviors emerged. As the second method we used the *simultaneous* execution of primitive behaviors. Here, the robot uses its pre-coded primitive behaviors to generate *new* behaviors that are more effective in reaching its goal than any of the primitive behaviors, through blending of the primitive behaviors. This is achieved by driving the motors of the robot using a value which is equal to the weighted sum of the motor parameters of each primitive behavior. The weight (contribution) of each primitive behavior is proportional to the similarity of the predicted effect for that behavior to the desired effect the robot wants to create.

We showed that using these strategies, our robot was able to develop different higher-level behaviors using its three primitive behaviors and the learned affordance relations. First our robot demonstrated the “traverse” behavior, using which it was able to wander around perceiving the “traversability” of the objects in the environment. We also showed that the performance of this behavior increased with the number of training interactions. As a second example, the robot demonstrated a classical obstacle-avoidance behavior. Here, it avoided contact with any object in the environment while wandering around. The third behavior was the “approach” behavior, where the robot approached and drove towards the objects. We also showed that using the *simultaneous* execution of the primitive behaviors, our robot was able to develop *novel* behaviors in approaching the object.

Discussion and comparison with other robotic behavior learning approaches

Lastly, we will compare the framework and the behavior development scheme we have proposed, with the other approaches in robotics and machine learning in general. This will be conceptual discussion rather than a quantitative comparison. In machine learning, there are three main learning paradigms: unsupervised learning, supervised learning, and reinforcement learning [19]. We will discuss where our study falls in the space that these three axes of learning paradigms constitute, and its pros and cons in relation to them.

Unsupervised learning [19] is the name given to a set of methods, that are used in forming clusters in a collection of data, according to some similarity measure. Even though unsupervised learning is used in robotics studies in various ways, these are mostly at the perceptual level, restricted to the methods used on perceptual feature data, and the action data are not used. Therefore, if the aim is to make the robot learn some behaviors/actions, then unsupervised learning methods do not prove enough by themselves. In this restricted way, we also made use of an unsupervised clustering method (k-means), as a part of our learning

framework.

In *supervised learning* [19] there is a teacher, that gives the correct answers to the learning agent, in the course of the training. Even if it is possible to use supervised methods in robotic behavior learning, the implementation of these training schemes requires a very tedious work. That is because the robot will need a teacher that tells it what reaction to give in which situation, and generally, this teacher is either a human controlling the robot continuously, or a control program that is already demonstrating the behavior that we are trying to make the robot learn. Moreover, in such a learning scheme, the behavior that the robot learned would be one that is trying to imitate the performance of the human or the program that is controlling it, but that could not ever over-perform it. These issues create difficulty in making use of totally supervised approaches in training robots. Even so, this type of supervised methods are used, for example in bootstrapping reinforcement learning [68].

In our learning framework, supervised learning exists, not as the direct supervised teaching of a specific behavior, but as the usage of a supervised learning method: SVMs are trained, such that, given a description of the environment they tell the robot what kind of an effect it will be able to create in the environment, if it executes a specific primitive behavior. But in our framework the target values that are supplied during the training of the SVMs (that is, the supervision), are not supplied by a human or some “intelligent” controller. In this process, the structures that the robot uses are extracted using an unsupervised clustering method over the *effect* data it collected during its interactions. In this respect, our robot learns its behaviors in an autonomous way, without the need of an external teacher.

Reinforcement learning [75] is the most widely used paradigm in robotic behavior learning [2, 50, 41], among the three learning paradigms we have discussed. In reinforcement learning, there is not a teacher that tells the agent what to do in which situation, but rather, the robot gets “rewards” when it reaches certain states, and “punishments” when it reaches some other states. The agent’s main goal is to maximize the total amount of rewards it receives in the long-term.

In reinforcement learning, the robot has to be trained separately for each different behavior it is to execute, whereas this is not the case for our behavioral development scheme. In reinforcement learning, even if a robot is trained very well and is very successful for a specific behavior, one will still need to train it for different behaviors, since the training is task-dependent. The primary difference of our work from the studies that use reinforcement learning to make a robot learn a behavior is that, rather than aiming to make the robot learn a specific behavior, our work proposes a generic development scheme. This difference

becomes obvious when one notices that, the training our robot goes through is independent from the behaviors that it is able to display at the end. That becomes possible, because during training, our robot learns generic relations about the interactions of its body and the environment. These structures are task-independent, and holds the actual information about the effects the robot can create in its environment, using its primitive behaviors. Therefore these structures can then be used to achieve several different behaviors.

The structure and utilization of state representations in reinforcement learning also differ from the way the state representations are used in our learning framework. For reinforcement learning to take place, one has to map the situations of the environment to state representations. Ideally, these state representations are discrete, and as in [2], this is achieved through the mapping of environmental situations to a set of discrete states by the human programmer, employing domain knowledge. In our learning framework, the complete *entity* representation consist of 35100 raw features, and these raw representations are also mapped to discrete state representations (*entity equivalence classes*) implicitly. But this mapping is not due to a human programmer who has the expert knowledge as in reinforcement learning. Instead, this mapping is discovered by the robot itself, using the clustered *effect* representations which includes similar effects that the robot can create in the environment.

In reinforcement learning, the ‘utility’ values of each state are learned by the agent, whereas in our study the actual *effects* the robot can create at each state are learned. The ‘utility’ values in reinforcement learning are meaningful only for a specific behavior, and indicates the amount of reward the robot can receive from that state. But in our development framework, by using an *effect* representation that is unifiable with the state representation (*entities*), our robot is able to predict the future changes in the environment, and this constitutes also a base for planning [7].

Lastly, our work is not an isolated behavior learning study, but is the part of a bigger developmental framework for robotics [6], which includes robot control, learning, planning, and conceptual development. In this framework, by making a robot learn generic affordance relations between its behaviors and the environment, we investigate how such an approach would perform in solving various problems in robotics [13, 79, 17, 78, 81, 7].

REFERENCES

- [1] R.C. Arkin. *Behavior-based Robotics*. MIT Press, Cambridge, MA, USA, 1998.
- [2] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, V23(2):279–303, May 1996.
- [3] E. Bizzi. Motor control. In R. Wilson and F. Keil, editors, *The MIT Encyclopedia of Cognitive Sciences*. Cambridge, MA: MIT Press/Bradford Press, 1998.
- [4] E. Bizzi, N. Accornero, W. Chapple, and N. Hogan. Posture control and trajectory formation during arm movement. *Journal of Neuroscience*, 4(11):2738–2744, November 1984.
- [5] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, (47):139–159, 1991.
- [6] M. Çakmak, M.R. Doğar, E. Uğur, and E. Şahin. Affordances as a framework for robot control. In *7th International Conference on Epigenetic Robotics, EpiRob'07*, 2007. Submitted.
- [7] Maya Çakmak. Robot planning based on learned affordances, 2007. MSc thesis, Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey.
- [8] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001.
- [9] A. Chemero. What events are. *Ecological Psychology*, 12(1):37–42, 2000.
- [10] A. Chemero. An outline of a theory of affordances. *Ecological Psychology*, 15(2):181–195, 2003.
- [11] A. Chemero, C. Klein, and W. Cordeiro. Events as changes in the layout of affordances. *Ecological Psychology*, 15(1):19–28, 2003.

- [12] S. Cornus, G. Montagne, and M. Laurent. Perception of a stepping-across affordance. *Ecological Psychology*, 11(4):249–267, 1999.
- [13] E. Şahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk. To afford or not to afford: A new formalization of affordances towards affordance-based robot control. *Adaptive Behavior*, 2007. (in press).
- [14] Danica Kragic Nicolas Pugeault Sinan Kalkan Florentin Wörgötter Dirk Kraft Norbert Krüger Daniel Aarno, Johan Sommerfeld. Early reactive grasping with second order 3d feature relations, 2007. Presented in ICRA'07, Workshop: “From features to actions”.
- [15] P. Doherty, T. Merz, P. Rudol, and M. Wzorek. Combined deliverable report: D4.2.1 and d4.3.1. Technical report, Linköping University, Dept. of Computer and Info. Science, 2005. MACS Project Deliverable, draft, version 1.
- [16] G. Dorffner, J. Irran, F. Kintzler, and P. Poelz. Robotic learning architecture that can be taught by manually putting the robot to action sequences. Technical report, The Austrian Research Institute for Artificial Intelligence (OFAI), 2005. MACS Project Deliverable 5.3.1.
- [17] M. R. Doğar, M. Çakmak, E. Uğur, and E. Şahin. From primitive behaviors to goal-directed behavior using affordances. In *To appear in the Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'07)*., November 2007.
- [18] A.P. Duchon, W.H. Warren, and L.P. Kaelbling. Ecological robotics. *Adaptive Behavior*, 6(3):473–507, 1998.
- [19] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
- [20] J.A. Effken and R.E. Shaw. Ecological perspectives on the new artificial intelligence. *Ecological Psychology*, 4(4):247–271, 1992.
- [21] L. Fadiga, L. Fogassi, G. Pavesi, and G. Rizzolatti. Motor facilitation during action observation: a magnetic stimulation study. *Journal of Neurophysiology*, 73:2608–2611, 1995.

- [22] P. Fitzpatrick, G. Metta, L. Natale, A. Rao, and G. Sandini. Learning about objects through action -initial steps towards artificial cognition. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation, ICRA*, pages 3140–3145, 2003.
- [23] G. Fritz, L. Paletta, M. Kumar, G. Dorffner, R. Breithaupt, and R. Erich. Visual learning of affordance based cues. In S. Nolfi, G. Baldassarre, R. Calabretta, J. Hallam, D. Marocco, J-A. Meyer, and D. Parisi, editors, *From animals to animats 9: Proceedings of the Ninth International Conference on Simulation of Adaptive Behaviour (SAB)*, LNAI. Volume 4095., pages 52–64, Roma, Italy, 25–29 September 2006. Springer-Verlag, Berlin. in press.
- [24] V. Gallese. Agency and motor representations: new perspectives on inter-subjectivity. SC working papers 2000-6. Workshop on autism and the Theory of Mind. Lyon, May 2000.
- [25] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119:593–609, 1996.
- [26] R. Gelman. Cognitive development. In R. Wilson and F. Keil, editors, *The MIT Encyclopedia of Cognitive Sciences*. Cambridge, MA: MIT Press/Bradford Press, 1998.
- [27] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, September 1986.
- [28] E.J. Gibson. Perceptual learning in development: Some basic concepts. *Ecological Psychology*, 12(4):295–302, 2000.
- [29] E.J. Gibson. Where is the information for affordances? *Ecological Psychology*, 12(1):53–57, 2000.
- [30] E.J. Gibson. The world is so full of a number of things: On specification and perceptual learning. *Ecological Psychology*, 15(4):283–288, 2003.
- [31] E.J. Gibson, G. Riccio, M.A. Schmuckler, T.A. Stoffregen, D. Rosenberg, and J. Taromina. Detection of the traversability of surfaces by crawling and walking infants. *Journal of Experimental Psychology*, 13(4):533–544, 1987.
- [32] J.J. Gibson. *The senses considered as perceptual systems*. Houghton Mifflin, Boston, 1966.

- [33] J.J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986. Originally published in 1979.
- [34] S.T. Grafton, M.A. Arbib, L. Fadiga, and G. Rizzolatti. Localization of grasp representations in humans by pet: 2. observation compared with imagination. *Experimental Brain Research*, 112:103–111, 1996.
- [35] J. Greeno. Gibson’s affordances. *Psychological Review*, 101(2):336–342, 1994.
- [36] S Harnad. The symbol grounding problem. *Physica D*, 1990.
- [37] G. Humphreys. Objects, affordances ... action !!! *The Psychologist*, 14(8):5, 8 2001.
- [38] J. Irran, F. Kintzler, and P. Pölz. Grounding affordances. In R. Trappl, editor, *Cybernetics and Systems*. Austrian Society for Cybernetic Studies, 2006.
- [39] H. Jaeger and T. Christaller. Dual dynamics: Designing behavior systems for autonomous robots. In S. Fujimura and M. Sugisaka, editors, *Proceedings International Symposium on Artificial Life and Robotics (AROB)*, pages 76–79, Beppu, Japan, 1997.
- [40] K.S. Jones. What is an affordance? *Ecological Psychology*, 15(2):107–114, 2003.
- [41] Leslie Pack Kaelbling, Michael L. Littman, and Andrew P. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [42] J.M. Kinsella-Shaw, B. Shaw, and M.T. Turvey. Perceiving walk-on-able slopes. *Ecological Psychology*, 4(4):223–239, 1992.
- [43] K. Kira and L.A. Rendell. A practical approach to feature selection. In *ML92: Proc. of the 9. Int. workshop on Machine learning*, pages 249–256, 1992.
- [44] J. Lee, M. Likhachev, and R. Arkin. Selection of behavioral parameters: Integration of discontinuous switching via case-based reasoning with continuous adaptation via learning momentum, 2002.
- [45] M.A. Lewis and L.S. Simo. Certain principles of biomorphic robots. *Autonomous Robots*, 11:221–226, 2001.
- [46] M. Likhachev, M. Kaess, and R. Arkin. Learning behavioral parameterization using spatiotemporal case-based reasoning, 2002.

- [47] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, December 2003.
- [48] D. Lyons and M. Arbib. A formal model of computation for sensory-based robotics. *IEEE Transactions on Robotics and Automation*, 5(3):280–293, 6 1989.
- [49] K.F. MacDorman. Responding to affordances: Learning and projecting a sensorimotor mapping. In *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, pages 3253–3259, San Fransisco, California, USA, 2000.
- [50] Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. In *National Conference on Artificial Intelligence*, pages 768–773, 1991.
- [51] L.S. Mark. Eyeheight-scaled information about affordances: A study of sitting and stair climbing. *Journal of Experimental Psychology: Human Perception and Performance*, 13(3):361–370, 1987.
- [52] J. McGrenere and W. Ho. Affordances: Clarifying and evolving a concept. In *In Proceedings of the Graphics Interface*, pages 179–186, Toronto, 2000. Canadian Human-Computer Communications Society.
- [53] C.F. Michaels. Affordances: Four points of debate. *Ecological Psychology*, 15(2):135–148, 2003.
- [54] R.R. Murphy. Case studies of applying gibson’s ecological approach to mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(1):105–111, 1999.
- [55] R.R. Murphy. *Introduction to AI Robotics*. MIT Press, 2000.
- [56] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi. Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences, USA*, 91(16):7534–7538, August 1994.
- [57] D.A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.
- [58] D.A. Norman. Affordance, conventions, and design. *Interactions*, 6(3):38–42, 1999.
- [59] R. Oudejans, C. Michaels, B. vanDort, and E. Frissen. To cross or not to cross: The effect of locomotion on street-crossing behavior. *Ecological Psychology*, 8(3):259–267, 1996.

- [60] P-Y. Oudeyer, F. Kaplan, V. V. Hafner, and A. Whyte. The playground experiment: Task-independent development of a curious robot. In D. Bank and L. Meeden, editors, *Proceedings of the AAAI Spring Symposium on Developmental Robotics, 2005*, pages 42–47, Stanford, California, 2005.
- [61] V. N. Papudesi and M. Huber. Learning behaviorally grounded state representations for reinforcement learning agents. In *Proc. of EpiRob'06*, 2006.
- [62] J. Piaget. Piaget's theory. In P. H. Mussen, editor, *Carmichael's Manual of Child Psychology*. New York: Wiley, 1970.
- [63] J. Piaget. *The Essential Piaget*. New York: Basic Books, 1977. edited by H. E. Gruber and J. J. Voneche.
- [64] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3:131–141, 1996.
- [65] G Rizzolatti and M. Gentilucci. *Motor and visual-motor functions of the premotor cortex*, pages 269–284. Chichester: Wiley, 1988.
- [66] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. 0-13-103805-2. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.
- [67] J.T. Sanders. An ontology of affordances. *Ecological Psychology*, 9(1):97–112, 1997.
- [68] William D. Smart and Leslie Pack Kaelbling. Effective reinforcement learning for mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2002)*, volume 4, pages 3404–3410, May 2002.
- [69] M. Steedman. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy*, 25, 2002.
- [70] T.A. Stoffregen. Affordances and events. *Ecological Psychology*, 12(1):1–28, 2000.
- [71] T.A. Stoffregen. Affordances and events: Theory and research. *Ecological Psychology*, 12:93–107, 2000.
- [72] T.A. Stoffregen. Affordances as properties of the animal environment system. *Ecological Psychology*, 15(2):115–134, 2003.

- [73] A. Stoytchev. Behavior-grounded representation of tool affordances. In *In Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 18–22, Barcelona, Spain, April 2005.
- [74] A. Stoytchev. Toward learning the binding affordances of objects: A behavior-grounded approach. In *In Proceedings of AAAI Symposium on Developmental Robotics*, pages 21–23, March 2005.
- [75] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 1998.
- [76] A. Szokolszky. An interview with Eleanor Gibson. *Ecological Psychology*, 15(4):271–281, 2003.
- [77] M.T. Turvey. Affordances and prospective control: an outline of the ontology. *Ecological Psychology*, 4(3):173–187, 1992.
- [78] E. Uğur, M.R. Doğar, M. Çakmak, and E. Şahin. Curiosity-driven learning of traversability affordance on a mobile robot. In *Proc. of ICDL'07*, 2007.
- [79] E. Uğur, M.R. Doğar, M. Çakmak, and E. Şahin. The learning and use of traversability affordance using range images on a mobile robot. In *Proceedings of IEEE Intl. Conf. on Robotics and Automation (ICRA 07)*, pages 1721–1726, Rome, Italy, 2007.
- [80] E. Uğur, M.R. Doğar, O. Soysal, M. Çakmak, and E. Şahin. MACSim: Physics-based simulation of the KURT3D robot platform for studying affordances. Technical report, Dept. of Computer Eng., Middle East Tech. Univ., Ankara, Turkey, 2006. MACS Project Deliverable 1.2.1.
- [81] Emre Uğur. Direct perception of traversability affordance on range images through learning on a mobile robot, 2006. MSc thesis, Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey.
- [82] W.H. Warren. Perceiving affordances: Visual guidance of stair climbing. *Journal of Experimental Psychology*, 105(5):683–703, 1984.
- [83] W.H. Warren and S. Whang. Visual guidance of walking through apertures: body-scaled information for affordances. *Journal of Experimental Psychology*, 13(3):371–383, 1987.

- [84] R. Wehner. “Matched filters” - neural models of the external world. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 161(4):511 – 531, 1987.
- [85] A.J. Wells. Gibson’s affordances and turing’s theory of computation. *Ecological Psychology*, 14(3):140–180, 2002.
- [86] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.

APPENDIX

List of related publications

- [1] E. Şahin, M. Çakmak, M.R. Doğar, E. Uğur and G. Üçoluk. To afford or not to afford: A new formalization of affordances towards affordance-based robot control. *Adaptive Behavior (in press)*.
- [2] M.R. Doğar, M. Çakmak, E. Uğur and E. Şahin. From Primitive Behaviors to Goal-Directed Behavior Using Affordances. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007 (accepted)*.
- [3] E. Uğur, M.R. Doğar, M. Çakmak and E. Şahin. The learning and use of traversability affordance using range images on a mobile robot. *In proceedings of ICRA'07, IEEE International Conference on Robotics and Automation, 2007*.
- [4] M. Çakmak, M.R. Doğar, E. Uğur and E. Şahin. Affordances as a Framework for Robot Control. *EpiRob 7, 7th International Conference on Epigenetic Robotics, 2007 (submitted)*.
- [5] E. Uğur, M.R. Doğar, M. Çakmak and E. Şahin. Curiosity-driven Learning of Traversability Affordance on a Mobile Robot. *In proceedings of ICDL 2007, 6th IEEE International Conference on Development and Learning, 2007*.