# ROBOT PLANNING BASED ON LEARNED AFFORDANCES

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

BY

MAYA ÇAKMAK

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

JULY 2007

**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 | : | Maya Çakmak |
| Signature | : | |

# ABSTRACT

ROBOT PLANNING BASED ON LEARNED AFFORDANCES

Çakmak, Maya
M.Sc., Department of Computer Engineering
Supervisor: Asst. Prof. Dr. Erol Şahin

July 2007, 74 pages

This thesis studies how an autonomous robot can learn affordances from its interactions with the environment and use these affordances in planning. It is based on a new formalization of the concept which proposes that affordances are relations that pertain to the interactions of an agent with its environment. The robot interacts with environments containing different objects by executing its atomic actions and learns the different effects it can create, as well as the invariants of the environments that afford creating that effect with a certain action. This provides the robot with the ability to predict the consequences of its future interactions and to deliberatively plan action sequences to achieve a goal. The study shows that the concept of affordances provides a common framework for studying reactive control, deliberation and adaptation in autonomous robots. It also provides solutions to the major problems in robot planning, by grounding the planning operators in the low-level interactions of the robot.

Keywords: Affordances, Planning, Learning, Autonomous Robotics

# ÖZ

ÖĞRENİLMİŞ SAĞLARLIKLARA DAYALI ROBOT PLANLAMASI

Çakmak, Maya

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Erol Şahin

Temmuz 2007, 74 sayfa

Bu tezde, otonom bir robotun ortam ile olan etkileşimlerinden ortamın sağlarlıklarını nasıl öğrenebileceği ve öğrendiği sağlarlıkları kullanarak nasıl plan yapabileceği çalışılmıştır. Çalışma, sağlarlıkların, erkin ve ortam arasındaki etkileşime mahsus bir bağıntı olduğunu öne süren yeni bir sağlarlık biçimlendirmesine dayanmaktadır. Robot, farklı nesneler içeren ortamlarla basit hareketlerini uygulayarak etkileşir ve ortamda yaratabildiği değişiklikleri ve bu değişikliğin belirli bir hareket ile yaratılmasını sağlayan sabit ortam özelliklerini öğrenir. Bu, robota, gelecekteki etkileşimlerinin sonuçlarını öngörme ve düşünerek plan yapma kabiliyetini kazandırır. Çalışma, sağlarlık kavramının, robotlarda tepkisel denetimin, düşünmenin ve öğrenmenin çalışılabileceği ortak bir çerçeve sunduğunu göstermektedir. Ayrıca, plan gösterimlerini robotun alt seviye etkileşimlerine dayandırarak, robot planlamasındaki önemli bir soruna çözüm sunmaktadır.

Anahtar Kelimeler: Sağlarlık, Planlama, Öğrenme, Otonom Robotbilim

# ACKNOWLEDGMENTS

First, I would like to thank my advisor Erol Şahin for giving me the opportunity to work on such an interesting and important problem and for his supervision and guidance throughout this work. I am grateful that I took his advice of getting my M.Sc. degree in Computer Engineering, and I would like to thank him for making it possible to complete my scientific prep. and M.Sc. course work, together with my thesis, in two years. Most of all, I am indebted to him for making me a member of the KOVAN family.

Secondly, I would like express my gratitude to two very special people: Onur Soysal, for being there every time I lost my motivation to work, for believing in me more than myself, for teaching me almost everything I know in Computer Science, and for his unconditional love, and Münire Sağat, for sharing hardship, for getting my mind off work by making me laugh more than anyone can, for helping me in any way she could and for being my best friend for more than 10 years.

I am grateful to all members of the KOVAN Lab for creating an excellent research environment as well as for their true friendship. I would like to acknowledge Mehmet Remzi Doğar's contributions on my thesis, and thank him for answering all of my technical questions, and for the valuable and inspiring discussions. I am deeply thankful to Emre Uğur for setting the best examples in every aspect and for saving me from trouble several times. I am indebted to Ali Emre Turgut, for sharing his wisdom on every subject and for being the one to propose me the position at KOVAN. My sincere thanks go to Hande Çelikkanat, for her sympathy and admiration which has been a great source of motivation, Fatih Gökçe, for his eagerness in helping me on any matter and for always being a source of positive energy and Yaman Çakmakçı for bringing laughter and joy to the lab atmosphere.

Finally, I would like to thank all my family, especially my parents Mariepaule and Kenan Çakmak, for encouraging all my decisions and for their endless love and support.

This work was 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

# LIST OF FIGURES

FIGURES

# LIST OF TABLES

TABLES

# CHAPTER 1

# INTRODUCTION

Autonomous robots that will be operating in dynamic and unstructured environments are required to satisfy some minimal competences. They should be:

- *reactive*; to be able to sense and respond fast in a dynamic environment,
- *deliberative*; to be able to solve complex tasks they are faced with, and
- *adaptive*; to be able not only to tune their responses to a changing environment, but also to discover its capabilities in relation to its operating environment.

Different lines of research on autonomous robotics have produced advances towards one or two of these competences, but not all. Early research on the hierarchical robot control paradigm, which is influenced by the classical Artificial Intelligence (AI) research, produced robots with deliberative capabilities that allow them to generate plans for complex real-world tasks. However these robots have been criticized for their lack of responsiveness and reliance on abstracted world models. Reactive robot control paradigm (which has led to the behavior-based approaches) with its emphasis on the embodiment and situatedness of robots, produced robots that can respond fast in dynamic environments. Yet, their lack of ability to deliberate limited them to simple tasks and operations. Research on learning and evolution have provided mechanisms of adaptation which are used at different levels of robot control, mostly as an add-on feature with little concern on how it may be integrated to support the other two competences.

Recently, through the emergence of hybrid robot control architectures, the link between reactive and deliberative paradigms has been established. However, the hybrid approach introduced new problems arising from the fact that deliberative processes use symbol systems as models of the world and the robot's actions, while reactive behaviors directly map sensory input to a motor output without relying on any kind of representation. This presented a major challenge: formulating the interface between the symbolic planning component and the reactive control of a robot. As a consequence of this, in these architectures, the design of reactive behaviors, as well as the the planning operators that holds information regarding the “workings of the world” are mostly hand-coded. Adaptation is mostly studied as an add-on feature and confined to abstracted problems within these architectures. Although a number of specialized robots have been constructed using the hybrid approach, the solutions proposed for interfacing planning with reactive control tend to be ad hoc and do not provide a general framework.

The MACS (Multi-sensory Autonomous Cognitive Agents) project proposes to achieve the basic competences in autonomous robotics with the concept of affordances. Introduced by J.J. Gibson, affordances, refer to the action possibilities that the environment inherently offers the agent interacting with it. The objectives of the project include “making affordances a first-class concept in all levels of robot control architectures” and “developing affordance-based control as a method for robotics”. The project claims to provide a new way to connect reasoning and learning with reactive robot control by interfacing perception and action in terms of affordances. This thesis is part of the research conducted within the MACS project.

Specifically, the thesis studies how an autonomous robot can *learn* affordance representations from its interactions with the environment to make it *reactive* to dynamic situations and provide abstract representations to be used for *deliberation*. It is based on a new formalization of affordances which proposes that affordances are relations that pertain to the interactions of an agent with its environment [10]. The formalism suggests that such relations can be acquired from interaction instances and it implies that these relations can be used in planning. Earlier studies have demonstrated how affordances can be acquired and used in direct perception [14] and goal-directed reactive behavior [12], based on the proposed formalism. This study presents an implementation of the acquisition approach and the use of acquired relations in planning. The study shows that the concept of affordances provides solutions to major problems in robot planning which are related to accounting for the interaction of the robot with its environment in its plan operators and autonomously acquiring these operators.

In the rest of the thesis, first the concept of affordances, as proposed by J.J. Gibson will be described, and the studies from a wide range of fields which were influenced by this concept will be reviewed. In chapter 3, the planning problem within the context of robotics will be presented and its philosophical problems will be pointed out. Chapter 4 will present the new formalization of affordances. Chapter 5 will describe the robot platform and the experimental framework in which the work was carried out. The proposed methodologies and experimental results towards the learning of affordances and their uses in robot planning are presented in Chapters 6 and 7. The conclusions and future directions of research will be provided in Chapter 8.

# CHAPTER 2

# THE CONCEPT OF AFFORDANCES

The term *affordance* [24] was coined by the perceptual psychologist James J. Gibson who was interested in how the environment is perceived in terms of its meanings for certain behavior. J.J. Gibson suggests that, entities in the environment, such as objects or surfaces, are perceived in terms of their affordances, like throwability or walkability. The concept is best understood through examples. For a human, a rigid horizontal surface affords walking-on, a door affords passing through, a bench affords sitting on, a stone affords throwing and etcetera.

J.J. Gibson formed the theory of affordances during his studies with pilots, where he was responsible for performing perceptual tests to assess whether a person would make a good pilot. It was then believed that a strong depth and distance perception capability was essential for an adequate performance in landing a plane, considered one of the toughest task for a pilot. J.J. Gibson's criticism was that, the conditions in which the tests were being conducted were significantly different from the conditions in which the task of landing is performed, above all, it is stationary unlike the high-speed conditions in which a plane is landed. He realized that the meaningful optical variable for a pilot landing a plane is the *optical center of expansion* of his visual field, as it indicates the direction of the glide [23]. It occurred to J.J. Gibson that studying perception as an isolated process was a mistake and that perception should be studied in relation to the behavior that it serves, within the context in which the behavior is performed. Although J.J. Gibson's focus was on visual perception, his approach of studying perception of organisms within their natural environment and his emphasis on the complementarity of the organism and the environment, influenced other psychologists and originated the school of Ecological Psychology.

J.J. Gibson's new theory was opposing to the contemporary theories of perception, which were based on recognition and classification of objects in the environment. Instead, J.J. Gibson suggested that the environment is perceived in terms of its opportunities for actions. In his commonly quoted writing, Gibson describes affordances as follows.

> "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)

Affordances emphasize the close relation between perception and action. The reciprocity of perception and action is inherent in the relation between an organism and its environment [23]. J.J. Gibson believed that perception is designed for action, implying the co-evolution of perception and action in a way that they serve each other.

Affordances are relative to the organism. Although affordances are often understood as properties of the environment, they are meaningless when considered without reference to the organism. J.J. Gibson explicitly pointed out that he coined the term affordance as something that refers both to the organism and the environment. Indeed, affordances are identified relative to the organism. For instance, a stone is said to afford grasping for a human because it has a size about the same size of human hand, has a shape that fits the hand and has a weight compatible with the arm force of a human. A stone that affords grasping and throwing for a human, does not offer the same affordances to a mouse. It affords different actions such as hiding behind.

J.J. Gibson argued that affordances are directly perceived by the organism through meaningful variables in the sensory stream. With the term "direct perception", he meant an "information pick-up" process that does not require any mental representation or inference. He suggested that only the relevant information is picked up from the environment, saying "perception is economical". He believed that animals pick the minimal information that is critical for guiding their behavior. For example, perception of the sittability affordance does not require the recognition of an entity as a chair or a bench. Instead, we directly perceive sittability through a set of invariants, such as a flat rigid surface roughly parallel to the ground, at a height around the knee height and etc. Consequently, a wall or a box can be perceived to be sittable, though they are not intended for sitting.

The concept of affordances has influenced studies from various fields. The next section reviews studies within Developmental Psychology, Cognitive Science and Neuroscience. The last two sections are devoted for discussing the relation of the concept of affordances with robotics and planning, and presenting affordance-related studies within these fields.

## 2.1 Affordance-related Research

### 2.1.1 Developmental Psychology

Affordances of an animal change throughout its lifetime due to both its physical and mental development. Animals adapt to the growth of their body, gain control over their muscles and learn specific motor patterns to achieve their goals. For example, while a pen hardly affords holding for a newborn, it affords writing for an adult human. This suggest that animals can acquire new affordances.

The role of affordances in perceptual development was primarily pronounced by E.J. Gibson. Her theory of perceptual development evolved hand-in-hand with J.J. Gibson's theory of affordances and was influenced by it. She did not agree with the view that learning, in general, could be explained with the principles of conditioning [57]. Particularly for perceptual learning, she believed that the underlying process was differentiation, i.e. the specification of significant information, rather than construction or association. To her, perceptual learning is the discovery of distinctive features and invariant properties and what is learned are affordances [22].

E.J. Gibson pointed out the crucial role of exploratory activities in perceptual learning [22]. She suggested that infants are innately "exploratory creatures" searching for new information. She noted that the exploratory activities bring about the information of the changes in the environment produced by an action and that infants develop an anticipation of the outcome of their actions. With the discovery of affordances, the exploratory activities become performatory, she suggests.

### 2.1.2 Cognitive Science

Besides being able to reciprocally perceive and act in an environment, human and certain animal species possess cognitive abilities, such as problem solving, inference, planning, imagination, abstraction and etcetera. The role of affordances in these high-level processes is an open question.

Followers of J.J. Gibson often criticize the use of the term "affordance" as part of high-level cognitive processes and object to the idea of their internal represention within the agent. Although, J.J. Gibson himself did not say much about the role of affordances in cognition, he did not imply the isolation of affordances from cognition. His remark that the perception of an affordance does not require a world representation or inference, should not be interpreted as a claim that it cannot be explicitly represented and consciously utilized. Although some researchers consider this approach to be opposing to J.J. Gibson, a few believe that it is consistent with Gibson's theory, and that indeed some examples in J.J. Gibson's writings imply so.

One example from J.J. Gibson's writings that imply the relation of affordances with cognition is the commonly referred “mailbox” example. Saying that a mailbox affords sending letters, J.J. Gibson certainly agrees that a “mailbox”, which is something that one can recognize only with the knowledge of the concept, has certain affordances. Steedman argues that the perception of such an affordance cannot be direct in the Gibsonian sense [51]. Again, Greeno believes that the perception of that affordaces includes the classification of the mailbox [26].

A few studies attempted to establish theories that relate affordances to high-level cognitive processes. Neisser proposed a two-layered perceptual system [40] in which the lower layer performs direct perception of affordances while the higher level, working in parallel, is responsible for perception related to cognitive abilities such as problem solving. In a similar vein, J. Norman suggested that perception is based on two interacting systems, one of which picks the information about affordances to modulate actions, while the other is concerned with high-level perceptual tasks [43]. Duchon and Warren postulated that the non-inferential information that lies in the perception-action association guiding an agent's actions, can also be used as a base and a restriction for different aspects of cognition such as planning and reasoning [13].

### 2.1.3 Neuroscience

The concepts of affordances, especially with its aspect of direct perception, has attracted the interest of researchers from the fields of Neurophysiology and Neuropsychology. Several studies relate parts of the brain with the perception and use of affordances and provide evidence that can support our understanding of the concept.

J. Norman [43], presents a study with a patient who lacks a ventral system in her brain. Due to this lack, the patient is unable to recognize the objects she is interacting with, and report what they are. However, she can successfully perform tasks like avoiding obstacles or inserting mail into slots in correct orientation, using her dorsal system. Supported by these results, J. Norman, proposes that visual perception is composed of two different systems interacting with each other. While the dorsal system responsible for the direct perception of affordances, the ventral system is concerned with high level tasks, like recognition and identification.

Studies on *mirror* and *canonical neurons* are also associated with the idea of affordances. Mirror neurons are observed to fire both while performing and observing an act. The discovery of mirror neurons supports the view that action and perception are closely related. The mirror system is used both for the execution of an action as output of the system, and for perceiving that action as an input to the system [20]. Canonical neurons [47] were observed to fire when executing an action, like grasping, as well as when observing an object that affords the same action, like a graspable object. This implies that both the motor action and the perceptual features that trigger the action are encoded in the canonical neurons [19].

Another study by Humphreys [28] showed that patients lacking the ability to identify a tool, were able to gesture the appropriate movement for using it. This suggests that an abstract representation of the tool is not necessary, and the motor actions for using the tool are directly triggered by the visual features of the tool.

The findings from neuroscience studies support that there is a strong link between perception and action in terms of neuropsychological activity.

## 2.2 Affordances and Robotics

The concept of affordances is highly related to autonomous robot control and has influenced studies in this field. While some notions of affordances are inherent in the principles of Behavior-based Robotics, the concept has been explicitly used in recent studies within Cognitive and Developmental Robotics.

### 2.2.1 Behavior-based Robotics

The concept of affordances is particularly linked with Behavior-based Robotics. The two concepts emerged in very similar ways as opposing suggestions to the dominant paradigms in their fields. J.J. Gibson's theory was based on the criticism of the representational theory of perception and cognition. Likewise, behavior-based robotics was motivated by the criticism on hierarchical control architectures. The parallelism between the outset of the two fields, holds as well for their gist against modelling and inference. This similarity was earlier noted by several researchers from the field (p. 244, [2]; [13]).

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 [3].

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 [38] suggested that robotic design can benefit from 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. [13] 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.

### 2.2.2 Cognitive and Developmental Robotics

Cognitive Robotics aims to develop robots that display cognitive abilities similar to that of human. It involves highly interdisciplinary research bringing together theories from Psychology and Cognitive Science with recent technologies in robotics. Several cognitive abilities have been studied on robots so far, including vision, perception, focus of attention, planning, reasoning and communication.

Developmental Robotics has its roots in Weng et al.'s proposal of autonomous mental development which he describes as the development of mental capabilities in an embodied system, under the control of its intrinsic developmental program, through interactions with the environment by using sensors and effectors [65]. Developmental Robotics tries to achieve this by employing theories from developmental psychology and neuroscience. This, in turn, can be viewed as employing robots as embodied models to test theories from developmental sciences [33].

The concept of affordances has recently attracted interest among researchers from the cognitive and developmental robotics community. Several studies using affordances in learning, action selection, decision-making and planning have been presented. Although the community lacks a common understanding of the concept, these studies have shown several benefits of the affordance-based approach.

A number of robotic studies focused on the learning of affordances in robots. These studies tackled two aspects of learning affordances. In the first, affordance learning is referred to as the learning of the consequences of a certain action in a given situation. The second focuses on the learning of the invariants that specify an affordance and the affective consequences of performing different actions.

Cooper and Glasspool [5] referred to the learning of action affordances, as the acquisition of environment-action pairs that result in successful execution of the action. This paper associated the affordance to the whole perceived situation of the environment and asserted the consequences of actions, rather than learning them, by judging the outcome of actions as to reinforce successful ones.

Cos-Aguilera et al. [7] used affordances in action selection by learning the relation between perceived features of objects and the consequence of performing an action on the object, where the consequence is judged by the robot in terms of the change in homeostatic variables in its motivational system. In a later study [8] they gave more emphasis to learning the "regularities" of objects and relating them to the outcome of performing an action.

Stoytchev [55, 54] studied the learning of *binding* and *tool* affordances. Learning binding affordances corresponds to discovering the behavior sequences with which a robot arm binds to a tool. Learning tool affordances, on the other hand, meant discovering the behavior that gives the desired effects when the robot arm is binded with a certain tool. Learning these affordances provides a behaviorally grounded representation of tools, in the sense that the robot knows what it can do with a tool using each behavior. However, this study does not associate the learned affordances of tools with their distinctive features, since the tools are identified with their colors.

In [18], Fitzpatrick et al. studied the learning of the rollability affordance of objects by a robot. The robot learned what it can do with an object by acting on it and observing the effects in the environment. After learning, the robot can purposively roll an object by selecting the appropriate action for the particular object. Again, in this study, the affordances of objects are not associated with their visual features, hence, it does not support the generalization of the affordance to novel objects.

## 2.3 Affordances and Planning

The ability of an agent to plan future actions, strongly relies on a representation of what it can do in the environment. This representation must describe both the conditions in which the actions of the agent are applicable and the effects that are inflicted by the application of the actions. Affordances, being action possibilities that the environment offers the agent, can be the source of information for such representations. The relation between affordances and some concepts in planning has been discussed in the literature and a number of planning studies attempted to use the concept of affordances.

The link between planning and affordances was pointed out by Greeno, saying that "affordances are preconditions for activity" [26]. Later Amant [1] defined a *simple affordance* of an action as the possibility to reach a state where its preconditions hold. The relation between affordances and planning was emphasized by Steedman who stated that a situation affords an action if it satisfies its preconditions [52].

In a later study [51], Steedman proposed to encode affordances and effects of actions in separate associative networks and use them in planning. A perceived situation triggers the retrieval of an affordance from the first associative network, which in turn triggers the retrieval of the future state for the action related to the affordace from the second network. The future state then triggers new affordances, and planning proceeds as a reactive process starting from the current state an chaining actions towards the goal.

MacDorman [35] suggested that the concept of affordances should be a base for the process of grounding representations within an agent in the interaction of its sensory-motor capabilities with the environment and proposed to use these representations in planning. He extracted invariant features of different affordance categories where, the invariant features are defined as image signatures that do not vary among the same affordance category but vary among different affordance categories. His affordance categories were defined in terms of internal indicators, such as tasty, poisonous, and were not directly related to the actions. Then, the learned categories were used for planning the sequence of actions that satisfies a motivation in terms of these indicators.

The work done in this thesis is carried out within the MACS (Multi-sensory Autonomous Cognitive Agents) project which aims to explore how the concept of affordances could affect our view to autonomous robot control. The main objective of the project is stated as "exploring and exploiting the concept of affordances for the design and implementation of autonomous mobile robots acting goal-directedly in a dynamic environment"[1]. With this objective, researchers work on both formalizing affordances and using the concept of affordances at different aspects of robotics, including behavior-based control, deliberation and learning.

As it appears in the discussions and the presented studies, the concept of affordances is strongly related with and has potential implications on planning. Particularly robot planning is concerned with issues for which the concept of affordances can provide valuable insights. The next chapter provides more detail on robot planning and its problems.

[1] http://www.macs-eu.org/

# CHAPTER 3

# ROBOT PLANNING

Planning is a central issue for robot control. A robot can be faced with a task that requires it to perform an action that is not immediately beneficial for its goal, but prepares necessary conditions for the actions that will eventually achieve the goal. This kind of behavior is difficult to obtain with a controller that reactively acts according to its goal and its immediate perception. Even though the necessity of planning has been debated within the robotics community, it has proved to be essential in achieving complex tasks and is still an area of great interest. Nowadays, robot planning has deviated from AI planning, and is concerned with the challenges related to control of robots in the real world.

Humans are capable of anticipating the consequence of their actions and constructing plans that achieve their goals. This ability, as a method of problem solving, has inspired the Artificial Intelligence (AI) community to formulate the planning problem and propose several solutions. However, as with most problems within AI, the proposed methods are based on abstractions that make them inapplicable in real-world control of robots, as well as, implausible explanations for the biological foundations of the problem. Robot planning, in this sense, is an excellent opportunity for planning to get embodied and situated, and profit more from biological inspiration, while contributing better models of human planning.

This chapter reviews the planning problem from different points of view. First, it presents the basics of classical planning and methods for dealing with planning in the real world. Afterwards, it reviews the role of planning in robot control and the relation between learning and planning. Finally, it presents some discussions about planning in biological systems and the philosophical problems in AI that are related to the planning problem.

## 3.1 Classical Planning

Planning is defined as an abstract, explicit deliberation process that chooses and organizes actions by anticipating their expected outcomes to achieve some pre-stated objectives [21]. In classical planning, problems are formulated with three components: (1) the initial state of the world, (2) the goal, and (3) the set of applicable actions [64]. All three components must be described in a common formal language. Central to this formulation, is the representation of states. The initial state is by definition a state, and the goal is often a partially specified state. Actions are described as operators that induce transitions among states.

In a classical planning domain, a state is represented as a conjunction of literals, which are propositions or first-order predicates, either grounded or free. The state representation has the *closed world assumption* which means that any condition not explicitly mentioned in a state are assumed false.

The formal description of the actions that are available to the agent is called a *domain theory*.

A common way to describe actions is to use precondition and effect lists. For an action to be applicable in a state, its *preconditions* must be satisfied by the state. When an action is applied, some of the facts no longer hold while new facts are inserted to the situation. These are held in *add-delete lists* which constitute the *effect* of the action.

In classical planning, the environment is assumed to have a number of properties that simplifies the problem. It is considered to be fully observable, deterministic, finite, static (changes are caused only by the agent) and discrete in time, actions, objects and effects [48].

The earliest approach to planning was to use state-space search. One of the first planners, STRIPS, used state-space search with means-end analysis. Planners that followed, used different approaches, either progression or regression, different search algorithms, such as A* or hill-climbing, and they proposed different heuristics.

The early planners produced totally ordered plans. They decomposed the goal into subgoals and ordered the sub-plans generated for each subgoal. This approach called linear planning is known to be incomplete [49]. For completeness, a planner must allow sub-plans to be interleaved. This led to the idea of partial-order planners, which are based on detection of conflicts between sub-plans and protection of achieved conditions [56]. This approach is also referred to as plan-space search planning. Several algorithms for partial-order planning have been developed and implemented. Planning graphs were also introduced to the field as a tool for useful heuristics for state-space and plan-space planners. They were as well used directly for planning in the GRAPHPLAN algorithm. Also, in order to benefit from the methods for solving the satisfiability problem, the idea of converting the planning problem to a satisfiability problem was proposed [30]. SATPLAN was the first algorithm to apply this idea.

## 3.2 Planning in the Real-World

The assumptions made for classical planning are often violated in the real world. States are not fully observable and deterministic. In order to be used in a real domain, representation in planning needs to be extended to handle some concepts like resource constraints, temporal measures or non-determinism [48]. Conditional planning [63] can deal with incomplete information by including the sensing action in the plan to check whether a conditions is satisfied or not. Conformant planning [25] constructs plans that do not use any sensors, but rather work in belief states. It generates a plan that achieves the goal at any possible circumstance.

Execution monitoring [16] is another method used for adapting planners to real world applications. The idea is to continuously monitor the execution of a plan in order to check whether it reaches the expected states. Re-planning might be possible in the case where the plan fails. Another option is to interleave planning and execution, such that the first action is executed before the whole plan is constructed.

*Universal plans* [50], being representatives of the reactive planning approach, compactly represents every possible classical plan that achieves the goal. The part of the universal plan that is actually executed depends on how the environment behaves at execution time. They also identify predicates that require monitoring during execution and they do not require re-planning.

These methods were developed to make planning feasible in real-world applications, mostly for robot control. The next section explains the role of planning in robot control architectures.

## 3.3 Planning in Robot Control Architectures

Planning has been one of the central issues in the study of robot control architectures. This is evident in the way that robot control paradigms have evolved. The discussion that separated the robotics community into two paradigms, in the first place, was whether planning is a necessary component of control architectures. While planning was the core of *hierarchi-*

*cal* architectures, the *reactive* paradigm opposed to all sorts of world modelling and hence dropped the planning module from the control architectures. As with most problems, robot control architectures evolved within two contradicting paradigms, ending up with better solutions somewhere in between. Modern control architectures combine the aspects of hierarchical and reactive architectures, for which they are called *hybrid* or *deliberative/reactive* architectures. These architectures brought back the planning module, but kept the tight coupling of sensors and actuators.

## 3.4 Planning and Learning

The major weakness of robotic systems using symbolic planning is that the plan operators are hard-coded into the system. As Pissokas states in [45], this brings several limitations. Firstly, mapping actions to motor signals manually may results in leaving out some parts behavioral space available for manipulation. Similarly, only part of the perceptual space is available in the planning space, due to the difference of perception and experience between a robot and a human who decides how sensory input will be mapped to internal symbols. Lastly, due to noises in the sensors and actuators, actions may eventually differ from what they are designed to do.

Using machine learning techniques to improve planning and plan execution has recently become an important challenge within the planning research. Although this includes a wide range of topics, from learning situation-plan associations to heuristic optimization, what concerns robot planning is the learning of the domain. In robotics, this corresponds to learning representations of the robot's capabilities and the characteristics of the environment in which it operates. Most of the plan-based controlled robots in the literature use domain knowledge provided by an expert. Proper execution of the robot hence depends on how well the programmer can think of every detail and accurately analyze the properties of the environment.

McDermott, suggest that the most plausible practical application of learning within robot planning is the learning of statistics of the domain, which he defines as "what to do under what circumstances or what is likely to happen under what circumstances" [37]. He further explains:

> "...what gets learned is likely to be a mapping from low-level perceptions to recommended low-level actions or to the predictions of future low-level perceptions."

This accurately describes the learning problem considered in this thesis.

Sub-symbolic planning [45] is a good example of approaches for learning domain statistics. It proposes to obtain associations between an action and the environments before and after executing the action, within the behavioral and perceptual space of the robot. In [39], neural networks are trained to learn these associations which are later used for predicting the perceptual change that the execution of an action will cause. In this way, the robot can select the appropriate actions to achieve a goal.

Perceptual anchoring [6] is another way in which learning techniques improve robot planning. Even if the domain knowledge is hard-coded into the robot using a symbol system which is manipulated by the planner, mapping the symbols to the sensor readings of the robot remains a hard problem. Learning these mappings instead of manually tuning them, improves the accuracy of the mappings while facilitating the process.

## 3.5 Biological Foundations of Planning

Planning is one of the core abilities possessed by human. Everyday we make plans at different levels of complexity and abstraction and different time-scales. For example, we plan how we will make a living after graduation, we plan how we will accomplish our daily responsibilities and we plan how we will cook our dinner. Although there has been a number of studies attempting to discover the mechanism that underlies this ability of human, there is no complete theory that explains it.

Although several forms of planning are known to exist, two general types of planning, forward search and backward search (means-end analysis), are considered to subsume most forms the of planning either singly or in combination [15]. Forward search is a problem solving technique that starts from the current state and progresses by chaining actions until the goal is satisfied. Means-end analysis is the method of starting at the goal state and reducing the difference between the goal and the start state [42]. The two methods are also referred to as forward vs. backward chaining or progression vs. regression in different contexts.

Humans are not the only ones equipped with the skill to plan ahead. Several species of animals also display behaviors that cannot be explained merely with instinct driven reactive behavior. One of them is our close relatives, chimpanzees. Most of the evidence on the planning capability of chimpanzees was revealed by Koehler's well-known work on chimpanzee cognition, in his book "Mentality of Apes" [32]. Sultan, one of Koehler's chimpanzees, was able to find several solutions to obtain a banana hanging at a height out of its reach. He found a box in the room, moved it beneath the banana, and jumped up from the top of the box to get the banana. In another case, Sultan would stack the boxes one on another to reach the banana. He would also combine two short sticks by inserting the narrow one into the thicker one to obtain a long stick that he uses as a tool to get the banana.

This shows that chimps can take actions that, when considered in isolation have no relation to the goal, like grasping a stick that lies on the ground when the goal is to eat a banana, but when considered together with the sequence of actions that follow, is the key to achieving the goal. This is what Koehler calls "the making of implements" and is strong evidence that primates posses the same mechanism that in human has evolved to generate from very simple to more elaborate plans.

Based on one of Koehler's experiments, Steedman suggests that planning in animals should be in the form of forward chaining. He explains this with the fact that an animal's access to the affordances is limited with the perception of the immediate environment and that affordances are indexed by *objects* rather than end-states [52]. Planning is thus likely to be based on the perceived objects and proceed reactively from the current situation towards a goal.

On the other hand human plan construction is generally based on goal-regression planning. To achieve a goal, we consider an action that would achieve it under some specified circumstances, and then try to find a way of putting ourselves in those circumstances in order to achieve the goal by performing the action [46]. Putting ourselves in those circumstances becomes a subgoal. The basic idea of goal-regression is to work backwards from the goal through subgoals until we find a subgoal that we are already in.

Backward search seems to be a more complex ability as suggested by the course of evolution and strongly related with the symbolic representation capacity brought by linguistic tools. Indeed, it has a major adventage over forward chaining in that it allows us to consider only actions relevant for the goal [48]. Nevertheless, human planning, in most cases, is much more complex than simple goal-regression. For instance, it is often hierarchical rather than sequential. Likewise, animal planning is observed to be goal oriented, rather than simple random progression. If it was totally reactive, the probability that forward chaining would end up in the goal state would be very small. However goal achievement, as in chimpanzees, does not seem to be accidental. This implies the use of heuristics directing the animal to the goal.

## 3.6 Philosophical Problems Related to Planning

Robot planning inherits several problems from Artificial intelligence. These problems have been discussed, mostly from their philosophical aspects, for half a century. A review of these problems is essential as they are part of the problem tackled in this thesis.

**Symbol Grounding Problem** Planning assumes the existence of symbols which provide abstractions over the world, however a robot's successful execution of a plan requires its symbols to be meaningful in terms of its interaction with the world. This is closely related with the symbol grounding problem, which was formulated by Harnad in 1990, with the following questions [27]:

> "How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads? How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?"

As clarified in [58] the symbol grounding problem deals with specifying how an artificial agent can develop a semantics for its symbols by interacting with its environment and achieve this "autonomously" and "from scratch". It has also been referred as the problem of "embodying symbols" [41, 34].

Although a number of solutions have been proposed for the symbol grounding problem, it turns out that none of them satisfies the *zero semantic commitment condition* [58] which requires that no semantic resources be innately or externally provided to the system. It only allows the agent to have some capabilities and resources so that it can ground its symbols.

The symbol grounding problem is often confused to be the problem of mapping internal symbols to objects in the environment, as they appear in the perceptual array. This problem may assume the existence of symbols that are extrinsic to the agent, and is a sub-problem of the symbol grounding problem, in the way Harnad meant it. This sub-problem has indeed been given different names such as *perceptual anchoring* [6].

**Qualification Problem** The impossibility of representing all the preconditions for the successful performance of an action, so that it has its intended effect is referred to as the qualification problem [36]. The qualification problem is especially concerned with conditions that can prevent the successful execution of the action, rather than what is required for it. One can specify all the necessary preconditions that makes the action applicable, while there's no limit to situations that can prevent the action from being successful. Consider the example of a monkey grasping a banana. The grasp can successfully be accomplished in the presence of a banana within the arm-reach of the monkey with the condition that its hand is free. However one could come up with precondition statements like "the action will be successful unless a bear comes and grabs the banana first". As McCarthy puts it, its a matter of "taking into account".

**Ramification Problem** Ramification problem is related to specifying indirect consequences of an action. It deals with the representation of what happens implicitly when an action is executed. For instance, consider a monkey jumping down from the top of a box with a banana in its hand. The consequence of jumping in this example is not only that the monkey gets down, but also that the banana gets down.

**Frame Problem** The problem of expressing a dynamic domain, without explicitly specifying the conditions that are not affected by the execution of an action is called the representational frame problem [48]. A similar problem in psychology, the inferential frame problem, is described as the problem of limiting the beliefs to be updated in response to actions [48]. One can use frame axioms to express that a fact persists unless an action changes it. The frame problem arises from the fact that every literal in the state representation requires such a frame axiom and as the state representation grow, this becomes intractable. STRIPS planning deals with the frame problem by using an add-delete lists for each action, separate from its precondition list. All facts that are not mentioned in the add-delete list are assumed to persist when the action is applied.

One can list several other problems concerning robot planning, most of which are related with the problems reviewed above, dominantly with the symbol grounding problem. The common source of these problems is the represention of the world within an agent.To date, methods used in representing the world knowledge of an artificial agent, have not provided general and extendible frameworks for dealing with such problems. Furthermore, world knowledge is often built into the agents while it seems crucial that it must be acquired by the agent based on its own experiences.

The concept of affordances provides insights into how an agent perceives and represents the world. However, building artificial agents based on this concept, requires it to be formalized. The available affordance formalizations do not provide a complete framework that makes the concept adoptable in robot control, learning and planning. The next chapter presents a new formalization of affordances that claims to establish such a framework. The proposed formalism makes it clear how an agent must acquire affordance representations and how such representations can be used in planning.

# CHAPTER 4

# FORMALIZING AFFORDANCES FOR ROBOTICS

The concept of affordances has been referred in several research fields, including robotics and planning, however, their conception of the concept are different and inconsistent. Making use of a psychological concept, especially in such research field where everything is to be reduced to a computer program, a concrete definition and a formalization of the concept is essential.

There have been several attempts to define and formalize affordances, mostly within the Ecological Psychology school. These provide a good discussion for understanding the concept and interpreting J.J. Gibson's writings. However, from the pragmatic viewpoint of robotics, none of them provide a coherent framework that can be used in representing affordances within an agent and building mechanisms to acquire such representations.

The recent formalization proposed by Şahin et al. [10] tries to establish such a framework. This chapter first reviews the prior affordance formalisms within Ecological psychology and then describes the new formalization. The last section elicits the link between this formalization and the formal descriptions of the planning problem.

## 4.1 Prior Formalizations of Affordances

Turvey [59] defines affordances as dispositions, i.e. properties that present potentials, of the environment. In his formalism, the complementarity of the animal and the environment, is reflected in the dual nature of dispositions, in the sense that dispositions exist in pairs and get actualized when they combine with their compelement dispositions. In this vein, he proposes the effectivities, which are dispositions of the animal, to be the complement of affordances, which are dispositions of the environment. The interaction of the animal with its environment thus corresponds to the actualization of the two dispositions.

Stoffregen criticized Turvey's formalism for attributing affordances to the environment alone. Instead he defines affordances as emergent properties of the animal-environment system. He thinks they are emergent in the sense that they are not inherent in neither the animal nor the environment alone [53].

Chemero's formalization is in agreement with Stoffragen's, but it refines the "animal" aspect of the formalism to "abilities of the animal", and similarly, the "environment" aspect to "features of the environment" [4]. He defines affordances as relations between the two, emphasizing that they are relations rather than properties.

One of the most controversial discussions on affordances, as reviewed above, is whether they reside in the environment, the animal or in the system that accounts for the interaction of the two. Şahin et al. advocate the existence of three perspectives to view affordances, namely *agent*, *environmental*, and *observer* perspectives, and suggest that the seemingly contradictory positions of different authors about where to place affordances arise from the fact that they pose their arguments from different perspectives [10].

Prior formalizations of affordances provide a good framework for understanding affordances, by questioning where the affordances reside, whether they are properties or relations, and whether they can be internally represented. Yet none of them provides a complete semantics which can be used in constructing artificial agents that perceive affordances and act accordingly in the way animals do. Furthermore, these formalisms lack to explain how affordances are learned by interaction with the environment and explain how they fit into E.J. Gibson's theory.

## 4.2 A New Formalization of Affordances

The formalism proposed by Şahin et al. builds on Chemero's argument that affordances are relations within the agent-environment system. It proposes that affordances are general relations that are derived from interactions with the environment. One interaction is called an *affordance relation instance*. In the formalism, a relation instance has the form (*effect*, (*entity*, *behavior*)), meaning that a certain *effect* is generated when the *behavior* is applied on the *entity* by the agent.

The term *entity* denotes the environmental aspect of the relation. It can be thought to correspond to *features* or *object* as used in different contexts. It represents the state of the environment as well as the state of the agent, as perceived by the agent. The *behavior* represents the physical embodiment of the interaction of the agent with the environment. The *effect* is defined as the perceivable change in the environment, or in the state of the agent, as a result of this interaction.

An affordance relation instance does not constitute a relation by itself. Affordances should be general relations with predictive abilities over future interactions. The formalism suggest that such relations are acquired from a number of relation instances, through the formation of *equivalence classes.* Based on this, a formal definition is given as follows, where the brackets < .. > denote equivalence classes.

> An affordance is an acquired relation between a certain <*effect*> and a certain <*(entity, behavior)*> tupple such that when the agent applies a *(entity, behavior)* within <*(entity, behavior)*>, an *effect* within <*effect*> is generated. The relation can be represented as:
>
> (<*effect*>, <*( entity, behavior)*>).

In the definition <*effect*> denotes an effect equivalence class, whereas <*(entity, behavior)*> can be one of the other three types of equivalences classes, namely entity, behavior and affordance equivalence classes. There is a crucial difference between effect equivalence and the other equivalence types. The effects of two relation instances are considered to be equivalent if they both achieve the same goal. On the other hand, the *(entity, behavior)* pairs of two relation instances are considered to be equivalent if their effects are equivalent. This equivalence, corresponds to the *entity equivalence* of two different entities if the behavior is the same in both interactions, whereas it refers to the *behavior equivalence* of two different behaviors if the entity is the same. If both are different this is called an *affordance equivalence.* Hence, the three types of equivalences rely on the equivalence of effects, while the equivalence of effects is determined by the goals of the agent. The close relation between the effect and the goal, is the basic reason for making the effect aspect of affordances explicit and distinguishing it from the other aspects of affordances.

Before describing equivalences in more detail and presenting some examples, it should be noted that the idea of explicit inclusion of effects in the affordance representation was set forth within the MACS project and in [11, 29] the learning of affordances was proposed as the learning of bilateral relations between three components of the representation.

**Entity Equivalence** The class of *entities* which support the generation of the same *effect* upon the application of a certain *behavior* is called an *entity equivalence class.* Consider the example depicted in Figure Figure 4.1. A monkey is trying to obtain a banana that hangs high out of its reach by poking it with a tool. The monkey uses either a steel stick or a broken tree branch. In both cases the behavior pattern is the same, the monkey moves the tool above its head. The obtained effect is the same as well, since in both cases the tool hits the banana. The two different tools in this case, are said to have entity equivalence.

The notion of entity equivalence emphasizes the relevant attributes and invariant properties of entities that have the same affordances. In the given example both tools have one dimension that has a length approximately equal to the distance from the monkey's hand to the banana. Also they both fit to the monkey's hand on one end. Furthermore, they are both rigid and both not too heavy such that the monkey can lift them. These are some properties that make these tools suitable for achieving the desired effect, with a particular behavior. The entity equivalence class formed by these instances, should therefore be described with these properties and the learning process should account for differentiating these invariants. Notice that the two tools have very different appearances, are made of different materials and may have different colors, which are all irrelevant to the affordance.

Entity equivalence also relates to the prediction ability over novel situations. It provides general entity descriptions based on which one can use novel entities to achieve their goals. For instance, the monkey in the example would be able to use a golf club for the same task, though it had never seen one before, and doesn't know anything about golf clubs.

**Behavior Equivalence** The idea behind entity equivalence can also be applied to the behavior aspect of the formalism. Keeping the entity constant, one can achieve the same effect by performing different behavior patterns. These behaviors are said to be behaviorally equivalent. For example consider a monkey that tries to grab a banana that someone offers. It could grasp the banana either with its left hand or right hand, both of which would achieve the desired effect. One could say that the behavior that achieves the goal of taking an offered banana, is a grasp with either hand. This constitutes the description of a behavior equivalence class.

**Affordance Equivalence** The next step in the discussion of equivalence classes accounts for cases in which the same effect is obtained with different entity-behavior pairs. Such pairs form affordance equivalence classes. Consider the two cases where a monkey climbs onto a stick or steps over a stack of box to reach a banana. The behavior pattern involved in the two cases are very different as well as the means used for climbing up. However in both cases, the effect of being elevated is achieved.

**Effect Equivalence** The three types of equivalences described above strongly rely on the notion of effect equivalence which is somewhat different from the three. Two effects are considered equivalent if they achieve the same goal. This implies that part of the effects generated by the application of a behavior are irrelevant for the agent's goal, which in the context of the goal can be considered as side-effects. Similarly, two effects can be different in magnitudes, still equivalent if they both achieve the same goal.

Consider the effect equivalence that was implicit in the examples given above. In the first example, when the monkey moves the tool above its head, the effects that it visually perceives are not quite the same in the two cases, since the appearance of the tools are different. However in both cases the effect of hitting the banana is achieved, and therefore the effects are equivalent. In the second example, when the monkey grabs the banana with different hands it ends up with the banana in either hand which are indeed different effects. Yet again, they are equivalent effects in terms of the goal of having the banana. Similarly, in the third example, the two very different effects are equivalent since they both satisfy the goal of being elevated.

As mentioned earlier, the effect aspect of the formalism is strongly related with the intentions or goals of the agent. The perception of affordances is modulated by intentions. For instance, if we consider our capabilities as humans, even the simplest environment offers an infinite number of affordances. Yet we only perceive the ones we are interested in, the ones that relate to our intentions. The distinction of effects becomes important because we express our intentions or goals in terms of the effects that we are able to create in the environment.

Making the effect explicit in the formalization of affordances, turns out to make the relation between planning and affordances more evident. There is, in fact, a strong similarity between the proposed formalism and the formal descriptions of actions used in planning. The next section clarifies this correspondence.

## 4.3 Affordance Relations as Plan Operators

As detailed in Chapter3, classical planning systems work with operators which consist of three main components: *pre-condition*, *action*, and *effect* denoting the initial requirements for the action to be applied, the atomic action to be taken, and the expected changes to be inflicted in the environment, respectively. The planner uses the operators, which are assumed to be pre-coded, to generate a sequence of operators, such that its application would take

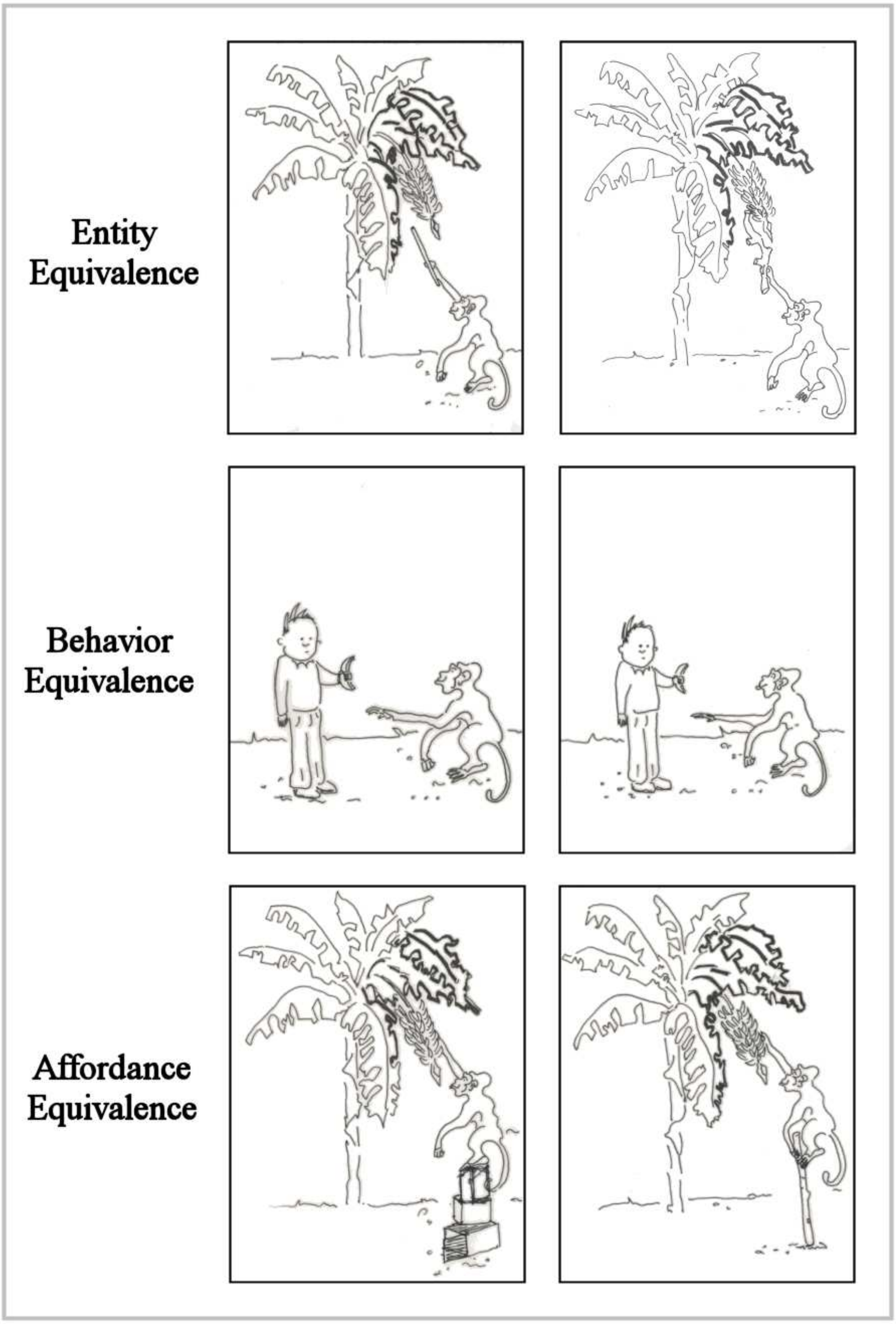


Figure 4.1: Entity, behavior and affordance equivalence. (Drawings courtesy of Münire Sağat)

the system from a given initial state to a desired goal state.

It can be argued that the affordance formalism proposed in [10] creates relations that can also be used as operators for planning. An affordance relation is indexed by its *effect* and include tupples which store how that particular effect can be achieved. The $<$*entity*$>$ and $<$*behavior*$>$ components in the proposed formalism, can be considered to correspond to the pre-condition and action components in the STRIPS representation. A major difference between the STRIPS representation and the affordance representation is the way the operators are indexed. In STRIPS, operators are indexed by their actions, whereas affordance-based operators are indexed by their effects. For instance, the proposed formalism implies that the traversability affordance can be represented as a planning operator:

```
(index: get-banana
 effect: have-banana
            (entity: stick, behavior: poke)
            (entity: box, behavior: climb-on)
)
```

whereas, the same relations could represented using two different operators in STRIPS as:

```
(index: poke
 action: poke
            pre-condition: stick,  effect: have-banana
)
(index: climb-on
 action: climb-on
            pre-condition: box, effect: have-banana
)
```

For example an affordance-based agent may represent the "traversability" affordance which would be actualized by either a "swim" or "walk" behavior. A STRIPS database, on the other hand would contain the operators indexed as "swim" and "walk". In this respect, a more appropriate example of planning representations resembling our affordance representation would be the Reactive Action Packages (RAPs) [17]. Just like affordances, RAPs are grouped according to their desired effects.

The different representations of operators have important implications for planning. In STRIPS, the whole environment is assumed to be perceived before the planner can start planning, a plan effectively consist of a sequence of actions (since operators are indexed by their actions), and that any change in the environment during execution, may require the plan be revised by the planner. These are important limitations, which can be addressed by the operator structure implied by the formalism. Not surprisingly however, these limitations were discussed and addressed by some of the relatively more recent work in robotics, such as RAPs.

A RAP may have more than one method to perform the same task. The method whose context is satisfied at the time of execution is chosen and its task network is executed. Each method of a RAP is applicable in a different context and its task-network consist of different actions, however all methods are supposed to create the same effect and hence they have one common success criteria. The success test typically contains an algorithm which judges whether the application of a method was successful or not. This is very similar to the proposed formalization where affordances that have the same effect are considered equivalent. The methods of a RAP are similar to different entity-behavior pairs that have the same effect, where the context of the method corresponds to the entity and the task-network corresponds to the behavior. The task network denotes a partial plan which may use one or more RAPs, whereas the context specifies the situation that the method is applicable, similar to the pre-condition component of STRIPS. Note that all methods are subject to the same success test.

The traversability relation can be represented as a RAP as follows:

```
(index:  get-banana)
       (success-test: have-banana? )
               (context: stick, task-network: poke)
               (context: box, task-network: climb-on)
)
```

The proposed affordance formalism also shows similarities with the represention of planning problems in situation calculus. In the simplest version of situation calculus, an action is described by two axioms: a "possibility axiom" that says when it is possible to execute the action and an "effect axiom" that says what happens when a possible action is executed [48]. This description of an action strongly resembles the representation of an affordances in the formalism. The possibility axiom corresponds to the description of the required "entity" for the action and the effect axiom corresponds to the "effect" of created by the action.

Within our formalism, a behavior itself may be a sequence of more primitive actions.

That is to say that a sequence of actions, which was at first derived through planning, may be stored by the agent as a new behavior with its own entity and effect. Constructing a plan for the first time requires some mental processing time and power, however after having frequently used it, the plan can be directly retrieved and it is associated with the entities initial to execution of the plan and the effect of executing the whole plan, ignoring the entities appearing throughout the execution or the effects of individual behaviors. This implies a dual relationship between planning and affordances. Not only do affordances provide a good framework to ground the representations used in planning, but also, planning can lead to the inference of new affordances in the environment.

This chapter has presented the conceptual framework on which the work done in this thesis is based. In the rest of the thesis the application of the ideas implied by the formalism is presented. The next chapter gives the experimental framework used in this study. The following chapters present the implementation and results of learning affordances and using affordances in planning.

# CHAPTER 5

# EXPERIMENTAL FRAMEWORK

The experiments in this study were conducted on the KURT3D robot platform and its simulator MACSim. This chapter first provides some information about the robot and its simulator, focusing on the components that were used in the study. Then, it describes the perceptual framework and the atomic actions of the robot.

## 5.1 The Robot: KURT3D

KURT3D is a medium-sized, differential drive mobile robot which was originally developed for sewerage inspection at Fraunhofer-IAIS [1]. It has dimensions of 45 *cm* (length) x 33 *cm* (width) x 47 *cm* (height) and it weights 22.6 *kg*. The robot's locomotion system is composed of six identical wheels, connected by a toothed belt on each side of the robot, resulting in a differential drive system. Dead-reckoning is made possible by the encoders on two sides. Kurt3D is equipped with a number of additional sensor modalities, including two pan-tilt color cameras, eight infrared proximity sensors, and two tilt sensors.

The fundamental sensor of KURT3D is its 3D range scanner, which is based on a SICK LMS 200 2D laser scanner, rotated vertically with an RC-servo motor. The 2D scanner uses the time-of-flight method to measure distance. It has a horizontal range of 180° and a maximum resolution of 0.25°. Using the pitch mechanism has a step resolution 0.23° and can sweep a vertical range of approximately 180°. Although a full scan takes a relatively long time of 45 *seconds*, the scanner has been used at maximum resolution to prevent noise-related failures. The time issue can be overcome by using state-of-the-art realtime range cameras and has no implication on the results obtained in this study.

To increase the manipulation capability of the robot a crane with three degrees of freedom

[1] Fraunhofer Institut für Autonome Intelligente Systeme

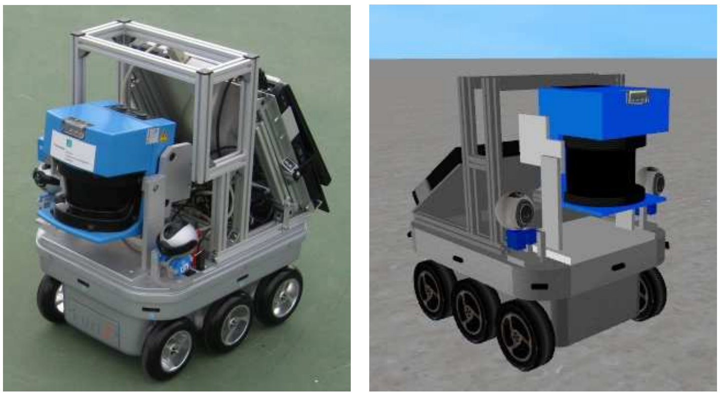

Figure 5.1: The KURT3D robot without the crane and its simulated model.

has been designed. It consists of an arm with a magnet is hanging. The arm originates from the top of the robot and heading of the arm can be changed with the help of a motor. With two other motors the magnet is slided along the arm to make the magnet move radially away and towards the robot, and the rope holding the magnet is released or pulled to make the magnet move up and down.

## 5.2 The Simulator: MACSim

MACSim is a high fidelity physics-based simulation environment that models the Kurt3D robotic platform and its environment. It has been developed at the Kovan Research Lab, within the MACS project. MACSim is built on top of a commercial quality open-source physics engine, ODE[2] (Open Dynamics Engine) and KODEX (Kovan ODE extensions) [44], which extends the capabilities of ODE in several aspects.

The major strength of MACSim is its accuracy in modelling the characteristics of the real robot. Several studies have shown that robot control programs developed in MACSim can successfully be ported to the real KURT3D [60, 12]. This is achieved with precise calibration of several parameters and realistic modelling of the sensors.

The sensor and actuator models in the simulator are based on samples taken from the real robot and are validated with several cases. Physical parameters of the simulated robot are faithful to the real robot. This includes properties of the robot, such as mass, size, and center of mass.

The laser range scanner of the robot, which is the most critical sensor for this study, is modeled using the ray tracing method. Similar to the actual working principle of the

[2] `http://ode.org`

scanner, a ray is rotated step by step, where the step interval is determined by the resolution of the scan, either 1°, 0.5° a 0.25°. The distance to the first intersection of a ray, is directly returned as the sensed value, without any processing, since the actual laser scanner has a measurement resolution of $+/-$ 15 $mm$ within 8 $m$ range.

Some experiments were performed in order to compare the scanner model against the real one. Firstly, the robot was placed at a 1m distance from a large wall and several scans were performed, for different vertical and horizontal apex angles at three different resolutions. In each case it was observed that local errors stayed within the range $-1$ to 1 $cm$. A Gaussian distributed noise with a zero mean and 0.25 $cm$ of standard deviation was also added to each distance measurement to make the scanner model more realistic.

The accuracy of range images also relies on the modeling of the servo motor that rotates the range scanner. The resolution of the servo motors were determined empirically, by counting the number of steps taken to complete a full angle and adjusted accordingly.

The modeled laser scanner has been validated with several cases in which different simple shaped objects were scanned. The snapshots from one of these experiments are given in Figure Figure 5.2 and the scan data obtained from the simulator and the real scanner are given in Figure Figure 5.3. It can be observed that although there are differences in the environment, for instance at the edge between the wall and the floor, the scan output corresponding to the carefully modeled cylindrical object in the simulator matches quite well with the real scanner's output.

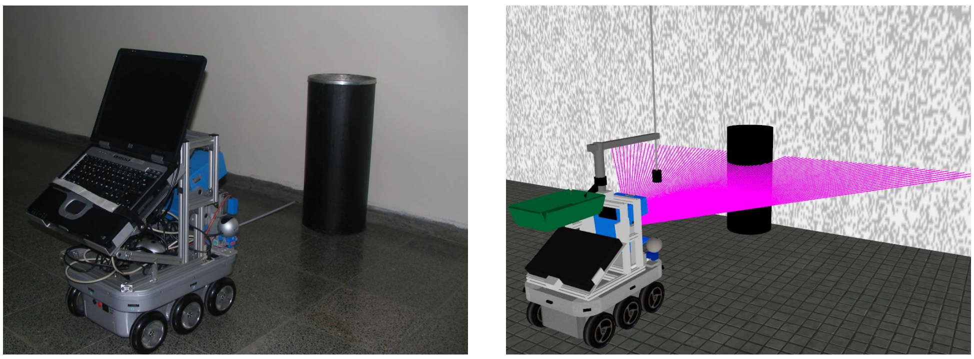

Figure 5.2: The real and simulated robot cased against a cylindrical object are making a 3D scan.

To make the motion of the robot in the simulator similar to reality, parameters like

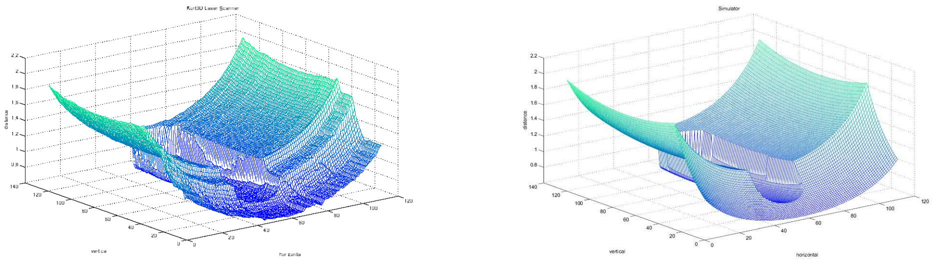

Figure 5.3: Real and simulated 3D range data obtained from the cylindrical object scan.

motor torques, friction coefficients between every moving body, moments of inertia need to be adjusted. However, trying to calculate these values with some general experiments is often not enough to eventually obtain the motion of the robot in real world. The conventional method to adjust the motion of a robot in the simulator is to estimate these parameters roughly and tune them to adjust the motion to be as close as possible to reality. MACSim has been calibrated with this method such that motor parameters employed in the simulator produce a proper motion on the real robot.

The simulator also provides encoder readings similar to that of the robot. The encoder values are obtained from the actual motor speeds in the simulator.

The crane has been modelled according to the supplied design. In this study the heading of the crane arm and the radial distance of the magnet are kept constant. The arm always faces forward and can lift objects that are about $10cm$ away from the front of the robot. The liftability of objects depends on their magnetizability property as well as the shape of the upper surface of the object. Only objects with a flat upper surface can be lifted. Weight of the object is also a criterion however in this study no heavy objects were used. The attachment of suitable objects to the magnet was modelled by forming a fixed joint between the object and the magnet, when the magnet is turned on.

## 5.3 The Environment

In this study KURT3D interacts with an environment including one of the two types of simple objects: rectangular boxes ( ) and spherical objects ( ). When contacted by the robot, spheres roll away while boxes block the robot's motion. Boxes can be lifted with the crane if they are positioned under the magnet whereas spheres cannot be lifted since they do not have a horizontal surface to attach.

## 5.4 Perception

The robot perceives its environment primarily 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. The feature set also contains two features obtained from its encoders and one feature obtained from the crane rope tension. To obtain the scanner features, the range image is down-scaled to reduce the noise and split into uniform grids. For each grid, a number of distance and shape related features are extracted. The distance related features are the closest, furthest, and mean distances within the grid. The shape related features are computed from the normal vectors in the grid. The direction of each normal vector is represented using two angle channels $\varphi$ and $\theta$, in latitude and longitude respectively and two angular histograms are computed. The frequency values of these histograms are used as the shape related features. The scanner features used in this study were proposed in [60].

## 5.5 Actions

The manipulation capability of the robot is restricted with five atomic actions. Tree of these actions use the wheels of the robot while the other two use its crane as actuators. The motion-related atomic actions are *move-forward*, *turn-left* and *turn-right*. The *move-forward* action drives the robot straight ahead and places the robot $40cm$ away from its initial position, if the move is not obstructed by any obstacles. The *turn-left*, and *turn-right* actions turn the robot in place for $45°$. The atomic actions utilizing the crane are *lift* and *drop*. The *lift* action lowers the rope until it touches something, turns on the magnet, and raises the rope back to its original position. The *drop* action lowers the rope, turns off the magnet, and raises the rope back. In both action, the limit to which the rope can be lowered is detected by the sudden decrease in the rope tension, when the magnet, together with an object whenever it is initially loaded, touches the ground or an object placed under the crane magnet.

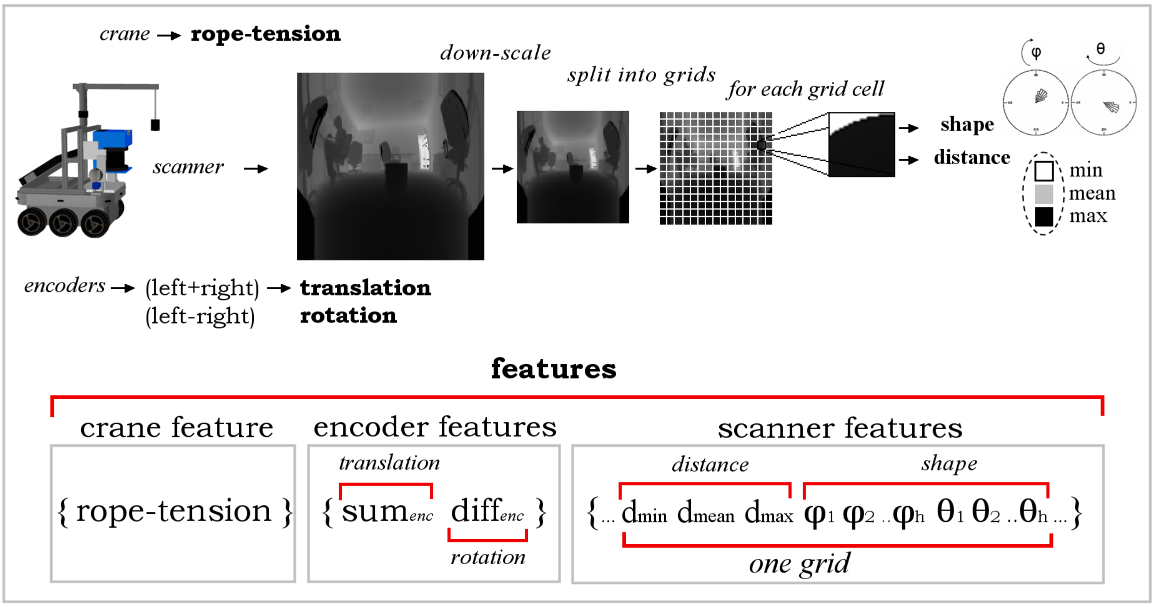


Figure 5.4: Perception of the robot. The robot perceives its environment as an array of features. Sensory data to extract these features comes from various sensors. A pressure sensor on the crane arm measures the tension on the crane rope, giving an estimation of the weight lifted by the crane magnet. This value is directly used as a feature. Encoders on both sides of the robot, give counter values, which indicate wheel positions in comparison with an initial position. These readings are converted to features that relate to the translation and rotation of the robot, by taking the sum and difference of the two readings from left and right encoders. The range image taken by the scanner, is first downscaled and slit into grids. Then, a set of features corresponding to distance and shape characteristics of each grid is extracted.

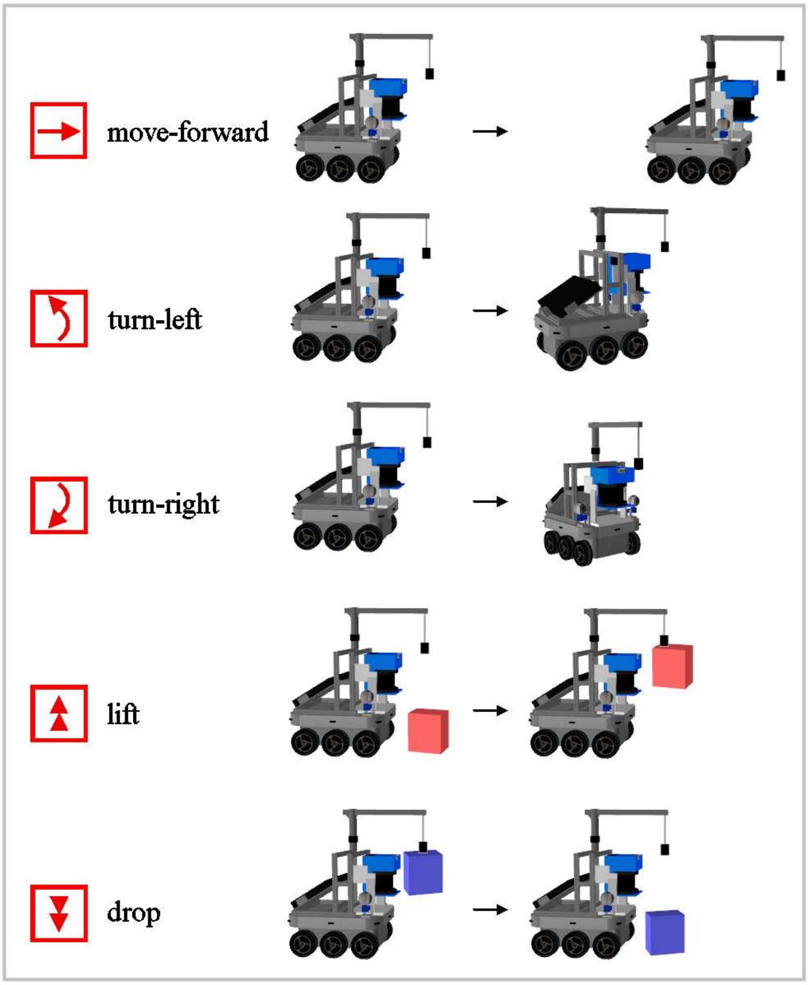


Figure 5.5: Atomic actions of the robot. The robot is endowed with five pre-coded actions: *move-forward, turn-left, turn-right, lift and drop.*

# CHAPTER 6

# LEARNING AFFORDANCES

The affordance formalization described in Chapter 4 proposes that learning of affordances occurs by the differentiation of equivalence classes. In this chapter an implementation of this learning scheme is described.

Development of the robot starts with the exploration of unintentional reflexive actions in an environment. It first executes these actions randomly, and discovers the changes it can consistently create in the environment. As a part of its development it should also associate these changes with the situations in which the actions are executed. After learning these associations the robot should be able to use its actions purposefully, to achieve a goal. The stage of discovering the changes it can create, corresponds to forming *effect equivalence classes* in the formalization. Associating these changes with behaviors, and the necessary situations, corresponds to linking *effect equivalence classes* with *entity equivalence classes* and *behavior equivalence classes.*

The learning of affordance relations consists of two phases: an *interaction phase*, in which relation instances are collected, and a *learning phase*, in which affordance relations are formed from the collected instances.

In the first phase, the robot collects *affordance relation instances* by executing its primitive behaviors one at a time, in different environment. It perceives and records the environment before executing a action, and after executing it. In the second phase it derives general *affordance relations*, using the set of collected instances. This requires forming *entity equivalence classes* and *effect equivalence classes* for each action, and connecting them in an *affordance relation.*

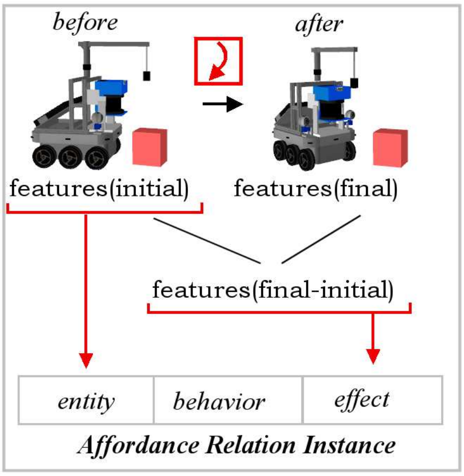


Figure 6.1: Representation of *entities* and *effects*. Within a single interaction with the environment, the robot perceives the environment, performs an atomic action and perceives the environment again. The feature array perceived before the execution of an action constitutes an *entity*. The vectoral difference between the initial and final feature arrays is used to represent the *effect*. The *entity* and *effect* vectors together with the action that created the interaction, forms a triple, that corresponds to an *affordance relation instance*.

## 6.1 Interaction

In the interaction phase the robot is put in different situations where it performs each of its actions. The interaction environments contains a single object which differs in shape, size, position and orientation. In this phase the robot interacts only with spheres and box shaped objects. The object is placed in front of the robot, within an square area of 1.2 $m \times 1.2$ $m$.

In each setting, the robot performs all of its actions once. It extracts the perceptual features both before executing the action and after executing it. The feature array extracted before the execution of the action consist the entity of the interaction instance. The vector difference between the features extracted before and after execution of the action corresponds to the effect of the action. The entity and the effect associated with each action is stored as an affordance relation instance (Figure Figure 6.1). In this study, a total of 1000 relation instances are collected for each action.

Note that the interaction phase is merely a means of collecting samples. No adaptation or on-line learning occurs throughout this phase. An on line learning method for a similar framework was proposed by Ugur et al. [61].

## 6.2 Learning

The aim of the learning phase is to derive *affordance relations* from the set of *relation instances* collected in the interaction phase, through the formation of equivalence classes. This phase also includes the differentiation of perceptual features that are relevant to an action, in order to achieve perceptual economy. This includes the detection of relevant features within both the entity and the effect arrays.

The first step in the learning phase is to detect relevant features of the effect array, which are the features on which the action can assert a certain change. Next, similar effects within the set of *relation instances* of an action, are grouped together to get more general description of different kinds of effects that the action can create. This is achieved through the unsupervised clustering of the effect instances. This, in a sense, 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 has two aspects. Firstly, the robot selects the features describing the *entity* which are *distinctive* 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 *invariant* properties of the entities that result in the same effect upon 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 the four steps in the learning phase. We also present the results of applying these steps on the data collected in the interaction phase.

### 6.2.1 Selecting Relevant Effect Features

Each action, when executed in different environments, can create changes only in some parts of the robot's perceptual field. After a certain learning phase, the robot should expect to perceive changes only in a certain subset of its perceptual array. Paying attention to only this subset, the robot can achieve perceptual economy during the perception of the effects generated by its actions. The notion of "perceptual economy" here, should not be confused with the economy in perceiving affordances, that J.J. Gibson suggested. Nevertheless, learning the relevant effect features brings about a certain attention mechanism and reduces the cost of extracting features for unchanged sensory input.

Perceptual development of human also includes the differentiation of the perceptual channels that are relevant for perceiving the effect created by own actions. We know where to expect to perceive a change when we perform an action. For instance, when we move our left arm we anticipate a change in the left side of our visual field. Similarly, if we turn up the heater we would expect to feel warmer, but not ever would we expect an increase in the ambient light. In a similar way, the proposed mechanism, will discover the features that are most likely to change as a consequence of an action of the robot.

The relevant effect features are selected over the whole set of interaction samples. In a single instance, an effect feature is taken to have considerably changed if it has an absolute change value larger than 2% of the maximum possible change. This means that very small changes are neglected. Then, the features that change more than the threshold in each sample, are considered over the complete set of samples. Features that have changed within at least 2% of the complete set of interaction instances are considered relevant for the action.

This selection, is performed for each action separately. The relevant grids for the five atomic actions of the robot, detected as described above, are presented in Figure Figure 6.2. The ratio of selected features to the total number of features in the effect array are respectively, 10.02%, 9.70%, 9.47%, 10.86% and 14.15%, for the actions *move-forward*, *turn-left*, *turn-right*, *lift* and *drop*.

The robot profits from the differentiation of relevant effect features with a perceptual economy, when the effect that is actually created by an action is to be monitored. The selected features are also important in perceiving the entity, since they identify the features that will change after the execution of the action. In order to estimate future values of the changed features, the robot should perceive the initial values of these features.

### 6.2.2 Forming Effect Equivalence Classes with Clustering

An action may have different effects depending on the environment in which it is executed. Yet, some of the effects will be similar to each other. Although the meaning of "similar" is rather context dependent, we can define a constant measure of similarity that applies for most contexts. Using this similarity measure we can group together the instances that have similar effects. Each group will have a prototypical effect that is similar to the effects of

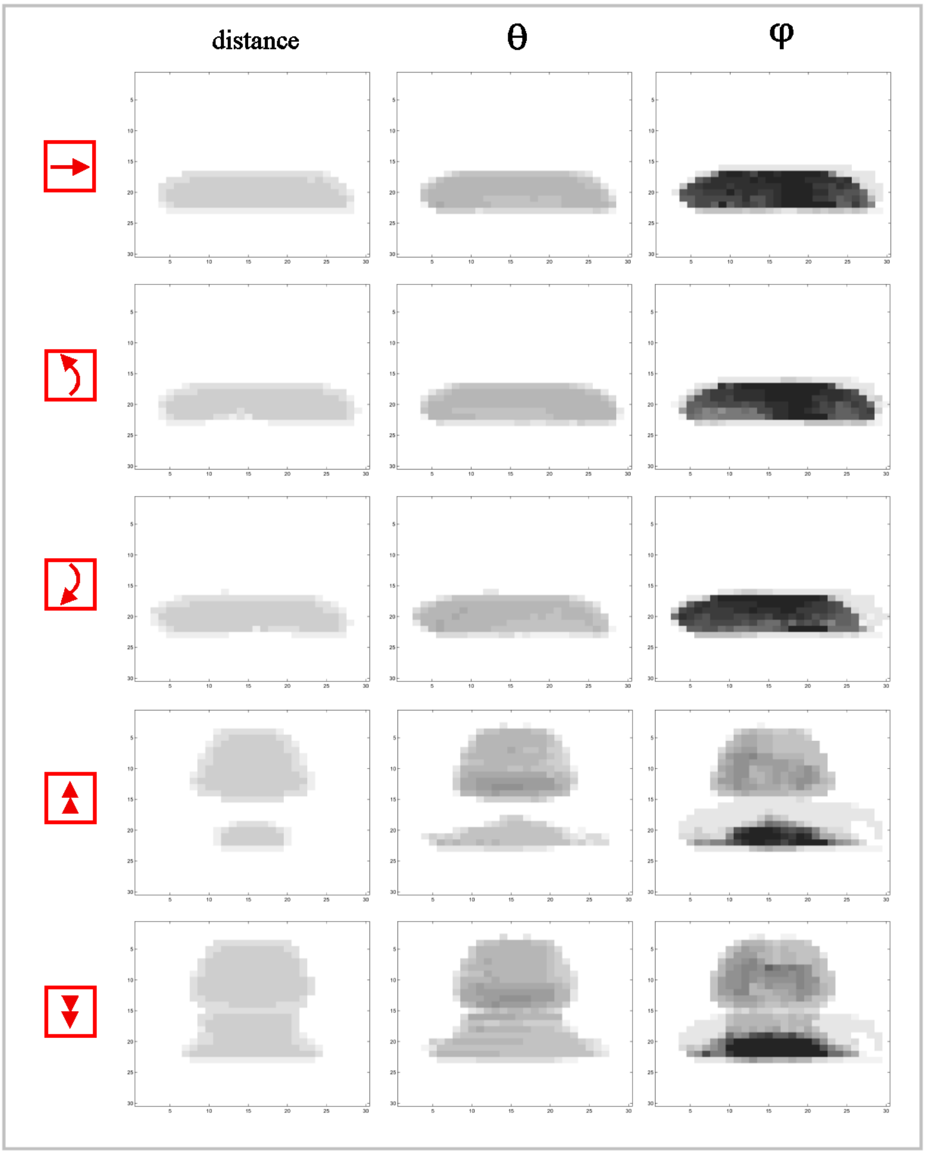


Figure 6.2: Relevant effect features. Each action exerts an effect only on a subset of the perceived features. These features are considered relevant for the perception of the effect of the action. A feature is taken to be relevant if it has a an absolute value greater than a certain threshold, in at least a certain number of the interactions performed with the corresponding action. The figure separately shows the grids on which an effect occurs for distance and shape features. The grids on which the action cannot create any effect are indicated with white. Darkness of the grid color increases with the number of relevant features in the grid.

instances in the group, but somewhat different from the prototype effect of other groups. This, in a way, is a method to compactly represent the kinds of effects that an action can create. Instead of memorizing the effects that occurred in every interaction sample, similar ones are grouped together and are represented with a prototype.

An obvious way of grouping the set of samples is unsupervised clustering. In this study, effects are clustered with the k-means algorithm where the similarity of two effects is a weighted sum of the vectoral difference of the effect features originated from the three different sensors, namely the scanner, the encoders and the crane. Each of the three components are normalized with the number of features extracted from that sensor, so as to prevent the domination of scanner features which are significantly more in number.

Clustering is performed over the relevant effect features which are selected as described above. The similarity measure accounts for the non-existence of the features within the relevant feature set as well as the differing numbers of features originated from the different sensors. Reduction of the effect vectors to be clustered, by means of selecting the features that actually differ among samples, does not have any significant effect on the result of clustering, since the features that have the same value over the whole set of samples are ruled out by the clustering algorithm anyway. However, it provides a considerable increase in the speed of clustering when compared with the clustering of total effect feature arrays, however this result does not have any significance for this study since learning occurs as a batch process.

Note that the encoder and crane features are not relevant for all actions. As expected, the crane features turned out to be relevant only for the *lift* and *drop* actions whereas encoder features were relevant for the movement actions. The *move-forward* action mostly created an effect on the encoder feature corresponding to translational displacement, whereas the rotational encoder was selected to be relevant as well, due to interactions in which the robot bumped into an object and was forced to turn either side. On the other hand, for the *turn-left* and *turn-right* actions the rotational displacement was highly relevant, while translational displacement was irrelevant.

After clustering, every *effect class* is assigned an *effect-id* and an *effect prototype*. Two prototype representations are proposed:

- Mean of the individual effects in that class
- The effect which is closer to the mean of the cluster

These prototype representations are calculated or detected for each cluster. The adventages of using either prototype will be discussed in following chapters, where they come to use for estimating future states by applying the prototype effect over a perceived entity.

Note that the effects generated by different actions can sometimes be the same. Clustering the effects of the union of samples collected with all actions would therefore result in a fewer total number of effect classes. Such an approach would also reveal the equivalence of actions. Two actions whose effects are clustered together to form an effect class, would be considered equivalent in cases that the effect predicted for both actions is this particular effect class.

In this study the number of clusters for the k-means algorithm has been set to 20. The means were initialized with randomly selected samples from the sample set. This initialization method is crucial in obtaining uniform clusters, because with random initializations, clustering tends to end up with all samples being in a single cluster, due to the high dimension of the vectors. Despite the initialization with samples, some clusters ended up with very few or zero members, while some attracted a majority of the samples. This is indeed due to the nature of interaction and the randomization of interaction environments. For example, the *lift* action can actually result with an object being lifted in a very small subset of its interactions, and in all other cases the interaction ends with no effect at all. This means that a majority of the interaction samples of this action has zero effect. It is then expected that these samples be clustered together, to form a single large cluster, while the samples in which a certain effect occurs, be distributed over few other clusters. Hence, the total number of effect classes is not 20 in for all actions since some clusters are left with no samples.

Although the clustering process is unsupervised, one can find conceptual correspondants for the emerging clusters. Figure Figure 6.3 and Figure Figure 6.4 presents such representative clusters for each action. The clusters obtained for each action are interpreted below.

**Move-forward** For the *move-forward* action, the cluster presented in (a) contains interaction samples in which the object in the environment was a box object and was placed close in front of the robot. The forward displacement feature of the effect prototype of this cluster has a very small value since the motion of the robot is blocked by the box object. The example cluster given in (b) contains samples in which the object was placed ahead of the robot, but far enough not to block its motion. Hence, the prototype of this cluster has a large forward displacement. Observing the change in mean distance on the grids of the scanner data, one can see that there is a decrease in the distance in the middle region of the scan image, where the object appears after the motion, while an increase occurs in the grids that the object appeared prior to the action and disappear after the action. In other words the object covers more grids, moves down on the grids since it gets closer with the action. Notice that in (a) there is no such coherent structure in the effect prototypes and the sample effect is rather different form the average effect of the cluster. The reason for this is that the robot is forced to turn either side when it bumps to the object, depending on the orientation of the box, resulting in a rather disturbed average effect for the cluster.

**Turn-left/right** In the effect prototype examples of the *turn-left* action, the object moves in the right direction among the grids of the range image. When it is initially on its left it appears on the front after the action as in the prototypes given in (a), whereas if it is initially on its front it disappears from the middle, and appears on the grid to the right. The inverse happens for the *turn-right* action, in which the object moves towards the left in grids. This shows that the robot learns two impotent effects of turn actions, which are clearing its front, as in example effects classes given in (a) for both actions, and bringing an object to its front, as in examples given in (b). The second effect can be obtained with one of the two actions depending on the position of the object in the environment. Notice that, in the interaction samples of each cluster, the objects are radially positioned roughly on the same angle according to the robot, which means that the grids in which the effect occurs is relevant for clustering, while distance of the objects is not relevant.

**Lift/Drop** The *lift* action has very few effect classes because it can create an effect only when a box object is placed exactly under the crane magnet. All of the interactions in which the crane is already loaded with an object or the object is a sphere or is not placed on the front of the robot, result with no effect as in the example given in (b). A few interactions result with the object being actually lifted. This gives an effect in which the rope-tension feature increases, and a very close object appears in the upper grids of the range image while the object on the front disappear, as in example (a). The *drop* action creates no effect when its crane is not initially loaded. In the case that it is loaded the *drop* action results in the object in the middle upper grids disappearing. If initially there are no objects on the floor under the crane magnet, the dropped object will appear on the lower middle grids after the *drop* action, as in the example of (b). If there is a box under the dropped object, the two objects will be stacked, and as a result the dropped object will reappear in the center grids of the range image, as in (a). If there is initially a sphere under the dropped object or a box which does not cover the center of mass of the dropped object, then the dropped object will fall down and reappear in a random location in the range image.

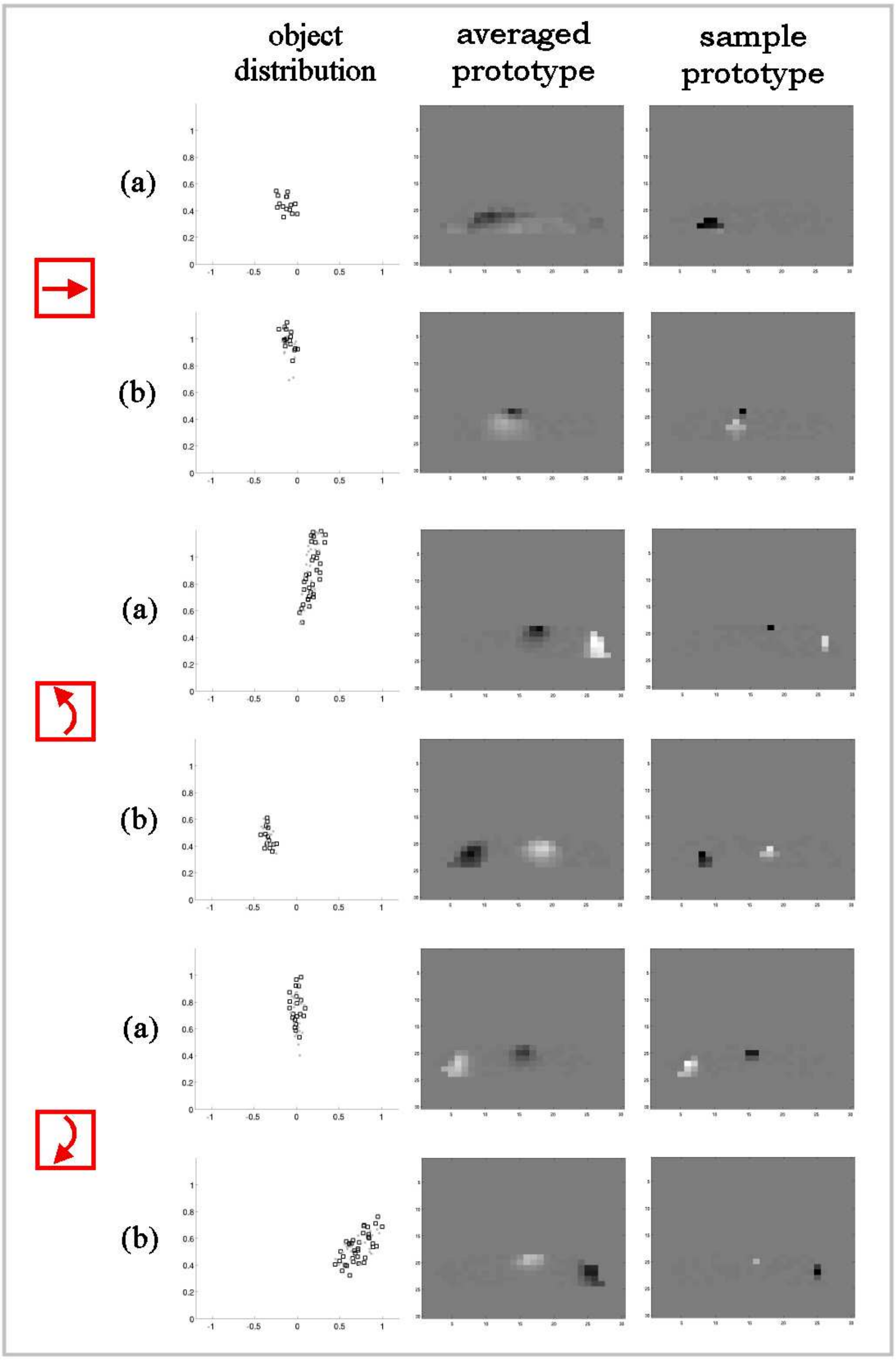


Figure 6.3: Effect class examples for *move-forward*, *turn-left* and *turn-right* actions. See Figure Figure 6.4 for details on how to interpret the images, and see text for interpretation of the data.

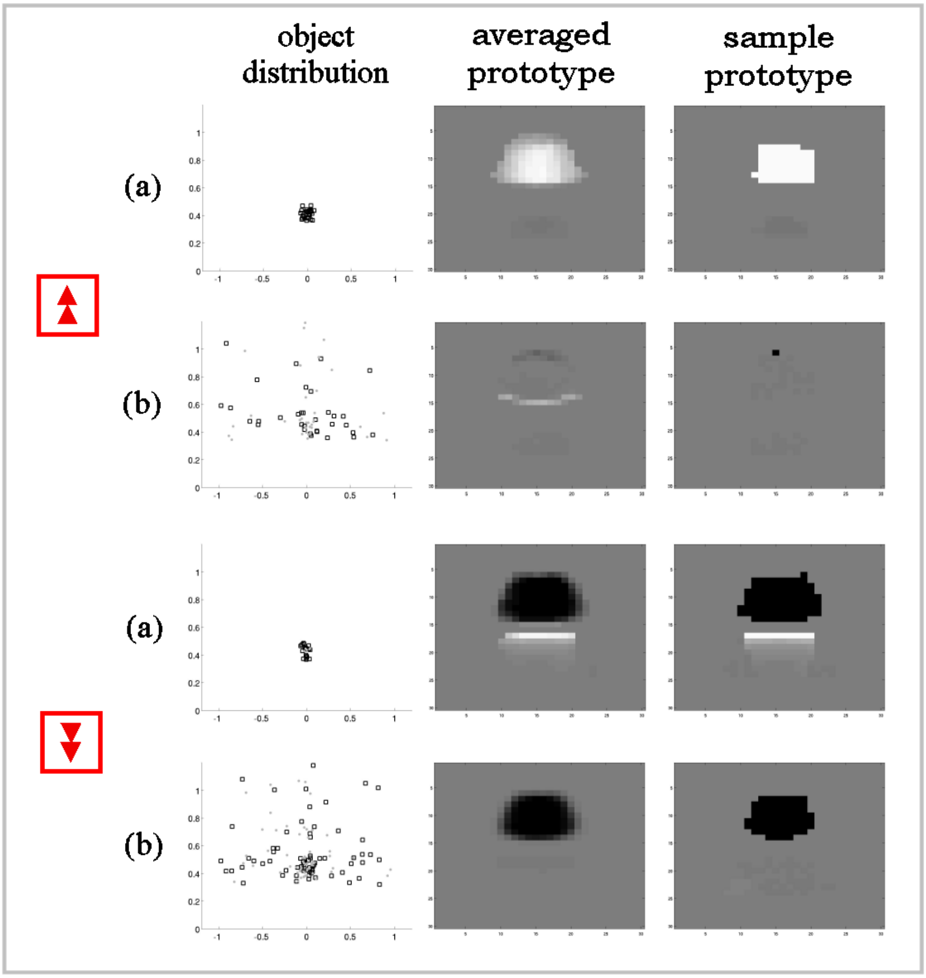


Figure 6.4: Effect class examples for *lift* and *drop* actions. Two representative effect clusters are presented for each action. The first column shows where the object was placed in the interaction samples in the cluster. Box objects are shown with squares and sphere objects are shown with circles. The other columns include representations of effect prototypes. The images correspond to the change in the *mean distance* over the $30 \times 30$ grid. Grids that do not change have a medium gray color, as in the backgrounds of the images. Grids in which the distance increases, tend to darker gray and black, whereas, grids in which the mean distance decreases tend towards white. The second column gives the average of all effects in the cluster whereas the third column is the effect of one sample in the cluster, the one which is closest to the mean of the cluster. See text for the interpretation of the presented data.

### 6.2.3 Selecting Relevant Entity 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, the relevant features in the *entity* are selected, using the corresponding *effect-ids* as their labels. Selection of relevant features is done using the ReliefF algorithm, originally proposed by Kira and Rendell [31]. 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, 20 samples from every class were randomly selected. As an implementation of ReliefF, the data-mining software WEKA [66] was utilized.

In Figure Figure 6.5, 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.

In the proceeding steps of learning as well as in the execution phase, the entity is represented with only the relevant features of the action under consideration. In other words it is filtered to contain only relevant features. The most relevant 1500 features, which correspond to 4.2% of all features, are used as relevant features. In the execution phase this provide a perceptual economy, since the irrelevant features need not to be extracted.

### 6.2.4 Learning the Relation between Entity and Effect

The relation between an entity and the effect that each action generates over this entity is learned in the form of a classifier, namely a Support Vector Machine (SVM) [62]. The SVM gives a mapping from a filtered entity vector to an effect id. A separate SVM is trained for each action. The training samples consist of the set of collected affordance relation instances. The relevant features of the entity part of these instances are filtered and the correspondant effect-id for the effect part is determined. The SVM is trained with such vector-label pairs, where the set of relevant features is the input and the effect-id is the target value. Thus, a trained SVM predicts an effect-id for a given set of relevant features.

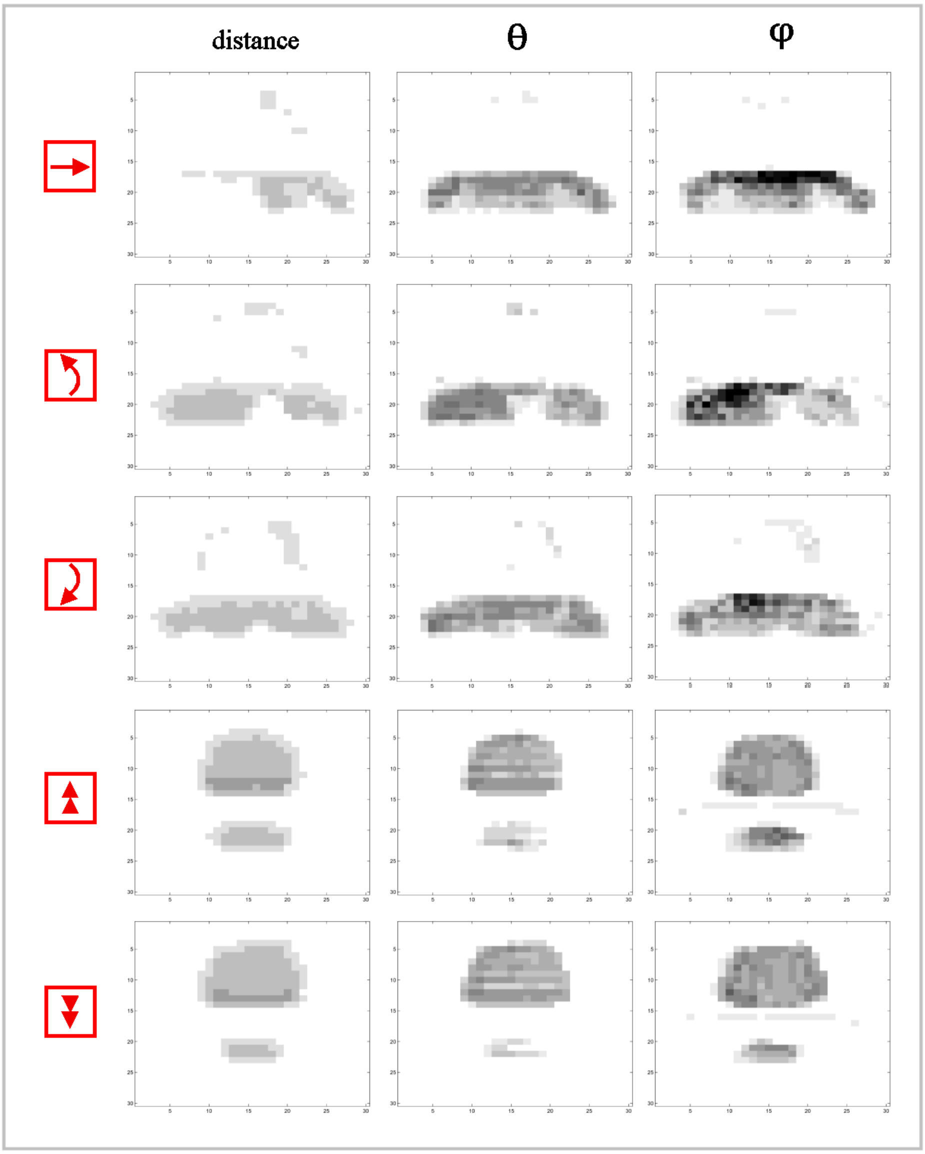


Figure 6.5: Relevant entity features.

# CHAPTER 7

# PLANNING WITH AFFORDANCES

The previous chapter describes a learning scheme with which the robot acquires affordance relations, from interactions with the environment through the execution of a set of predefined actions. The robot has no prior knowledge about these actions. However with learning, it acquires representations of effects that it can generate, and mappings from entities in the initial environment to these effects. Furthermore, it differentiates the perceptual features that are relevant for perceiving entities and effects. It has been demonstrated that such relations can be used to achieve goal directed reactive behavior with the appropriate specification of a motivation. In this chapter the same relations are used in planning action sequences to achieve pre-stated goals.

## 7.1 Goal Specification

It is difficult for a human to externally specify the goal when the robot autonomously learns its own capabilities. One needs to understand the structure of acquired equivalence classes to be able to specify a goal that generates meaningful plans. Three ways of specifying the goal can be proposed:

(1) goal as the effect-id of the desired effect,

(2) goal as a partial effect vector which includes the desired total effect, and

(3) goal as a partial entity vector which describes the desired entity.

The approaches (2) and (3) are more difficult but flexible than (1), since they propose going into the feature level for specifying the goal. The approach (2) describes the goal as desired changes in certain features, while the approach (3) describes the goal as desired final values of features. For example, in order to make the robot lift an object, the goal can be specified as increasing the value of the *rope-tension* feature as well as obtaining a decrease in the distance features among the grids in the upper middle part of the scan image. This requires a careful identification of each feature within the feature vector and a good estimation of what the change in the feature value actually corresponds to. It is certainly much easier to specify the goal as the *effect-id* of the effect class, that includes the samples in which an object was lifted. Indeed, the robot has acquired one such effect class, as can be seen in Figure Figure 6.4.

In this study the approaches (2) and (3) were employed for goal specification. Note that the two specification types can be converted to each other given the current state of the environment. The desired value of a feature equals to the addition of the desired effect onto the current value of the feature. Nevertheless, either method may seem more intuitive depending on the goal to be specified.

## 7.2 Planning

The learned affordance relations provide the capability to predict the effects of actions. These predictions can be used to estimate the future entities that the robot will perceive after the execution of different behaviors. This is achieved by adding the prototype of the predicted effect to the currently perceived entity. Since the effect representation itself is a difference between two entities, adding the effect to a perceived entity will give a new entity which estimates the entity that will be perceived after the execution of the corresponding action.

Having an estimation of the future state, it is then possible to predict the effects of action over future entities, again using the learned relations. The robot can estimate the total effect that a sequence of behaviors will create and it can predict the entity that it will perceive after the execution of the sequence. This constitutes the basic idea for using learned affordance relations in planning sequences of behaviors that lead to a desired goal.

Forward chaining is used for planning. The robot starts by perceiving the present entity, and predicts the effects that each of its primitive behaviors will create. It estimates the five future entities and proceeds by predicting the effects of actions on those future entities and estimating the next entities. This process can be viewed as the breadth-first construction of a plan tree where the branching factor is the number of possible actions. Meanwhile, the robot tests whether the goal is satisfied by the entities in the attained states or by the total effect of the sequence of behaviors that leads to those states. Planning stops when a sequence satisfies the goal. The approach of using forward chaining in affordance-based planning was

**Algorithm 1** Forward-chaining with learned affordance relations

Input the goal as desired effect $E_{goal}$ or desired state $S_{goal}$
Sense environment and extract feature vector $S_{start}$
$S \leftarrow S_{start}$
$Plan \leftarrow \emptyset$
$\mathit{Effect} \leftarrow \bar{0}$
$stateQueue \leftarrow (S, Plan, \mathit{Effect})$
**while** $length(Plan) < MaxPlanLength$ and $stateQueue \neq \emptyset$ **do**
  $(S, Plan, \mathit{Effect}) \leftarrow getFront(stateQueue)$
  **if** $S \approx S_{goal}$ or $\mathit{Effect} \approx E_{goal}$ **then**
    **return** Plan
  **end if**
  **for** each action $a_i$ **do**
    Input $S$ to $SVM_i$
    Get predicted effect $E_i$
    $S_{next} \leftarrow S + E_i$
    $Plan_{next} \leftarrow Plan \cup a_i$
    $\mathit{Effect}_{next} \leftarrow \mathit{Effect} + E_i$
    $addToBack(stateQueue, (S_{next}, Plan_{next}, \mathit{Effect}_{next}))$
  **end for**
**end while**
**return** $\emptyset$ {No plans found}

proposed by Steedman [52].

For this particular case, the time complexity of planning is proportional to the number of states that must be explored. Since each state has 5 possible successive states, which is the number of possible actions, the total number of nodes to be explored is $\frac{5^{d+1}-1}{4}$, where $d$ is the maximum depth of the plan. Thus the algorithm has $O(2^d)$ time complexity. Due to the queue data structure used in breadth first search, the space complexity is similarly $O(2^d)$.

## 7.3 Planning Results

### 7.3.1 Sample Plans in the Simulator

The robot constructs different plans depending on the given goal and the initial situation. Plans obtained for four different goals are presented in the following paragraphs. Three representative initial states are considered for each goal. The specification of the goals is schematically described in Figure Figure 7.1.

**Overcoming obstacles** Given the goal of achieving a certain forward displacement, planning generates the sequence of actions that can overcome obstacles, depending on the initial situation that the robot is placed in.

In this example the goal is specified as a total effect on the translational displacement feature, which correspond to a displacement of approximately $1.2m$. Additionally, the angular displacement feature of the total effect is desired to be zero, in other words, the course of action that the plan produces should not change the heading of the robot. The tolerance is set to be 5% of the maximum change of the feature, which corresponds to a tolerance of approximately $8cm$ for translation and 2.25° for rotation.

The plans generated in three sample environments are presented in Figure Figure 7.2. When the area on the front of the robot is clear or when there is a sphere on its way the plan is simply a sequence of two *move-forward* actions. When there is a box blocking the motion of the robot the plan consists of first clearing the the front and then doing the forward motion. The robot plans to clear its front by either turning or by lifting the object. When it turns to either side or lifts the object at the first step of the plan, in the estimated next state a *move-forward* will predict an effect with a high forward translation that will help achieving the goal. In the case that the robot plans a turn action for the first step, the plan also contains a turn to the opposite side in order to compensate for the change in the

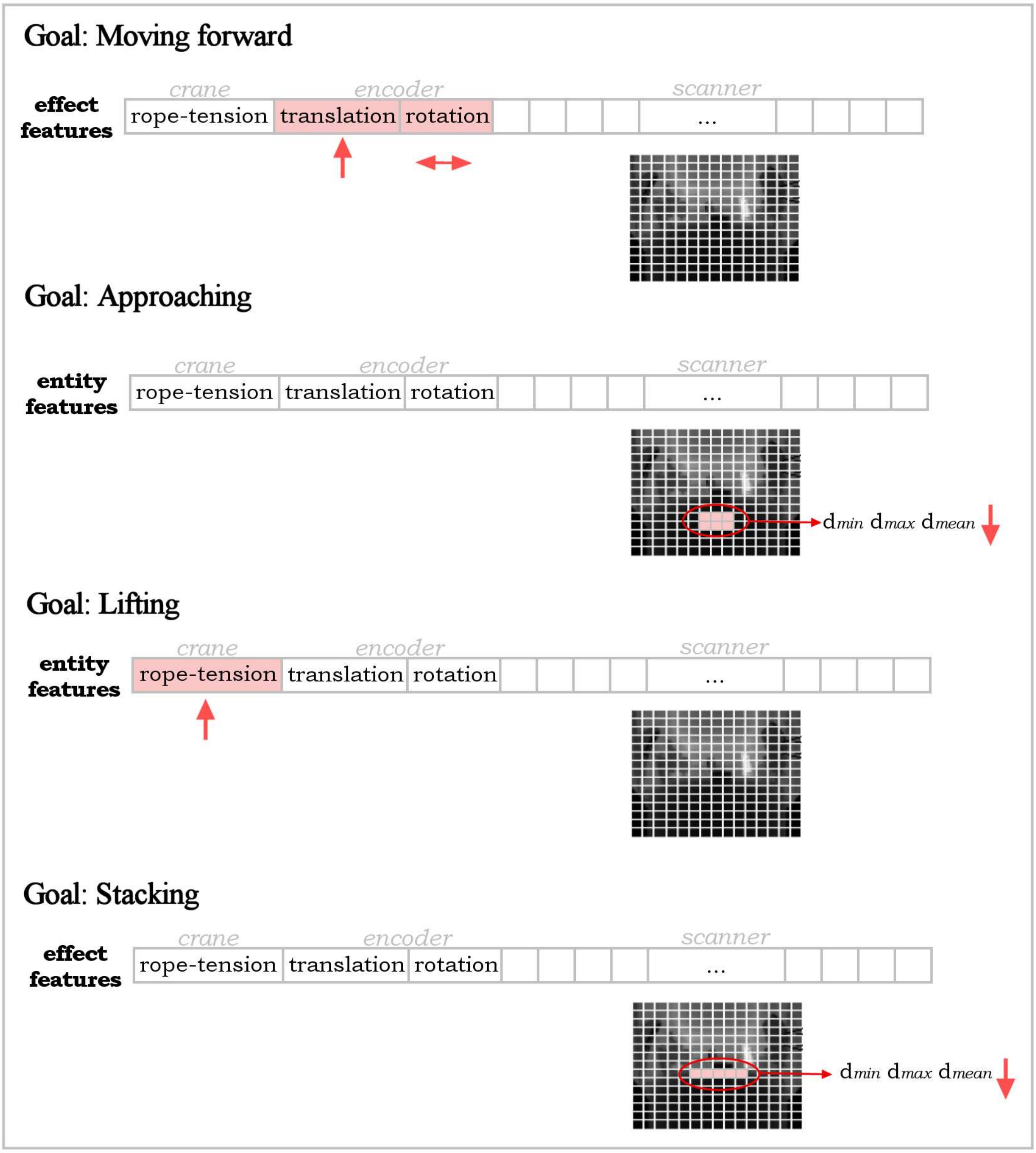


Figure 7.1: Goal specification examples. In the first example the goal is specified as an effect that increases the forward displacement feature and keeps the orientation displacement constant. In the second, the goal is specified as a desired entity in which the middle lower grids of the range image have low distance feature values, which must be achieved by approaching an object. The goal of lifting an object is again a future entity in which the rope-tension feature has a high value. In the last example, the goal is given as an effect that decreases the distance features of the grids in the center part of the range image, in order to make the robot stack objects.

orientation feature that occurs at the first step.

**Approaching Objects** The goal for approaching an object is specified as a desired future entity in which the distance features among the grids in the middle lower part of the range image have a certain value. The robot plans to turn towards the object in the environment and to move forward to get closer to the object so that the goal entity is obtained. Figure Figure 7.2 presents sample plans obtained with this goal.

**Lifting Objects** Goal specification for lifting objects is rather straight forward. It is given as a future state in which the rope tension feature has a certain high value that corresponds to the average weight of objects in the environment. Another way to specify the goal for lifting, is a future state in which the grids on the upper middle part of the range image have small distance values. The distance features of these grids normally have very high values and the only way to have something close on these grids is to lift an object. The lift action predicts an effect that increases the rope tension only in the presence of a liftable object, that is a box, in the area that falls under the magnet. The generated plans therefore consist of several movements that brings the robot in front of an object such that it can lift it. In the absence of a liftable object the robot fails to find a plan that achieves the goal. Generated sample plans are presented in Figure Figure 7.3.

**Stacking Objects** When two boxes are stacked in front of the robot, it perceives an entity in which the grids in the center part of the scan image have small distance values. Such an entity cannot be perceived unless two boxes are stacked, because the height of boxes is within a range that falls only onto lower grids of the range image. This particularity of stacked objects is used for specifying the goal of stacking. The goal is given as a decrease in those center grids. The grids that are specified in the goal can be observed in the sample effect prototype presented for the *drop* action in Figure Figure 6.4 and are roughly shown in Figure Figure 7.1. The sample plans obtained with these goals are given in Figure Figure 7.3. In the first situation, the robot starts with an object on its crane and plans to move forwards to have the other box object close on its front, and drop the object on the crane. In the second case the generated plan consists of lifting the object on the front turning towards the other object and dropping the object on the crane. In the initial state the prediction of the *lift* action is not effected by the second object in the environment because the relevant features for lift are in the middle portion of the range image. No plans can be generated in the last example since there is not stackable object in the environment.

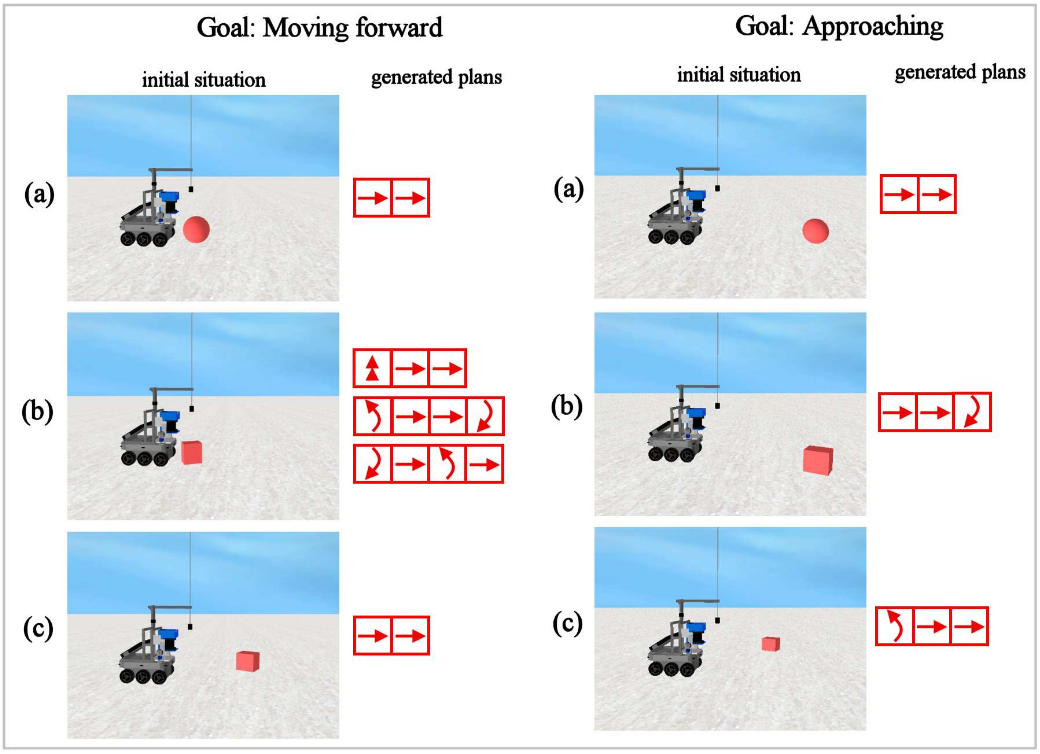


Figure 7.2: Sample plans for moving forward and approaching objects. In the first example, the robot is placed in three different environments with the goal of achieving a forward displacement of about 120$cm$ and maintaining its heading. In (a) and (c) the example plan is a sequence of two forward moves which gives the desired forward displacement without changing the heading. In case (b) the presence of the box changes the prediction for the forward movement at the initial state. The planner generates several plans. In the first it lifts the object and moves forward. In the others it turns to either side to get rid of the object on the front and performs its forward movements afterwards. It also performs a turn to the opposite side to gain its initial heading. In the second example, the goal is given as having something close on the front in the final state. The robot generates plans in which it moves towards the object by turning or moving forward towards the object in the environment.

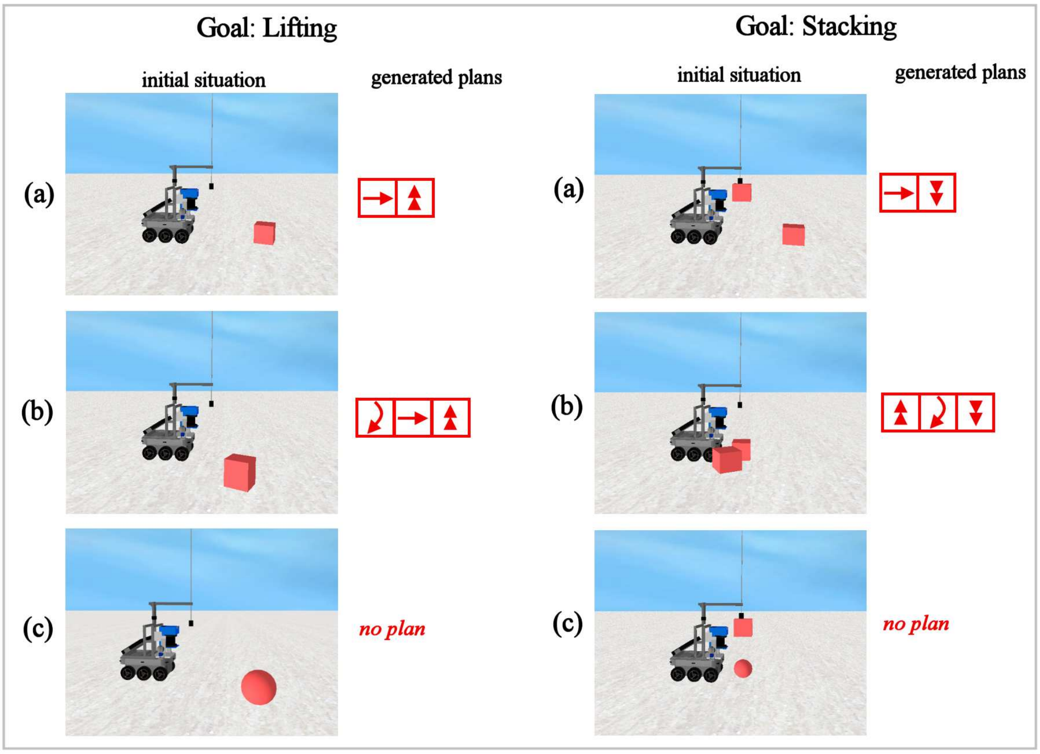


Figure 7.3: Sample plans for lifting and stacking objects. To generate a plan for lifting an object the goal is specified as an increase in the crane rope tension feature. In (a) the robot plans two *move-forward* actions to get the object near on the front so that it can afterwards lift it. Similarly in (b) it turns towards the object moves closer and lifts it. In (c) the robot is unable to find a plan because of the absence of a liftable object. For stacking, the goal is specified as a decrease in the distance features on middle grids of the scan image. In (a) the robot starts with an object on the crane and there is another object a little farther on its front. The robot plans to move forward so that the object becomes closer and it can drop the object on the crane such that the goal is satisfied. In (b) the plan consists of lifting the object on the front, turning towards the other object and dropping the object on the crane. In (c) no plans can be generated because the sphere does not permit stacking.

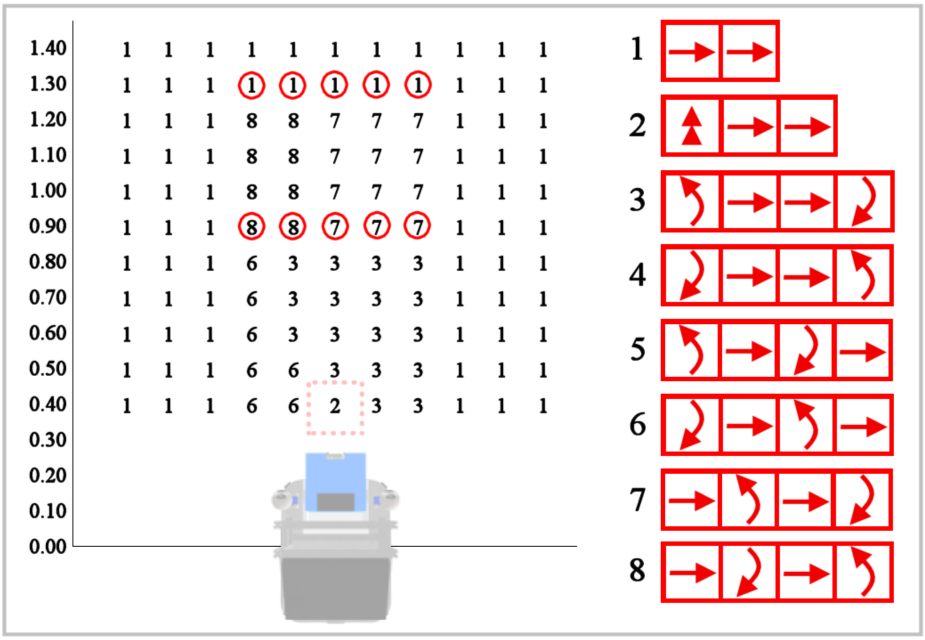


Figure 7.4: Plans obtained for test cases. A box shaped object of size 15 $cm \times 15$ $cm \times 15$ $cm$ is placed on different positions in front of the robot. The planner runs in each case and outputs the selected best plan. The different plans generated for these case are listed on the right. The generated plan for each position of the object are referred from this list.

The learned relations could not be successfully ported to the real robot since the predictions of actions for the perceived initial situation were different in the simulator and the real robot. Identifying the reasons for this is not straight forward since it requires understanding the nature of the classifier.

### 7.3.2 Analysis of Robustness

To analyze the robustness of the planner the goal of moving forward has been considered. A box shaped object is placed on different coordinates in front of the robot to generate and collect several plans. If the planner generates several solutions for one situation, the best plan is selected as the one having the smallest plan depth. In the case of equal depths the one which has the smallest distance to the goal is selected. The maximum plan depth is set to 5.

Figure Figure 7.4 presents the plans that generated for different positions of a box object on the front of the robot. The box is moved away and to the sides with 10 $cm$ intervals. Although these results do not provide qualitative data about the robustness of the planner, they show that the robot is able to consistently generate plans that are meaningful for the

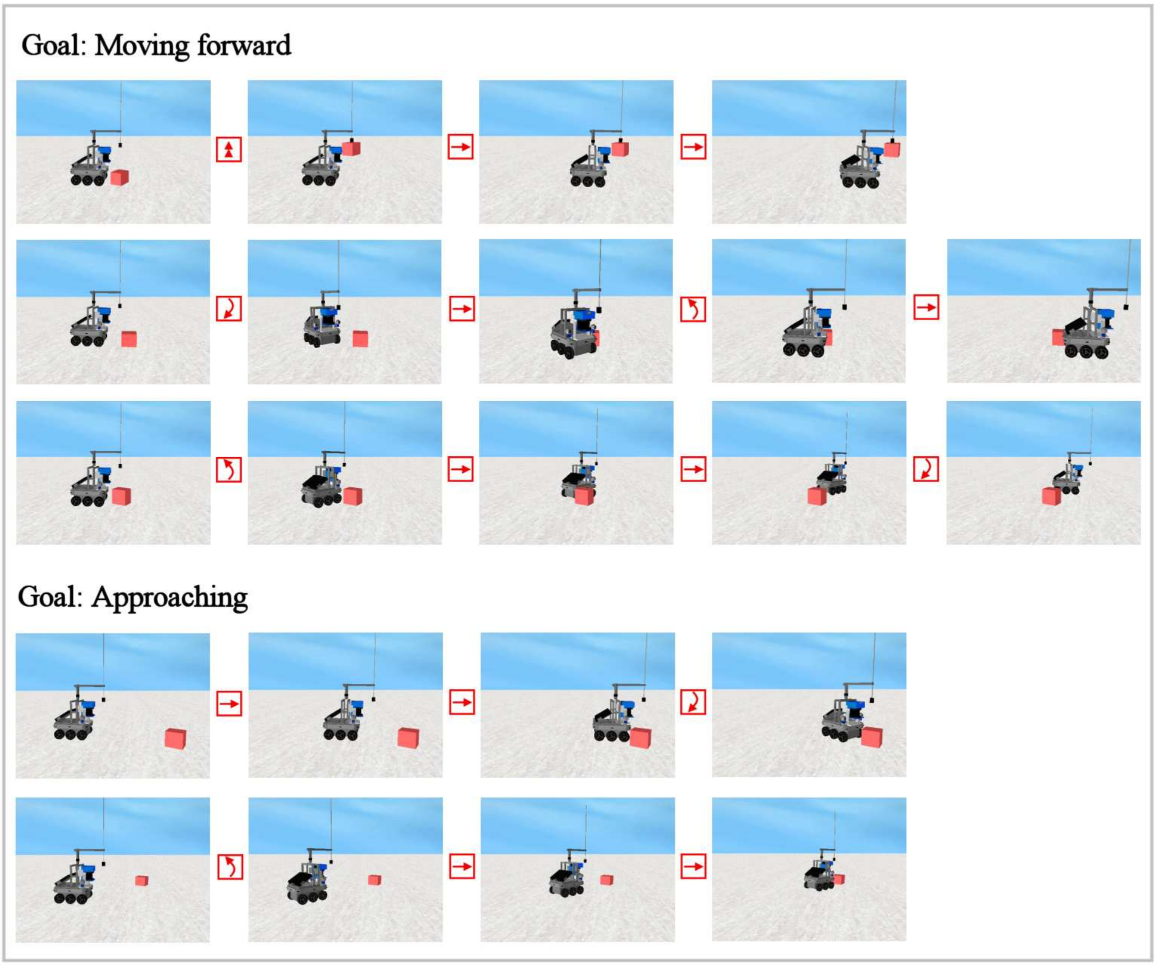


Figure 7.5: Execution of advancing and approaching plans in the simulator.

situations in which they are generated. One may notice that some plans are favored over the others in cases where multiple plans seem to be applicable. This is related to the differences in effect prototypes and to the specification of the goal.

## 7.4 Execution

### 7.4.1 Sample Executions

The course of actions followed by the robot during the execution of the sample plans obtained in the previous section for the four different goals on the simulator are presented in Figure Figure 7.5 and Figure Figure 7.6.

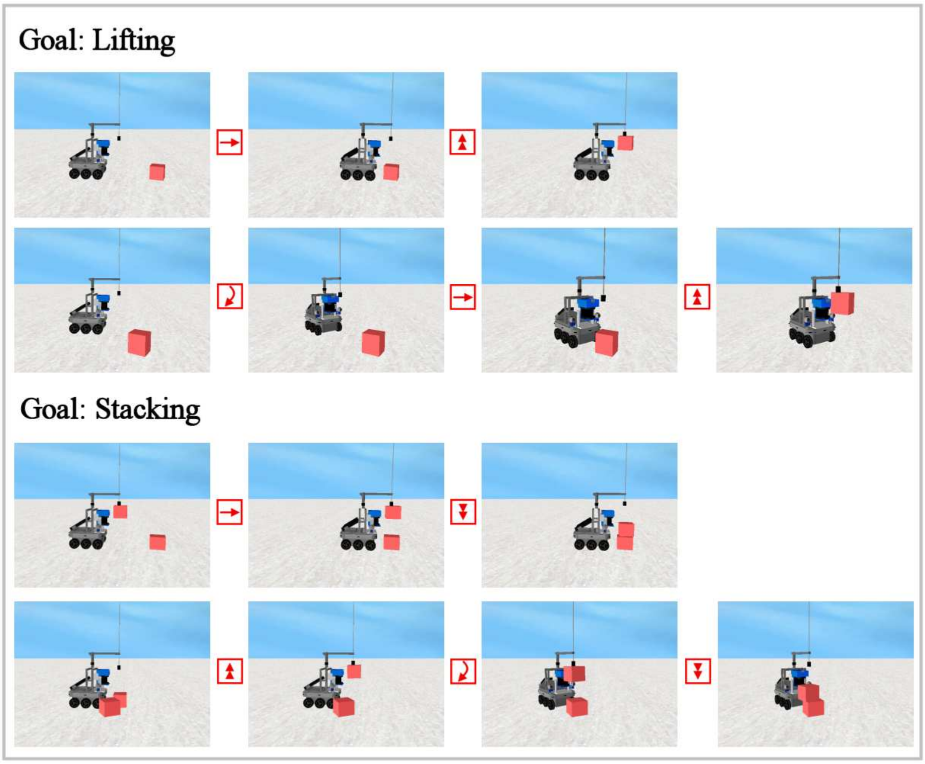


Figure 7.6: Execution of lifting and stacking plans in the simulator.

### 7.4.2 Execution Monitoring

Although this study is mostly concerned with plan generation, the used framework provides an intuitive and efficient way to monitor the execution of actions. The proposed method is to compare the entities that are reached during the course of execution of the planned action sequence, with the ones that were estimated during planning. A straight-forward way to compare entity vectors is to define a similarity measure between entities and set a certain threshold of acceptable similarity. However this approach might become inefficient depending on the similarity measure, considering the high dimensionality of the vectors. Instead, states can be compared in terms of their affordances. More specifically, two entities can be considered to be similar if all possible actions predict the same effects.

With this convention, execution monitoring consists of comparing the effects that are predicted thoughout the course of execution, with the effects that were predicted for estimated future states during planning. Since the planned sequence of effects is supposed to achieve the goal, a deviation from these predictions within the course of execution may result with a failure of the plan. While it is possible to limit the comparison to only the prediction for the action that is to be executed according to the plan, a more precise comparison would consider predictions of other actions as well.

The first approach has been applied on the plans that were generated for the cases presented in Figure Figure 7.4. The robot executes the plan and perceives the environment after the execution of each of its actions. It compares the effect predicted for the next action, with the effect that was predicted during planning. The plans that are circled in Figure Figure 7.4 are the ones that have failed to produce the predicted effects by colliding with the box object. The failures can be linked to the insufficiency of the averaged effect prototypes. Since the intensity of the effect prototype is lower than the actual effect, planning underestimates the forward movement of the robot and therefore collides with the objects during execution. The collision results it a deviation from the actual path of the action and the monitor detects a failure.

## 7.5 A Comparison of Affordance-based and Classical Planning

Although the algorithm described above corresponds to a simple forward chaining, there are crucial differences between the planning approach presented in above, and the classical planning approach. These can be summarized with a few points as follows.

**State-space** The state space of a planning domain is the set of all facts that describe the configuration of the environment. In classical planning, the state space consists of all propositions and predicates that are defined in the planning domain. Since the value of a predicate is either true or false and the set of predicates is bounded, the number of possible configurations of the environment is finite. In the planning approach presented in this thesis, the state corresponds to the perceived entity which is represented in terms of low-level perceptual features. The entity used for prediction is the subset of the feature vector and contains only the relevant features for each action. Therefore the state space can be considered as the finite set that consists of the union of relevant features for all possible actions of the robot. The number of configurations that the environment can have is, on the other hand, infinite since the value of a feature is continuous.

**Operator structure** In classical planning the plan operators are mappings from a bounded set of preconditions to a bounded set of effects, linked with an action. A different operator must be defined for the different executions of a certain action in different conditions. In this study the plan operators are affordance relations which are mappings from an entity to an effect-id. It maps a feature vector with continuous values to a symbol that represents the effect that an action will create on the feature space. Although the effect of the operator is explicit as in the classical operator, the precondition is inherent in the classifier that maps an entity to one of the effects. The set of entities for which the same effect are predicted can be said to satisfy the same precondition. Remark that this set is infinite.

In the classical approach the operators must be carefully defined not to exclude any possible state of the environment. Otherwise, the planning algorithm can reach states in which none of the possible operators is applicable. In the presented approach any possible configuration of the entity is mapped to an effect by the classifier. This is how the approach deals with the infinite possible configurations of entities.

The operators used in this study can in fact be converted to operators of a planning method called numeric planning. This requires that the mapping of the classifier be explicitly represented as a set of rules. The classifier used in this study, is a SVM, which determines the target value of a new input vector based on the value of a non-linear continuous function specified in terms of the learned support vectors and several constant parameters. Using these functions the preconditions of an operator can be written as a range for the value of the function, when the current entity is substituted in the function. The effect of the operator will be the effect prototype of the effect class that satisfies the corresponding precondition. In this study each of the 5 actions have at most 20 effect classes, which means that a maximum of 100 operators will be defined in this way. These operators can be used by any continuous planner, given the initial entity and the goal specified in the same way as it is specified in this study.

**Complexity** Due to the basic differences between the operator structures of the two approaches the complexities of planning algorithms using these operators will also differ. In a classical forward chaining, the number of successive states for a present state depends on the number of operators that the present state satisfies. Thus, the total number of nodes to be tested for the goal will differ. On the other hand the precondition test will be performed on all possible operators in each state. In the presented approach, the number of successive states is constant and is determined by the number of possible actions. Nevertheless, the complexity in both cases is exponential.

**Goal Specification** In classical planning the goal is given as a set of propositions or predicates that must be true. The goal can be considered as a subset of the state space, and the number of possible ways to describe a goal is finite. In the presented approach however, the number of ways to specify a goal is infinite since it is specified as value ranges on perceptual features.

## 7.6 Extending the Formalism to Action Sequences

In Chapter 6, while collecting affordance relation instances from the interactions of the robot, the behavior aspect of the relation instance corresponded to one of the five atomic actions of the robot. Since the presented learning scheme did not account for any kind of behavioral learning, the behavior aspect remained the same in affordance relations. A number of affordance relations were learned for each of the five atomic actions.

The planning approach presented in this study provides an opportunity to learn the affordances of action sequences. Given an entity, the algorithm generates sequences of actions that achieve a certain total effect. Hence, the entity aspect of the relation is the entity perceived at the time that the plan is generated, the behavior aspect is the sequence of actions which constitutes the plan and the effect is the total effect that the action sequence will create.

A number of different action sequences can be planned for a given goal, in different situations of in the same situation. The total effect that each action sequence creates will achieve the given goal. These effects are said to have *effect equivalence* since they all achieve the same goal, though they might all have different side effects.

The robot sometimes generates the same plan in different situations for a specified goal. The entities perceived in these different situations have *entity equivalence.* Given a certain goal the robot can plan several action sequences in one situation. These actions sequences are said to be *behaviorally equivalent.* Similarly, the robot plans different action sequences to achieve the same goal in different situations. The perceived entities of the different situations paired with the action sequence generated for that situation form an affordance equivalence class.

One can find examples of the equivalence notion in the plans generated for the cases given in Section 7.3.1. As shown in Figure Figure 7.2 the plans generated for the moving forward goal in cases (a) and (c) are the same. In situation (a) there is a sphere in front of the robot whereas in situation (c) the front of the robot is clear. The entities of the two different situations are equivalent since the same goal is achieved with the same action sequence in the two different situations.

Again in Figure Figure 7.2, three sample plans are presented in situation (b) for the goal of moving forward. The three example action sequences have behavioral equivalence, because they achieve the same goal in the same situation.

Lastly, consider the three plans generated in the three different cases for the goal of approaching, as shown on Figure Figure 7.2. Three different action sequences achieve the same goal for three different perceived entities. The entity-action sequence pairs in these examples are said to have affordance equivalence.

The extention of the formalism to planned action sequences shows that affordace relations can be acquired not only with learning mechanisms but also with high-level inference mechanisms such as planning. Such a mechanism is essential since the motor capabilities of even a simple robot with a limited number of atomic actions consists of unlimited number of action sequences. It is not feasible to learn the affordance relations for all possible action sequences. Biological systems, certainly have a mechanism to infer the effects of action sequences on entities, since they have an unbounded repertoire of motor actions.

The aim of this thesis is not to construct a highly specialized and robust planner for the specific robot used in the application of the approach. An implementation of the approach is presented without claiming that it is optimal. For instance clustering and classification methods should not be taken as mandatory components of the affordance-based learning since the choice of these methods is rather arbitrary. Indeed, several issues faced in this study are related to these choices. The following section presents some of these issues and proposes several future improvements.

## 7.7 Issues and Future Improvements

Some of the issues and concerns related with this study are mentioned in the following items.

- Several issues arise at the clustering step in the learning phase. The obtained clusters usually contain outliers which distort the prototype of the clusters. Averaging over clusters to obtain the prototype reduces the intensity of the prototype whereas taking the single effect closest to the mean as the prototype may end up making unintended generalizations, that do not hold for all the samples in the cluster. Also, the clustering method sometimes fails to distinguish effects that can make a significant difference for the goal. For example, clusters for turning actions do not distinguish the shapes of the objects. This results in effect prototypes in which shape features are mixed up. For instance, when the robot estimates a future state for a *turn-left* while there's a sphere on its left, the predictions in the estimated future state may be as if there was a box, instead of a sphere in front of the robot. This is a result of the effect prototype being the average of a cluster which includes effects in which the object on the left was a sphere as well as a box. Clustering therefore needs to differentiate such cases. This requires not only increasing the number of clusters, but also defining better similarity measures for clustering.

- The forward chaining algorithm does not address the irrelevant action problem, which means that each possible action is taken into account while extracting the successor states, even though they might be irrelevant for the goal. The computational cost due to this, can be reduced by the use of heuristics, based on prior knowledge about the goal and the actions, in better search algorithms like A*. A backward search is implausible in the framework of this study since it would require the specification of realistic goal states as complete vectors.

- The number of interactions used in learning was relatively small and the majority of the interactions were not interesting, as in the case of interactions of the *lift* action, in which an object could actually be lifted in very few interactions and in most of the interactions ended with no effect. This problem can be overcome by making smart interactions as proposed in [61]. Furthermore, the interaction environments were restricted in several aspects which limited the planning scenarios for the robot to rather simple ones. The interactions should also include environments with multiple objects of a variety of shapes and sizes.

- The action repertoire of the robot is limited to five discrete actions and the acquisition of behavior equivalence classes is skipped in this study. With a set of parametrized actions, the capabilities of the robot can be increased, while planning can still be possible with behavior equivalence classes.

# CHAPTER 8

# CONCLUSION

This thesis proposes to use a new formalization of affordances as a framework for acquiring representations of the capabilities in an environment and for using these representations in planning. Such a framework is claimed to be essential to achieve robots that are characterized as reactive, deliberative and adaptive.

The robot learns what it can do in the environment by interacting with it. From these interactions the robot learns the different types of effects it can create in the environment and discovers the invariants of the environments that support the generation of these different effects, with different actions. When placed in different environments, the learned relations provide the robot with the ability to predict the effects that each of its actions will create in the environment. With such predictions the robot can estimate future states and make predictions over the estimated future states. Proceeding in this way, the robot can find the sequence of actions that achieve a given goal.

The work presented in this thesis can be considered as a solution approach to the problems within robot planning. As McDermott suggest, the models of the robot behavior on which planning relies are crude [37]. This is linked to the fact that these models are built into the robots by experts and they consider the robot and the environment separately. Instead, as proposed in this thesis, models on which robot planning is based, should account for the interactions of the robot with its environment and should be acquired by the robot. This study has shown that the affordance formalism proposed by Şahin et al. [9] can be used as a framework for acquiring such models that are based on interactions and that can be used in planning.

The study revealed the problem of goal specification and its implications for communication to be significant. Goal specification is in fact the way in which the human communicates with the robot. Specifying the goal in terms of the *ids* that are assigned to the possible effects of the robot, corresponds to communicating with the robot through its acquired symbols. Specifying the goal in this way is much easier than specifying it with low level features, which in line with the fact that symbols facilitate communication.

# REFERENCES


[1] R.S. Amant. Planning and user interface affordances. In *Intelligent User Interfaces*, pages 135–142, 1999.

[2] R.C. Arkin. *Behavior-based Robotics.* MIT Press, Cambridge, MA, USA, 1998. ISBN:0262011654.

[3] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, (47):139–159, 1991.

[4] A. Chemero. An outline of a theory of affordances. *Ecological Psychology*, 15(2):181–195, 2003.

[5] R. Cooper and D. W. Glasspool. Learning action affordances and action schemas. In Robert M. French and Jacques P. Sougne, editors, *Connectionist Models of Learning, Development and Evolution*, Perspectives in Neural Computing, pages 133–142, London, 2001. Sixth Neural Computation and Psychology Workshop, Springer-Verlag.

[6] S. Coradeschi and A. Saffiotti. Perceptual anchoring of symbols for action. In *17th International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001.

[7] I. Cos-Aguilera, L. Canamero, and G.M. Hayes. Motivation-driven learning of object affordances: First experiments using a simulated Khepera robot. In *In Proceedings of the 9th International Conference in Cognitive Modelling (ICCM'03)*, Bamberg, Germany, 4 2003.

[8] I. Cos-Aguilera, L. Canamero, and G.M. Hayes. Using a SOFM to learn object affordances. In *In Proceedings of the 5th Workshop of Physical Agents*, Girona, Catalonia, Spain, 3 2004.

[9] E. Şahin, M. Çakmak, M.R. Doğar, E.Uğur, and G. Üçoluk. To afford or not to afford: From a theory of affordances to an affordance-based control architecture. Technical Report 2, Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey, 2006.

[10] 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.

[11] 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, draft, version 1.

[12] M.R. Doğar, M. Çakmak, E. Uğur, and E. Şahin. From primitive actions to goal-directed behavior. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[13] A.P. Duchon, W.H. Warren, and L.P. Kaelbling. Ecological robotics. *Adaptive Behavior*, 6(3):473–507, 1998.

[14] M. Çakmak E. Uğur, M.R. Doğar and E. Şahin. The learning and use of traversability affordance using range images on a mobile robot. Technical Report 3, Dept. of Computer Engineering, Middle East Technical University, Ankara, Turkey, 2006.

[15] W.V. Fabricius. The development of forward search planning in preschoolers. *Child Development*, 59(6):1473–88, 12 1988.

[16] R.E. Fikes. Monitored execution of robot plans produced by STRIPS. In *Information processing 71; Proceedings of the Congress*, pages 189–194, 1972.

[17] J. Firby. *Adaptive Execution in Complex Dynamic Worlds*. Technical report, yaleucsdrr #672,, Yale University, 1 1989.

[18] 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.

[19] V. Gallese. Agency and motor representations: new perspectives on intersubjectivity. SC working papers 2000-6. Workshop on autism and the Theory of Mind. Lyon, May 2000. http://www.isc.cnrs.fr/wp/wp00-6.htm.

[20] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119:593–609, 1996.

[21] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning Theory and Practise*. Elsevier, 2004.

[22] E.J. Gibson. Perceptual learning in development: Some basic concepts. *Ecological Psychology*, 12(4):295–302, 2000.

[23] E.J. Gibson. Where is the information for affordances? *Ecological Psychology*, 12(1):53–57, 2000.

[24] J.J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, 1986. Originally published in 1979.

[25] R. Goldman and M. Boddy. Expressive planning and explicit knowledge. In *Proceedings of 3rd International Conference on AI Planning Systems*, pages 110–117, 1996.

[26] J. Greeno. Gibson's affordances. *Psychological Review*, 101(2):336–342, 1994.

[27] S. Harnad. The symbol grounding problem. *Physica D*, 1990.

[28] G. Humphreys. Objects, affordances ... action !!! *The Psychologist*, 14(8):5, 8 2001.

[29] J. Irran, F. Kintzler, and P Poelz. Grounding affordances. In R. Trappl, editor, *Cybernetics and Systems*. Austrian Society for Cybernetic Studies, 2006. ISBN: 3 85206 172 5.

[30] H. Kautz and B. Selman. Planning as satisfiability. In *ECAI '92: Proceedings of the 10th European conference on Artificial intelligence*, 0-471-93608-1, pages 359–363, New York, NY, USA, 1992. John Wiley & Sons, Inc.

[31] K. Kira and L.A. Rendell. A practical approach to feature selection. In *ML92: Proceedings of the ninth international workshop on Machine learning*, pages 249–256, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[32] W. Kohler. *The mentality of Apes*. Routledge, 1925.

[33] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini. Developmental robotics: a survey. *Connection Science*, 15(4):151–190, 2003.

[34] K.F. MacDorman. Grounding symbols through sensorimotor integration. *Journal of the Robotics Society of Japan*, 17(1):20–24, 1999.

[35] 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 Francisco, California, USA, 2000.

[36] J. McCarthy and P. J. Hayes. *Some philosophical problems from the standpoint of artificial intelligence.* 0-934613-45-1. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[37] D. McDermott. Robot planning. *AI Magazine*, 13(2):55–79, 1992.

[38] 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.

[39] U. Nehmzow and J. Pisokas. Prediction of perception for subsymbolic action planning in mobile robots. In *European Conference on Mobile Robotics, ECMR, Warsaw*, 2003.

[40] U. Neisser. Multiple systems: A new approach to cognitive theory. *The European Journal of Cognitive Psychology*, 6:225–241, 1994.

[41] A. Newell. Physical symbol systems. *Cognitive Science: A Multidisciplinary Journal*, 4(2):135–183, 1980.

[42] A. Newell and H.A. Simon. *Human Problem Solving.* 0134454030. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1972.

[43] J. Norman. Ecological psychology and the two visual systems: Not to worry! *Ecological Psychology*, 13(2):135–145, 2001.

[44] O.Soysal. A systematic study of probabilistic aggregation strategies in swarm robotic systems. Master's thesis, Middle East Technical University, 2005.

[45] J. Pisokas and U. Nehmzow. Experiments in subsymbolic action planning with mobile robots. In Daniel Kudenko, Dimitar Kazakov, and Eduardo Alonso, editors, *Adaptive Agents and Multi-Agent Systems III: Adaptation and Multi-Agent Learning*, volume 3394/2005 of *0302-9743*, page 216, 2005.

[46] J.L. Pollock. The logical foundations of goal-regression planning in autonomous agents. *Artificial Intelligence*, 106(2):267–334, 1998.

[47] G. Rizzolatti and M. Gentilucci. *Motor and visual-motor functions of the premotor cortex*, pages 269–284. Chichester: Wiley, 1988.

[48] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. 0-13-103805-2. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[49] E.D. Sacerdoti. *A structure for plans and behavior*. PhD thesis, SRI International, Menlo Park, CA., 1975. AAI7605794.

[50] M.J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, 1987.

[51] M. Steedman. Origins of universal grammer in planned action. University of Edinburgh.

[52] M. Steedman. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy*, 25, 2002.

[53] T.A. Stoffregen. Affordances as properties of the animal environment system. *Ecological Psychology*, 15(2):115–134, 2003.

[54] 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.

[55] 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.

[56] G.J. Sussman. *A Computer Model of Skill Acquisition*. Elsevier Science Inc., New York, NY, USA, 1975. ISBN: 044400159X.

[57] A. Szokolszky. An interview with Eleanor Gibson. *Ecological Psychology*, 15(4):271–281, 2003.

[58] M. Taddeo and L. Floridi. Solving the symbol grounding problem: a critical review of fifteen years of research. *Journal of Experimental and Theoretical Artificial Intelligence*, 17(4):419–446, 2005.

[59] M.T. Turvey. Affordances and prospective control: an outline of the ontology. *Ecological Psychology*, 4(3):173–187, 1992.

[60] E. Uğur. Direct perception of traversability affordance on range images through learning on a mobile robot, Sept. 2006.

[61] E. Uğur, M.R. Doğar, M. Çakmak, and E. Şahin. Curiosity-driven learning of traversability affordance on a mobile robot. In *Proceedings of the IEEE International Conference on Development and Learning*, 2007.

[62] V.N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, September 1998.

[63] D. Warren. Generating conditional plans and programs. In *Proceedings of AISB Summer Conference*, pages 44–354, 1976.

[64] D.S. Weld. Recent advances in ai planning. *AI Magazine*, 20(2):93–123, 1999.

[65] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599 – 600, 1 2001.

[66] 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 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, 2007.4 (in press).*

[2] 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.*

[3] 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.*

[4] 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).*

[5] 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).*