DIRECT PERCEPTION OF TRAVERSIBILITY AFFORDANCE ON RANGE IMAGES THROUGH LEARNING ON A MOBILE ROBOT

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

BY

EMRE UĞUR

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

SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Erol Şahin
Supervisor

Examining Committee Members

Prof. Dr. Volkan Atalay	(METU, CENG)	
Assist. Prof. Dr. Erol Şahin	(METU, CENG)	
Assoc. Prof. Dr. Göktürk Üçoluk	(METU, CENG)	
Assoc. Prof. Dr. Sibel Tarı	(METU, CENG)	
Assoc. Prof. Dr. Osman Parlaktuna	(ESOGU, EE)	

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: Emre Uğur

Signature :

ABSTRACT

DIRECT PERCEPTION OF TRAVERSIBILITY AFFORDANCE ON RANGE IMAGES THROUGH LEARNING ON A MOBILE ROBOT

Uğur, Emre

M.S., Department of Computer Engineering

Supervisor : Assist. Prof. Dr. Erol Şahin

September 2006, 85 pages

In this thesis, we studied how physical affordances of the environment, such as traversibility for a mobile robot, can be learned. In particular, we studied how the physical properties of the environment, as acquired from range images obtained from a 3D laser scanner mounted on a mobile robot platform, can specify the traversibility affordance. A physics based simulation environment is used during exploration trials, where the traversibility affordances and the relevant features for each behavior are learned through physical interactions with the environment. The prediction accuracy in perceiving the traversibility affordances of the world, which includes several spherical, cylindrical and box shaped objects, is found to be 94%. Furthermore, it is observed that the robot uses only 1.1% of extracted features while perceiving the affordances. This in turn saves the time 76.6% in scanning and 81% in feature processing. The robot is later tested in a simulated cluttered environment, surrounded by walls. It is able to successfully traverse in the environment, by selecting its behaviors based on the affordances provided, and performing them. The robot was able to avoid from the box shaped objects, and push-roll the spherical ones without making any object detection. In the last set of experiments, the trained affordance-based behavior selection scheme is partially verified in the real world with the Kurt3D robot.

Keywords: Affordances, Robotics, Traversibility, Direct perception

ÖZ

GEZGİN BİR ROBOTTA, ÖĞRENME YOLUYLA, UZAKLIK GÖRÜNTÜLERİNDEN ORTAMIN GEZİLEBİLİRLİĞİNİN DOLAYSIZ ALGISI

Uğur, Emre

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Assist. Prof. Dr. Erol Şahin

Eylül 2006, 85 sayfa

Bu tezde, ortamın gezer bir robota sunduğu fiziksel sağlarlıklar (geçilebilirlik gibi) çalışılmıştır. Özel olarak ise, 3-boyutlu lazer tarayıcı yardımıyla elde edilen uzaklık resimlerinden hesaplanan ortamın fiziksel özelliklerinin geçilebilirliğe etkisi incelenmiştir. Keşif deneyleri için fizik tabanlı bir simülatör kullanılmış ve ortamın sağlarlıklarıyla beraber değişik davranışlar için hangi özelliklerin ilgili olduğu öğrenilmiştir. Küre, silindir ve kutu şeklindeki nesnelerin olduğu bir ortamın geçilebilirliği $\%$ 94 oranında başarı ile tahmin edilebilmiştir. Ayrıca, robotun hesaplanan ortam özelliklerinin sadece $\%$ 1.1'ini kullanacak geçilebilirliği algıladığı gözlenmiştir. Bu da tarama zamanında $\%$ 76.6, özellik hesaplama zamanında ise $\%$ 81'lik bir kazanç sağlamaktadır. Robot daha sonrasında etrafı duvarlarla çevrili kalabalık bir ortamda test edilmiştir. Robot, ortamın sağlarlıklarına göre davranışlarını seçerek ve uygulayarak, bu ortamı başarılı bir şekilde dolaşabilmiştir. Son deney kümesinde ise, öğrenilmiş sağlarlık tabanlı yöntem gerçek dünyada, Kurt3D isimli robot kullanılarak kısmen doğrulanmıştır.

Anahtar Kelimeler: Sağlarlık, Robotbilim, Geçilebilirlik, Doğrudan algı

To my parents

ACKNOWLEDGMENTS

In the first place I would like to express my sincere gratitude to my advisor, Erol Şahin for his supervision, advice, and guidance. He gave me the chance to participate in several state-of-the-art projects, work in a superb environment, enrich my growth as a student and researcher, and taste the beers of various countries. I learnt almost everything about research and academic ethics from him. In every sense, none of this work would have been possible without him.

A special thanks goes to Göktürk Üçoluk who greatly enriched my knowledge with his exceptional insights into computer science.

I am grateful to the members of the Kovan lab for creating such an inspiring and pleasant atmosphere. Ali Emre Turgut has been both a teacher and friend for me, listen all the artificial problems that I create during writing of this thesis, and always stay on my side. It was great pleasure to discuss how the affordance concept could be grounded and applied to robotics with Mehmet Remzi Doğar, Maya Çakmak, and Selda Eren. I'd like to also acknowledge Hande Çelikkanat, Levent Bayındır and Fatih Gökçe for their moral support. Onur Soysal and Erkin Bahçeci deserves special credit for their constructive role within this lab and my studies. I hope to keep up our collaboration in the future.

My sincere thanks go to my friend Emre Akbaş who has always been the first person to help whatever the matter is. I'd like to acknowledge The Big Bros for their suggestions on perception related issues. I am also indebted to Ayça Müge Sevinç for her very valuable comments that saved me from making a catastrophic defense presentation. I am thankful to Prof. Volkan Atalay for providing me his valuable insights on the classification techniques. I also acknowledge Prof. Sibel Tarı and Prof. Osman Parlaktuna for their suggestions on the final draft of this thesis.

This work was partially funded by the MACS project, a Future and Emerging Technologies project (IST-FET) of the European Community, under grant IST-2003-2.3.2.4.

I am grateful to my little brother for he never once complained about how frequently I call him to pick me up in the nights.

Words (especially the English ones) fail me to express my feelings to my dear love Berfin. She was too kind, patient and lovely as she had always been. Every time I found myself on the verge of a breakdown, she gave me the strength to go on. Without her, I'm not really sure if I could have kept my faith to myself, and completed this work. Berfin, thank you for believing in me, and thank you for being there. . .

Finally, I am deeply and forever indebted to my parents for their love, support and encouragement throughout my entire life. I dedicate this thesis to Zehra and Muharrem Uğur.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Do we perceive all the qualities of the environment to accomplish a simple task like walking around? Do we detect the objects in our path, distinguish all their properties, and only then infer whether we can go over them or not? Do we think"there is 80-cm high wooden object with four legs ahead, it should be a table, therefore I cannot walk over a it" or"this circular gray object towards my right is a stone, and I know that the stones that are smaller than my leg length can be walked over, thus I can safely walk over it"?

In 1970's classical theories of perception in Psychology suggested that perception is a generic information processing system that generates a model of the world using sensory inputs. They claimed that actions, which rely on the world knowledge, would use these generic world models, and would require an additional mental inference process to extract the necessary knowledge.

J.J. Gibson, one of the most influential figures in the field of psychology, claimed that this traditional view of perceptual processing is invalid, and proposed a radically different perspective to the perception problem. According to him [1], the inputs that come from the passive sensors are not processed as described in traditional theories. Instead, he claimed, the required information is directly picked up from the environment, without any intermediate step. Such a *direct perception* is only possible with a richer input concept and specially attuned detectors in the perception system. Thus, instead of relying on the two-dimensional image that is sensed by the eye, the information is perceived over various structures and the perceptual system is able to detect the relevant characteristics during *information pickup*.

J.J. Gibson coined the term *affordances* to describe the directly perceivable action possibilities that environment offers to the animals. In his words:

> "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."[1].

A stone offers throw-ability affordance and a stair provides the climbability affordance if the observing agent possess the capabilities to perform the correspond actions. The concept of affordances is deliberately defined over the animal-environment ecological system, thus it includes the properties of both the animal and the environment. As a result, same object or environmental situation affords different actions for different agents. For example, a table which does not offer *traversibility* affordance for an adult may offer *traversibility* to a crawling-infant.

Although J.J. Gibson studied only human visual perception, and described the characteristics of the affordances concepts only over visual variables and examples, the theory provides valuable insights for the field of autonomous robotics as stated in [2].

The work reported in this thesis is carried out within the MACS (Multi-Sensory Autonomous Cognitive Systems interacting with dynamic environments for perceiving and using affordances) project [1], which specifically aims to "explore and exploit the concept of affordances for the design and implementation of autonomous mobile robots acting goal-directedly in a dynamic everyday environment." In order to attain such an objective, a completely new control architecture is being designed to fully utilize affordances in a goal-oriented perspective. As a result, "by interfacing perception and action in terms of affordances, a new way for reasoning and learning will be provided to connect with the reactive robot".

Traversibility is a fundamental affordance for autonomous robots, since most actions depend on their mobility. The traversibility problem becomes very interesting case for studying affordances when one does not limit himself/herself with simple obstacle avoidance. The classic approach to traversibility treats all objects around as obstacles, where the robot tries to avoid making any physical contact with the environment, and only heads for open-spaces to traverse. In general, the proximity sensors are employed to detect whether there is an object or not. When such ap-

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

proaches are used, the robot's response would be same whether it encounters with a wall that is unpenetrable or a balloon that can be just pushed aside without any damage. Although such modifications can always be made to the classical approach, the traversibility affordance depends highly on the interaction of the robot with its environment and is difficult to manually implement. A stair may be traversible for an hexapod robot, while it may not afford traversibility for a wheeled one. Likewise, a small stone might afford the traversibility for a wheeled robot, whereas it may afford non-traversibility for the indoor version of the same robot since the wheel structures are different. Therefore, each different robot should be hand-coded independently by considering the dynamics between it and the objects in the environment. A method that can automatically learn the traversibility affordance from its interactions with the world through learning is an important problem that needs to be tackled.

In this thesis, we studied how physical affordances of the environment, such as traversibility for a mobile robot, can be learned. In particular, we studied how the physical properties of the environment, as acquired from range images obtained from a 3D laser scanner mounted on a mobile robot platform, can specify the traversibility affordance.

In the rest of this thesis, first the theory of affordances will be discussed in detail and its applications in various fields will be provided. In Chapters 3 and 4, the mobile robot platform and its physics based simulation environment will be given. Next, the proposed affordance-based perception, learning and control of the robot is presented in Chapter 5. Lastly, experimental results are reported and discussed in Chapter 6.

CHAPTER 2

THE THEORY OF AFFORDANCES

> "Each thing says what it is . . . a fruit says 'Eat me'; water says 'Drink me'; thunder says 'Fear me'; and woman says 'Love me' "
>
> — Kurt Koffka

Originally coined by J.J. Gibson, the concept of *affordance* has been one of most elusive yet confusing concepts originated in Psychology, influencing fields from Robotics to Cognitive Science. J.J. Gibson argued that, what we perceive from the environment, when acting upon it, is not each tiny bit of information that we are able to sense. Instead, he claimed, we directly perceive what the environment affords or what it offers for our particular action. For example, if we need a rest, we look for a surface which provides support for sitting on it, not more. This view was proposed in reaction to the traditional view, which asserts that, "we perceive the objects as objects, and extract their meanings as they are stored in our mind'. J.J.Gibson argued that the affordances (or meanings) of objects are *directly perceivable* without any recognition or reasoning stage.

This chapter reviews the concept of affordances, its interpretations and applications in different areas. In the next section, the evolution of the term in Gibson's thinking, is reviewed. Then, the discussions in the Ecological Psychology community will be provided in Section 2.2, and the experiments that are done to study the mechanisms of affordances in the same community will be described. In the last section, affordance-related studies in autonomous robotics are reviewed.

2.1 The Evolution of the Theory

J.J. Gibson incrementally constructed his theory of affordances starting from the *valence* term of Gestalt psychology, which coarsely refers to some type of emotional response based on the stimulus received. For example in J.J. Gibson's words, "The field of safe travel has a positive valence" for an automobile-driving agent. As summarized in Jones's article [3], J.J. Gibson, in his very early publications, is talking about positive or negative inherent meanings of objects, the reciprocal relation between environment and agent during perception, and the existence and effect of environmental proportions on animal actions. But all of these concepts, which will become the main components of the affordances theory, are formulated very abstractly in these publications.

An important stage in development of affordances, although the term not appear yet, corresponds to the years of World War II [4], where J.J. Gibson was assigned with the task to evaluate the performance of pilots and other members of air crew. While all his colleagues tried to test the depth and distance perceptual abilities with standard methods mostly in stationary conditions, he questioned the value of such tests for an agent moving in high speeds. As a result, he concentrated on "the nature of information for perceiving the layout in motion and events occurring over time", and studied on the optical variables, which are used to directly perceive the affordances of the environment [1].

The concept of affordances first appears in his 1966 book [5], and is further refined in his later (and unfortunately last) book in 1979 [1]. Although J.J. Gibson could not finish the formulation of the theory, in his second book, he devoted a complete chapter to the description of the concept, laying out the fundamental aspects of affordances. His most frequently quoted definition of affordances is:

> 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)

As Gibson had put it, affordances do not belong to the objective environment,

[1] For example, *optical center of expansion* is identified as an optical variable, which is an indicator of the direction of a glide, and a means of seeing whether the present line of flight is correct or not.

or to the subjective world of the animal. They belong to ecological system that the animal and its environment forms. Therefore, in the perception of affordances one does not only perceive things of the environment, but perceive some combination of features of the environment and of himself.

In order to understand the background where the concept of affordances were born, and how the concept of affordances radically challenges the existing views, one can read J.J. Gibson's writing:

> ...Orthodox psychology asserts that we perceive objects insofar as we discriminate their properties and qualities. ...But what I now suggest that what we perceive when we look at objects are their affordances, not their qualities. We can discriminate the dimensions of difference if required to do so in an experiment, but what the object affords us is what we normally pay attention to.(J.J. Gibson, 1979/1986, p. 134)

Thus, we do not *i)* perceive all the properties of an object, *ii)* classify these properties into abstract objects, and *iii)* infer how these objects could be employed in certain circumstances. Instead, we perceive the *invariant combination of variables*, defined as affordances, and utilize them without use of any object recognition or labeling stage [2].

According to Gibson, perception of affordances also entails the economical usage of perceptual resources. For example, in order to throw a stone to an approaching dog, we try to find an object, which is large enough to hurt the dog, but not so large, in order to be graspable. But the color of the stone, or even the abstract type of the object have no importance in such a situation. We might also use our mobile phone as well, if it includes the specific combination of features for "throwability" and "hurtability".

> An affordance is an invariant combination of variables, and one might guess that it is easier to perceive such an invariant unit than it is to perceive all the variables separately. It is never necessary to distinguish all the features of an object, and in fact, it would be impossible to do so. Perception is economical.

[2] Invariance is defined as "persistence under change" in broad terms. J.J. Gibson mentioned the concept in many contexts through his book and devoted one section in Appendices for it. These invariants correspond to the properties which remain constant under various transformations, ie. invariants of optical structure under changing illumination conditions or under change of observation point. Although J.J. Gibson did not explicitly provide what these invariances are, he gave some clues on the perception and usage of them. "...There must be invariants for perceiving the surfaces, their relative layout, and their relative reflectances. They are not yet known, but they certainly involve ratios of intensity and color among parts of the array."(J.J. Gibson, 1979/1986, p. 310)
Likewise, J.J. Gibson is not clear about what he exactly meant by the term *variable*, and what is the relation between *object properties* and *variables*.

The discussions on the perception of object affordances naturally includes some philosophical consequences on the perception of object meanings.

> The theory of affordances rescues us from the philosophical muddle of assuming fixed classes of objects, each defined by it common features and then given a name. . . . You do not have to classify and label things in order to perceive what they afford. (J.J. Gibson, 1979/1986, p. 134)

When he includes the "object" term into his discussions, J.J. Gibson is not really talking on the objects that are defined with the names given by people, but instead with their concrete existence without use of any label:

> . . . it refers only to persisting substance with a closed or nearly closed surface and can either be detached or attached. I always refer to a "concrete" object, not an "abstract" one. (J.J. Gibson, 1979/1986, p. 39)

After J.J. Gibson, discussions on the concept of affordance and on its place in Ecological Psychology have continued. Also attempts to formalize the concept has been made, since it had an ambiguous description as Gibson had left it. In the next section, some of the most recent attempts from the EC community will be provided.

2.2 Attempts to clarify and/or extend theory of affordances

There has been a vast amount of discussion on the exact meaning of affordance, the contents of the term in the community of ecological psychology. The discussions have been intensified on the gaps of J.J. Gibson's description. Jones, in 2002, organized a symposium pertaining the topic of affordances, and a special issue in the Journal of Ecological Psychology was published based on the presented ideas during this meeting. In this symposium, especially two of the presented ideas include more concrete definitions of the concept.

In Stoffregen's formal definition [6], affordance is described as a higher order property, $h = p/q$, which is a (special) relation between properties of the environment (p), and properties of the animal (q). Based on this definition, the animal perception does not only include perception of the environment, but it also embodies the perception of self. Additionally, these two categorically different properties are integrated. Stoffregen asserts that, direct perception is not only fed from optic or acoustic array, but from **global array**, which includes self-information like somatosensory arrays.

Anthony Chemero, in [7], also tries to clarify the theory, by explicitly defining the affordances as "the relations between the animal's and environment properties". This is a similar definition with Stoffregen's. From animal's side, this relation includes the abilities of the self, and the physical properties like body dimensions. The effect of geometrical dimensions of bodies and parts of the environment are the most studied aspect of the affordances. The next section includes experiments, which study the affect of such dimensions on human affordance perception.

2.3 Experiments in Ecological Psychology

Following the formulation of the theory of affordances, Ecological Psychology community started to conduct experiments in order to verify that people are able to perceive the affordances of the environment and to understand the mechanisms underlying it. We must mention that, although the number of these experiments is quite high, the diversity is narrow and the research can not go beyond experimenting on already tested ideas in different contexts.

Since affordances can be roughly defined as the properties of the environment taken relative to the animal acting in it, there have been efforts to show that the ratio between an environmental property and a bodily property of the animal have consequences for behavior. This ratio must also be perceivable, so that the animal is aware of this measure which, in a way, determines its behavior's success. Warren's stair-climbing experiments (1984) [8] have generally been accepted as a seminal work on the analysis of affordances, constituting a baseline for later experiments which seek to understand this affordance based perception. In these studies Warren showed that animals perceive their environment in terms of *intrinsic* or *body-scaled* metrics, not in absolute or global dimensions. So, my judgment of whether I can climb a stair step is not determined by the global dimension of the height of the stair step, but by its ratio to my leg-length. These refer to functionally relevant variables[3] in the animal-environment system. These variables and some *constant ratios* of them help the animal to determine whether an action is afforded or not[4].

[3] Different from J.J. Gibson, "variables" term here, directly corresponds to some physical properties of the animal and objects. In Gibson's terminology, the concept of "variable" refers to some "high-order" information embedded in the light.

[4] These ratios which determined whether an action was afforded or not were called the *critical points* by Warren; also, the ratios which determined whether an action can be performed with minimum energy consumption and maximum ease were called the *optimal points*

This *intrinsic* nature of human perception is further elaborated in Warren and Whangs's *walking through aperture* [9]. As noted by Warren, "affordances could be perceived on the basis of *intrinsic information*, which specifies environmental dimensions relative to the dimensions of the observer in units of some body-scaled or, more generally, action-scaled metric". For example, eyeheight[5] is such an intrinsic metric, which is believed to affect the perception of geometric dimensions such as size and distance. In other words, size and distance of objects are directly perceived as they are already scaled with reference to observer's eyeheight. To understand the effect of this metric on width and height of objects, "perceived eyeheight" information is changed in these studies. In the experiments, when observers could not recognize these changes, they made contradicting judgments for objects with constant absolute values. Mark explicitly studied the factor of awareness in the change of this metric in [10]. In the experiments, human subjects wore $10cm$ blocks, and were asked to determine whether the heights of various surfaces afforded sitting or climbing. Since the subjects could recognize their eyeheight is changed, they are successfully able to perceive correct critical points. The results show that perception is based on body-scaled information, and humans are able to adapt themselves when they recognize any change in this information.

This view of affordance-perception has been studied from many different aspects. Some of these studies[11, 12] criticized former studies because they limited themselves to only one perceptual source, namely visual information. Instead of limiting themselves to visual perception, they studied haptic perception in *infant traversibility of surfaces* and *critical slant judgment for walking on sloped surfaces*. These studies are important because they underlined the fact that all available relevant sensor channels are employed during affordance perception. While in these experiments human subjects were asked to judge whether a certain affordance exists or not in a static environment, Chemero et. al. [13] conducted other experiments, in order to prove that *changes in the layout of affordances* [6] are perceivable in dynamic environments, and found out that the results are compatible with *critical ratio* values. Another important work is Oudejans et. al.'s [14] study of *street-crossing behavior* and perception

[5] In [9], eyeheight is defined as the height at which a person's eyes would pass through the wall while walking and looking straight in a natural and comfortable position

[6] Chemero et. al. categorized events in two groups: physical, and ecological events, where later refer to the changes in the layout of affordances.

of *critical time-gap* for safe crossing. It is shown that not only static properties of the agent, but also his/her dynamic state is important when deciding on his/her actions.

All referred experiments are performed in *one shot* manner, and either the subject is stationary or moving [9], either monocular or binocular vision [15] is employed, either haptic or visual information [11] is used, either the critical or optimal points [8] are determined, either searching for affordance or change in the layout of an affordance [16] is examined. In all cases, observers are asked to judge if the environment affords a particular action or not, and this is isolated from all other cognitive processes. Although the study on "detection of change in the layout of affordances" includes some discussions on human decision mechanisms, it does not provide any qualitative expansion, and limits itself to the iterative question of "whether this environment affords X or not?".

Despite their narrow scope, we can extract some concrete ideas from the experiments in our own application field. First of all, it is shown that both haptic and visual information is employed in the perception of environment, and in case of one of the information source is blocked, the remaining one is able to manage the situation. Additionally, there are clues how human perception system copes with contradictory or incorrect information. The results of these experiments may be employed in the design of multi-sensory perception system in a robotic platform.

The emphasis given to *intrinsic* and *action-relevant* measures also includes important ramifications for an affordance-based robotics research, since experiments show that instead of *absolute* or *global* measures, a *body-scaled viewpoint* should be employed in representation of any perceptual process. Additionally, in the experiments, authors were able to calculate the constant, so called π proportions, that depend on specific properties of the animal-environment system. There exists one such ratio per each affordance, and they solely depend on the **functionally relevant variables** of corresponding actions.

An overview of the related experiments shows that they are mostly employed only as test-beds to validate the existing ideas, and their scope is severely restricted by the *perception* of affordances. Other cognitive processes such as learning, high level reasoning and inference mechanisms are simply untouched, and the link between affordances and these higher level processes is not established. In the next section, we will try to close this gap, by presenting some existing studies on learning of affordances, and relation of affordances to high-level perception.

2.4 Cognitive Science

The concept of affordances has also been studied within Cognitive Science. In this section we will review studies on *i)* learning of affordances, and *ii)* relation of affordances to high-level perception.

E. Gibson was one of the few people, who tried to explain the mechanisms of *learning of affordances*. She claimed that[17], J.J. Gibson was not particularly interested in development, and "his concern was with perception" only. As a result, he did not discuss the concept of affordances from a developmental point of view, and only mentioned that affordances are learned in children [1].

Within her research, E. Gibson defines learning as a process of selection *(differentiation)*, not construction from smaller pieces *(association)* [18] in her research on developmental psychology. This point of view entails "discovering *distinctive* features and *invariant* properties of thing and events", while perceiving their affordances. In the experiments performed with infants, exploration is found to be crucial that, even newborns have active probing in order to i) gather environment information, and ii) explore their own capabilities. Furthermore, it is shown that when infants are presented the same information over and over, and then an opportunity is given to choose between that and a new information, they reliably attend the *novel* information. Additionally, as another principle of perceptual learning, organisms use shortcuts like perceiving an object as a *unity* or discovering order in events, in order to benefit from economy in actions and reduction of perceptual information. Many of the results of these experiments already exist in the original affordances theory of J.J. Gibson, however they include more concrete ideas that might be used while designing an affordance based control architecture.

Neisser, in his "Cognition and Reality" book [19], employed the concept of affordances in perception of the object meanings. According to him, J.J. Gibson was right, while stating that meanings of the environment are directly available, and *information is not processed, but it is directly picked up since it is already there (in the light)*. The invariance attuned detectors are used for this purpose. However, he claimed Gibsonian view of affordances of perception was inadequate, since "it says so little about perceiver's contribution to the perception act". Instead, he suggests a perceptual system where a cycling continuous activity over time and space occurs. This cycle "prepares the perceiver to accept certain kinds of information... At each moment the perceiver is constructing anticipations of certain kinds of information, that enable him to accept it (information) as it becomes available." Since every natural object has infinite number of affordances, this cycle could also be employed to prepare the perceiver to search for particular affordances at each moment, and attune specific detectors to perceive these affordances.

According to Neisser, both *constructivist*[7] and *direct* theories of perception should be integrated. As a result, in a later paper [20], he constructed a three-layered perceptual system, whose first and third layers correspond to direct perception and recognition, respectively[8]. While direct perception system is identified by the perception of the local environment, recognition refers to identification of familiar objects and situations. Contrary to J.J. Gibson, who states that affordances of environment are directly perceived, according to Neisser, affordances are also perceived by high level perceptual systems.

J. Norman [21] also "attempts to reconcile the constructivist and ecological approaches,", based on evidences from human dorsal and ventral systems. To do this, he suggests a perceptual system, where two different and interacting visual systems works. While the dorsal system is mainly responsible from pickup of information from light to modulate actions, the ventral system is concerned with high level perceptual tasks, like recognition and identification. Thus, according to Norman, it is straightforward to conclude that "the pickup of affordances can be seen as the prime activity of the dorsal system". He additionally limits the concept of affordances to situations where an action potential exists. To support his two perceptual system idea, he presents examples from a patient, who lacks a ventral system. The patient is able to successfully avoid obstacles around, or insert mails into slots in correct orientation using her dorsal system. However, while performing actions successfully, she is not aware of the objects she is interacting with, thus cannot report them.

J.J. Gibson was aware of the dual-process of perceptual system, while stating

> "The verb to perceive has two meanings, one being that of ordinary usage and the other coming from a puzzle in philosophy and psychology... The

[7] Constructivist theories favor a model construction phase in processing of the sensory input.
[8] The second layer is about inter-personal perception and will not be discussed here.

> two meanings need to be kept separate in the investigation of perceiving. The act of a perceiver and the content of his mind should not be confused."

While Gibson's affordance theory deals with first meaning, he left out the second meaning which has a content of "awareness, or of consciousness" out of his affordance concept.

2.5 Robotics

In this section, a number of related robotic studies and how they utilized the affordances concept will be presented. However, a rough overview of the history of autonomous robotics should be provided to prepare the readers for latter discussions.

When whole history of robotic applications is considered, robot paradigms might be divided into three broad categories, as stated by Murphy in [22]. The oldest approach, *hierarchical paradigm*, where the robot is controlled using AI problem solving techniques, classical AI problem solving techniques, suffer from many problems in real world conditions like frame problem, and they cannot react in a timely manner in dynamic environments with inconsistent and noisy sensing. Since the deficiencies in hierarchical architectures mainly stem from the slow planning module which locates between perception and action modules, in mid-80's, researchers tried to eliminate this heavy component. Influenced from robustness and flexibility in actions, and fast response abilities of animals and other biological systems, *reactive architectures* are emerged as a result. By direct coupling of sensing and action in terms of behaviors, and by means of interactions between concurrent behaviors, very successful real-time robotic controls systems were constructed and applied to various domains [2]. However, the absence of a planning module made it impossible to deal with complex tasks, and limited the application areas critically. Thus, in the very beginning of 90's, third generation architectures emerged from the need of putting planning component back into the control mechanism. *Hybrid architectures* integrate deliberative component with behavioral modules while not degrading the reactive performance, and become standard in modern robot control.

The concept of affordances is highly related to autonomous robot control and influenced studies in this field. The paralellism between the theory of affordances and reactive/behavior-based robotics has already been pointed out(pp 244,[2];[23]).

A similar parallelism also exists with studies carried under the heading of action-oriented perception (pp. 267, [2]). These studies suggested a "qualitative" representation of the environment based on the task/intention at hand, and criticized the classical approach to perception (particularly computer vision) which aimed to recover a metric model of the environment [24].

Many concepts within affordances theory are inherently included in reactive robotics, but they are implemented in these systems in an ad-hoc manner, bringing forth the famous *"art more than science"* argument. Although some of these reactive control systems are explicitly supported by the theory of affordances, since only certain aspects of the original theory as formulated by J.J. Gibson are used, and modern discussions of the concept are not utilized in these studies, they are extremely limited. The use of affordances are restricted to some modules, which are the components of already constructed systems. In this perspective, while a number of the control systems are solely designed to test the ideas on affordances, others do not have an explicit affordance-related component, but implicitly follow an affordances-based methodology.

To the best of our knowledge, existing robotic studies utilize affordances in order to either release or guide their behaviors, but not both. Thus, in this section, we decided to categorize these works based on this distinction.

Affordances for guiding behaviors

Murphy [23] and Duchon et. al. [25] applied affordances to guide the behaviors, and did not consider it as a *behavior selection* mechanism. Since all these studies include valuable discussions about affordances, we will describe them in detail.

Murphy [23] studied affordances in reactive robotics domain, in order to guide tracking and fine positioning behaviors in a number of navigation tasks. Arguing that *"progress in mobile robotics relies on progress in perception"*, Murphy discussed the application of a reactive "affordance-based control", where tasks are accomplished without the use of explicit perceptual models. In general, the discussions on the paper were focused on the advantages and limits of such methodologies in robotics. For example, although a much more simple methodology which relies on direct perception is employed in can collection task, it was found out to work as good as other applications which use heavy feature extraction and inference mechanisms. However, Murphy pointed out that affordances could be applied to only a set of simple problems, where great care should be taken while designing the behaviors. Additionally, she argued about the flexibility of the control systems in the face of novel environment conditions, and how brittle they may become when the robot is expelled from its ecological niche. Many ideas already included in the ecological and developmental psychology literature, such as the ones reviewed in this paper, are simply ignored by Murphy in her discussion of affordance based control architecture.

Like Murphy, Duchon et. al. [25] also studied affordances as a means of handcrafted low level motor control mechanisms based on direct perceptual information. They defined *ecological robotics*, as "the practice of applying ecological principles to the design of a mobile robot", and used a number of optic flow based control laws for implementing obstacle avoidance behavior in a wandering task, and escaping, chasing and docking behaviors in game of tag. They benefited the direct relation between perception and action (direct perception) by mapping perceptual information to control parameters of the behaviors, based on the current goals of the robot. According to Duchon et. al. , some task-specific memory and learning could be incorporated, but no central model should be employed in ecological robotics. This work includes implications on the relation of goal-directed behaviors, decision processes, and affordances, however the practical application is limited to *direct perception* of the environment.

In summary, Murphy, in three case studies, try to incorporate direct perception in certain reactive modules, while hand-crafting all behaviors in advance. A similar precoding is employed in Duchon et. al. 's work, but different from Murphy, other aspects of affordances are discussed, and explicitly embedded into the system, like egocentric view of the world. These works show that these type of controllers could only be employed in reactive modules of an affordance based system, since they have very loose relation with other aspects of the control like behavior modulation or high level inference mechanisms. In the next subsection, we will review the robotic studies, where affordances are used in behavior selection, thus have implications on many other aspects as well.

Affordances for releasing behaviors

This section will review the robotic studies, where affordances are used to select appropriate behaviors based on the perceived environment. Utilizing affordances in such a way seems to better fit to the definition of J.J. Gibson, which states "The affordances of the environment are what it offers the animal, what it provides or furnishes,...". Additionally, as described in Section 2.3, experiments in ecological psychology examine the role of affordances in releasing human behaviors, not in guiding them.

Cooper and Glasspool enhanced their existing action selection methodology, by utilizing the learned affordances of the environment. In their system [26], actions are organized in an interactive activation network, where both environment conditions and excitation of other nodes in the network determine the activation of each action. Through unguided exploration trials, robot learns which actions are afforded in which environmental situations using reinforcement learning. Although this approach, which learns environment/action associations, includes the basics of learning of affordances, the cognitive modelling environment, where experiments are conducted, severely limits the value of this study. This environment provides objects, actions, and interactions, which are all defined over symbolic representations. For example, when any object is inserted into the environment, its properties are inserted in it as well, like its shape and color, and the robot is able to acquire this information exactly in world reference frame. Likewise, the preconditions of actions and the consequences of them are defined precisely, which is unrealistic in real world situations. Additionally, the relation between their framework and the concept of affordances is loose since in Gibsonian view, "the concept of affordances went hand in hand with that of direct perception." Contrary to this view, their method relies on a heavy modelling process of the environment, where state of the world transduces to internal representations.

Affordances are utilized in behavior selection, also by Cos-Aguilera et. al. [27, 28] in a motivation-driven robotic architecture. Different from previous work, a kinematic simulation environment is used, and furthermore, not only perceived object features, but also low level simulated sensor data are employed while learning and then detecting the affordances of the immediate environment. The central tenet of their approach is that, the behavior is determined based on the motivations of the agent, however if the environment does not afford that action, robot should not attempt it, and seek for objects that provide corresponding affordance. In their former study, high-level features are extracted from objects, as size and shape information, and the mapping from these features to the actions are learned through the back-propagation learning method, in a two-layer feed-forward neural network. In the latter study, they tried to obtain the *invariancy*, which is an important concept in the theory of affordances, in the feature space by clustering the sensor data in an unsupervised fashion. Then, by interacting with the environment and carrying out each behavior in various objects, the robot learns affordances of clusters for each action in an incremental fashion. Though the navigation task suits well with the direct usage of raw sensor data, how this approach could be generalized to other domains is not clear. Additionally, searching invariant data solely within feature array without any reference to behavior of the robot has some drawbacks, ie. different affordances are related to different invariances, thus one such clustering cannot be applied to more than one action. MacDorman also spent special effort to *explicitly* extract invariant information from perceptual features and to use them while detecting the affordances of navigable environments [29]. To do this, he proposed a method where i) segmentation is performed on potential sources, ii) signatures of the segmented images are extracted as potential features for invariance, and iii) affordance categories are found by "statistically filtering out all signature values except those that tend not to vary among signatures of same affordance category but vary among signatures of different affordance categories". In this way, he used both feature array and the consequences of actions while extracting the invariant data, which persists constant within same affordance category.

Affordance representation

In [30], Fitzgerald et. al. set forth to construct a system, which elicits an action, without any object recognition stage, but instead by using some (visual) characteristics of the objects, inline with the affordance concept. The added value of this work is the explicit representation (although not formally defined) of affordances of objects, incorporating perception and action in this representation. During the experiments, the arm of a upper torso of humanoid robot, Cog, is used to poke the objects ahead.

Experimental setup is limited to four actions (pull in, side tap, push away, back tap) and four different objects (bottle, cube, car, ball). The aim is to first learn the objects' unique motion signatures for certain actions, and then select appropriate actions to make a given object roll in a certain direction. Since they use constant objects which are categorically different, they decided to use objects' orientation information in their representation. This feature might be used to distinguish the *roll-ability* affordance of "these" objects, ie. a bottle rolls perpendicular to its principle axis of inertia, where a car's movement would be parallel to this axis. As a result, the motion signature of an <object,action> pair is defined as the *angle of movement relative to its principle axis of inertia*. During test phase, when a particular object (in a certain orientation) is required to roll in a certain direction, the motion signatures of <object,action> pairs are searched, and an appropriate action is selected based on previous experience. Two points are important here: First, in affordance perception, only functionally relevant features are used. Since the objects are different from each other, the principle axis of inertia is important, and motion signature of various objects are differently related to this feature. The other point is that, although the formal representation of affordance is not explicitly done, it can be derived that, actions are embedded in affordance representation, by means of angle of movement. Although, actions and functionally relevant features are successfully included in affordance representation, all components are too task-specific. They don't provide any general framework, where these components are extracted automatically, thus the approach could not be applied to different domains.

Implications on affordance utilization in planning

As described above, affordances could be utilized in releasing and guiding the behaviors, based on the immediate environment. Thus, the concept have relations to both preconditions and consequences of the actions executed, having the potential of connecting and sequencing different behaviors to acquire a certain goal. In many robotic studies that deal with affordances, actions are either sequenced based on other existing mechanisms (ie. activation networks, robot's homeostatic systems), or they are not sequenced at all. Similar to ecological psychology field, relation of affordances with high level processes like planning is very loose. In this respect, MacDorman [29] studied low level learning of navigability affordances of the environment, and used the learned relation between actions, its preconditions and consequences, sensorimotor mapping in short, in planning of a navigation task. However, his application could hardly be generalized to other domains since planning was based on low level sensorimotor mapping.

CHAPTER 3

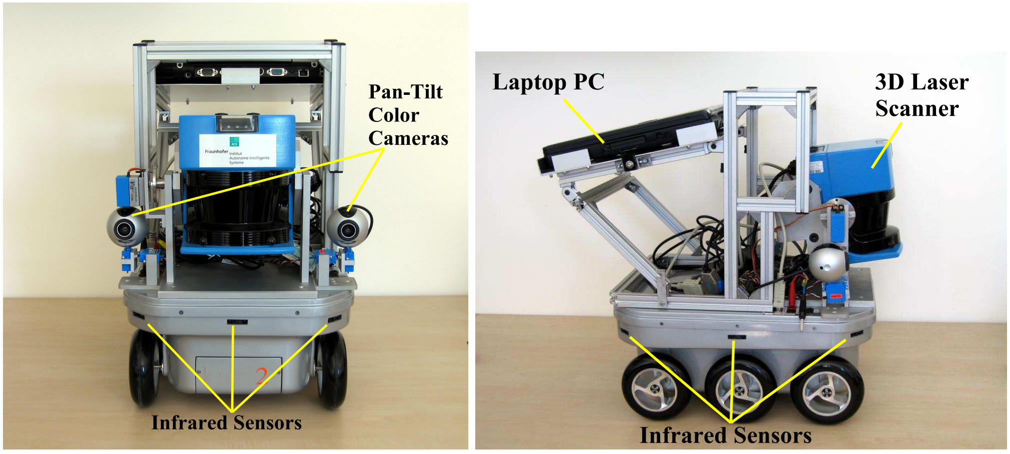
Kurt3D MOBILE ROBOT PLATFORM

A mobile robotic platform, Kurt3D, (Figure 3.1) and its physics based simulated model is used in this work. This chapter provides the specifications of the robot and its utilized components. The detailed description of its simulator will be given in the next chapter. Kurt3D robot platform is an extended version of Kurt 2, which was mainly designed for sewerage inspection task, and commercially available from KTO - Kommunikation und Technologietransfer Odenthal [1]. As the **3D** tag in its name implies, this robot is equipped with a 3D laser range finder, which is a product of Fraunhofer Institut für Autonome Intelligente Systeme[2].

3.1 The Robot Body

With a size of 45 cm (length) x 33 cm (width) x 47 cm (height) and a weight of 22.6 kg, robot's elongated rectangular base body carries several sensors and actuators on top of it. The robot's locomotion system is composed of six identical wheels, with diameters of 11 cm, three on the left, and three on the right sides of the base body. The wheels, which reside on the same side of the body, are connected by a toothed belt drive, resulting in a differential drive system. Besides its fundamental sensor, laser scanner, Kurt3D is equipped with a number of additional sensor modalities, including two pan-tilt color cameras, eight infrared proximity sensors, and two tilt sensors. While its 3D laser scanner is principally utilized to construct the 3D model of the environment in various applications, cameras might be used both in object recognition and to guide the driver of the robot in teleoperation tasks. Furthermore, laser scanner readings could be fused with camera data, to acquire a rich model of the

[1] http://www.kurt2.de/
[2] http://www.ais.fraunhofer.de/ARC/kurt3D/



(a) Front view

(b) Side view

Figure 3.1: The actuator and sensor modalities of the Kurt3D mobile robot platform are illustrated.

environment.

Although several sensors are provided with Kurt3D robot platform, since only 3D laser scanner is used in perception of the traversibility affordance of the environment, only this sensor will be described in the next section. [31] includes detailed specifications of all sensor modalities.

3.2 2D Laser Scanner

Since the information about shape and distance of objects in the environment is crucial for most of the applications in mobile robotics, laser scanners have become popular in the community. They are used in a broad range of applications, from obstacle avoidance [32] and feature extraction [33], to map building [34] and self localization [35]. They are also suitable for outdoor robotic tasks, since some of them have a range up to 100 meters [Cyrax 2500], and roughness of the terrain is explicitly available.

When compared with other sensor modalities, laser scanners provide more robust, accurate, dense, and reliable data. For example, unlike camera sensors, they are robust in the face of illumination changes. Moreover, while many proximity sensors are highly sensitive to material characteristics and surface orientations (ie. sonar sensor), these properties have negligible effect on the range image obtained from laser scanners. As a disadvantage, although laser scanners are quite fast in 2D mode, they

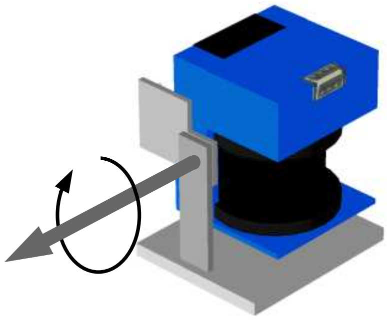


Figure 3.2: The rotating 2D laser scanner.

might become too slow for fast real-time robotic systems, when they are used in 3D mode (because of the slow vertical rotation). The details of 2D and 3D scanning will be provided in the next section.

3.3 3D Laser Scanner

In this section, how laser beams are utilized in distance measurements will be described, and then operation principles of laser scanners will be presented.

A broad range of scanning hardwares and methods are available in the market, to obtain the 3D range map of the environment. While some of them are able to fire their laser beams in 3D directly, by swapping the area of a cone [3], their lower cost alternatives utilize one or more 2D laser scanners in various setups. For example in [36], two 2D laser scanners, one mounted horizontally, and other vertically, are used, and their readings are combined to obtain 3D data. In our case, the 3D laser scanner of the Kurt3D robot platform is based on SICK LMS 200 2D laser scanner, which is mounted on the robot with a standard RC-servo motor. Having a horizontal rotation axis, the 2D laser scanner pitches up and down (Figure 3.2), scanning the 2D slice of the environment in each pitch angle.

The range of the objects are calculated by measuring the time interval between an emitted laser pulse and reception of the reflected pulse. A laser beam is emitted from the sensor, when it meets with an obstacle, it is reflected back, and received by the photo detector. The time between emission and reception, time-of-flight (TOF)

[3] `http://www.neigps.com/products_cy_2500.php` (Last accessed on August 28, 2006)

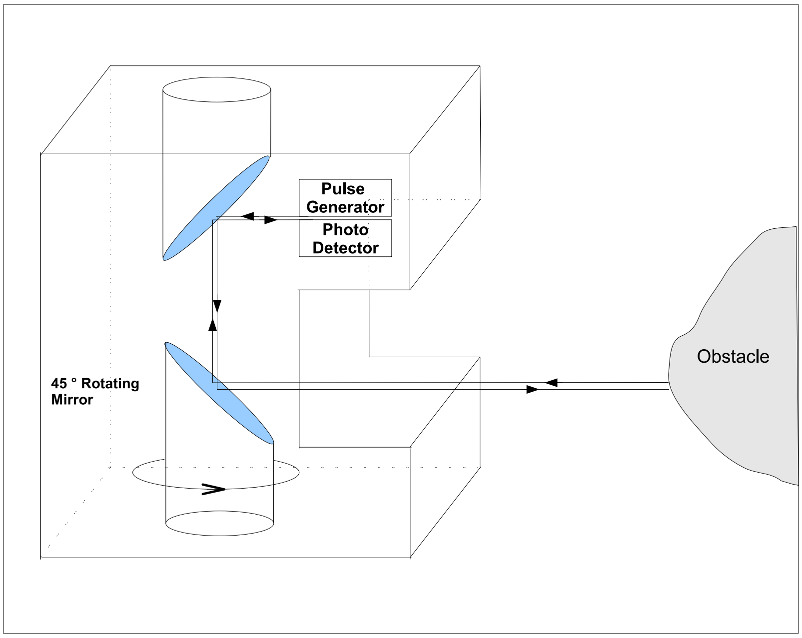


Figure 3.3: The operation principle of the 2D laser scanners. The mirror, which rotates (with 4500 rpm) around its vertical axis, allows the laser beams to be reflected in different directions, and and its $45°$ constant slope with the horizontal plane enforces the laser beams to be transmitted along the same scanning plane.

of the laser beam in other words, gives an estimate on the distance to the obstacle. However in this way, the range information in only one direction is obtained. Thus, in order to acquire the measurements in various angles (directions), a rotating mirror is utilized, which redirects the beams in successive angles. In this way, a rotating laser beam in a plane produces the 2D slice of the environment, a range value for each angle in this plane. This plane is horizontal relative to the scanner body, when SICK LMS 200 is considered. Figure 3.3 demonstrates the operation principle of the 2D scanner.

The SICK LMS 2D laser scanner has a horizontal range of $180°$, with resolution choices of $0.25°$, $0.50°$, and $1.00°$, resulting in 721 range readings at maximum in a two-dimensional scan. Using the pitch mechanism that is described above, in its 3D mode, the scanner is able to sweep a vertical range of $\pm 82.8°$, with a resolution $0.23°$ at maximum. Figure 3.4 shows a range image of an outdoor 3D scan, where the grayness corresponds to the distances of the objects.

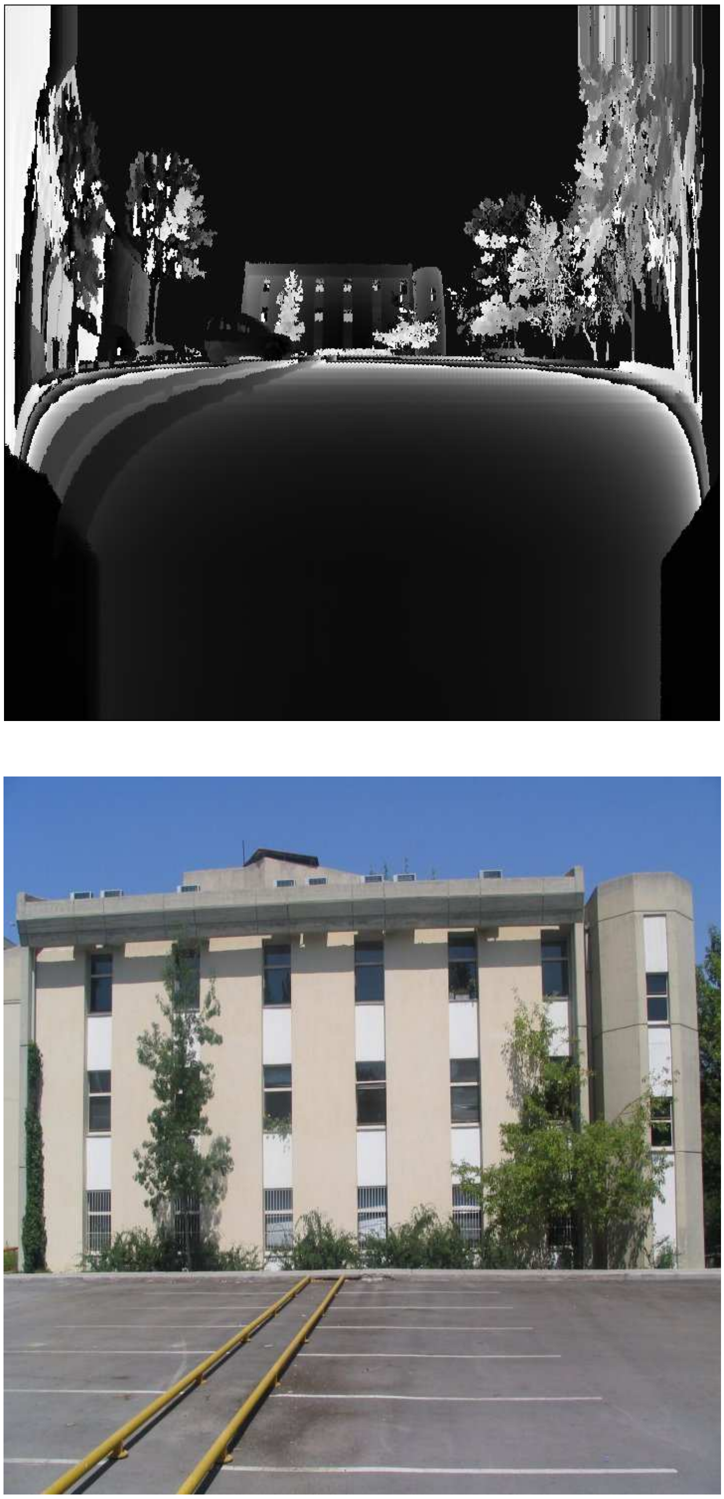


Figure 3.4: The range image obtained from 3D scan, and the photograph of the corresponding environment.

CHAPTER 4

MACSim: A PHYSICS-BASED KURT3D SIMULATOR

MACSim is a high fidelity simulation environment that models the Kurt3D robotic platform and its environment. Built on top of a commercial quality open-source engine, ODE[1] (Open Dynamics Engine) , MACSim accurately simulates the objects, robot parts, and their dynamics in a 3D world. MACSim additionally benefits from the functionalities of KODEX (Kovan ODE eXtension)[37], which extends the capabilities of ODE in many aspects.

Simulation model provided in MACSim matches closely to the real Kurt3D robot in many aspects. Based on their physical properties, such as mass, size, and center of mass, all parts that constitute the robot are modelled as rigid bodies. Later, junction locations of these components are measured, and they are assembled with appropriate joints to acquire the complete simulated robot. In order to simulate different actuators of the robot, such as wheel systems or camera servo motors, the joints are virtually constrained and motorized with the parameters obtained from real robot.

Realistic sensor modelling is also very crucial, since robot actions and control relies on its perception of the world. While ODE provides excellent support for modelling rigid body dynamics [2] based on laws of physics, similar to many low level engines, virtual sensors are not explicitly supported. For example, there is no ready-to-use acoustic signal or infrared beam that could be sent or received. Kurt3D is equipped with three major sensor modalities, as described in Chapter 3, and all of these sensors are simulated in MACSim. For laser scanner and infrared proximity sensors, ODE's ray geometry and collision detection routines are utilized, and ray

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

[2] The flexible or deformable bodies, like ropes and cloths are not supported in ODE. "Dynamics" is the key word here, which refers to the modelling based on physics laws. Movements of the objects and the collisions between them are computed relying on physical properties like mass , center of mass, inertia, gravitational force, frictions, etc.

intersection method is used. For color cameras, OpenGL's backbuffer data is employed. Since real world images and the data acquired from OpenGL rendering are qualitatively different, KODEX's shading, shadow rendering, and easy-to-use texture rendering features are utilized. Moreover, in order to close the gap between reality and simulation, sensor and actuator parameters are calibrated, based on the "same setup experiments" in virtual and real worlds.

It is very important to support easy maintenance of the whole simulation environment, such as creating a virtual world, accessing and changing any simulation element during simulation, etcetera. For these reasons, MACSim provides a very elegant interface for simulation supervisor functions. First, an artificial 3D world with a robot inside can easily be created by editing configuration files[3], which are loaded using KODEX's *file loading module*. Through this, users are able to construct the robot and environment according to their requirements, in a modular fashion. Moreover, both simple and complex objects can be inserted into or deleted from the virtual world during execution of the simulator. Using MACSim's supervisor interface functions, users are able to access all information about the objects, their sizes, geometries, or colors, and it is possible to change these properties in run-time. In summary, all these interface functions are extremely useful, for especially training experiments, where the environment should be created and changed dynamically according to the needs. Note that, various sample ready-to-use objects, from simple ones like boxes, and cola cans, to more complex ones like ramps, doors, and tables are included in simulator software as XML configuration files.

One other important characteristic of MACSim is that it shares same interface functions with real Kurt3D robot platform. This enables easy and very fast code migration from simulation to real world, and vice versa.

4.1 Overview of the Existing Simulation Environments

Table 4.1 presents an overview of the comparison among the software packages that are either designed as general purpose modelling tools, or for specifically robotic applications. Since the table intends to give a general and rough overview, all comparisons are performed on *Yes*/*No* basis. For example, although they have same *No*

[3] Configuration files are kept ing XML file structure

label, the cost of Karma physics engine, when used with UsarSim, would be much cheaper than other commercial products.

First major distinction appears on their design considerations, and how they model the world and time. Some software packages, that simulate the world only based on kinematic laws[4] provide models in very low fidelity. When compared with dynamics simulators, they provide very fast simulation environments. Articulated rigid body dynamics on the other hand, is a very crucial requirement for KURT3D simulation environment, in order to model the realistic interactions with world.

The cost of the simulation software is very important as well. Commercial packages generally provide many features, like realistic 3D world models, fast and accurate methods, and elegant graphics. They are specialized for different application environments like games (Havok), industry mechanical design (Adams), or robotics (Webots). Although they are appealing, we decided against their use, since the simulators built using commercial packages, often require renewal of licenses, which limits the use of developed simulators to the duration of the project that it was developed for. Open Dynamics Engine on the other hand, as a free and open-source software, provides 3D dynamic world models, which are compatible with most of the commercial packages, like Karma (of UsarSim) and Vortex. Additionally, many commercial softwares (ie. Webots and many games in the market[5]) use ODE as their core physics engine.

Another criteria is the general structure of the software packages. Some modelling environments are designed as general purpose simulators, and others are robot platform oriented. The construction of KURT3D and its environment requires flexibility, thus general purpose softwares are favored.

Besides their technical properties, the activity level of the developer and user community of the software packages is also very important. Recent open-source softwares (especially ODE with approximately 10 mails per day in its mail-list) are very active, when compared with older ones. The support for commercial ones are mostly based on the licence agreement.

---

[4] In kinematic simulators, objects are massless entities, thus object motions and interactions are poorly modelled. However in dynamics simulators, the physical properties of the world like mass in objects, a gravitational force, the frictions between objects are all modelled. As a result, they provide high fidelity in modelling of motion of the object and their interactions.

[5] `http://www.team6-games.com/,http://www.bloodrayne2.com/`

Table 4.1: Comparison of the existing simulation environments. The robotic centered packages limit their environment with simulated robot platforms and focus on the features of the robot. On the contrary, general purpose platforms are able to simulate both robotic and non-robotic systems, allowing more flexible (and realistic) modelling of environment objects. NA means not applicable for KURT3D modelling. In some simulations (with non-flexible overall architectures), the target robotic domain is so specific, that, it is almost impossible or very hard to implement your own robot control architecture.

	Teambots	**Swarm**	**Stage**	**Gazebo**	**Darwin**	**Sigel**	**Dyna**	**MDP**	**Adams**	**Dymola**	**Havok**	**Karma**	**Vortex**	**Webots**	**ODE**
Rigid body dynamics	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Free	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No	Yes
Active	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Robotic centered	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No
Sensor support	Yes	Yes	Yes	Yes	No	No	No	No	NA	NA	No	No	No	Yes	No
Flexible architecture	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Figure 4.1: The simulated Kurt3D robotic platform.

Either 2-D or 3-D, almost all robotic-centered architectures provide sensor support, based on robot platforms, they already simulated. Gazebo, Webots, and UsarSim are among these platforms, which models the world with high fidelity. They use the exact data of the objects around, and simulate the sensors accordingly. Likewise, we will extract the environment data from ODE, create our own virtual sensor models, and perform experiments for calibration of them with real ones.

Based on all the discussions above, there remains two feasible choices, *general purpose* ODE, and *robotic-centered* Gazebo. Although Gazebo provides many re-usable features that could be employed to simulate KURT3D, like laser scanners or cameras, or data visualization tools, it was in version 0.4 and was not mature enough at the start of the development of MACSim.

4.2 Simulated Model of the Kurt3D Platform

The hardware overview of the Kurt3D robot platform is given in Chapter 3. All sensor and actuator modalities are simulated in MACSim, as demonstrated in Figure

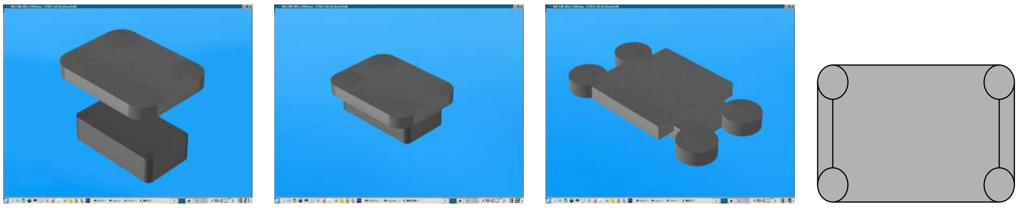
4.1. However, since solely 3D laser scanner is employed for perception of the world in traversibility experiments, other sensors and how they are modelled will not be described in this section. Likewise, the robot crane arm is not used, and thus modelling of this actuator system is not in the scope of this document. In this section, the construction of the simulated robot body, the modelling of the drive system, and the modelling of the 3D laser scanner will be described.

4.2.1 Robot Body

The robot body is the most important part of the robot since all sensors and actuators are placed to different locations on it. Additionally, because the interactions with the environment is done through this part, and the dynamics of the complete robot is mostly based on it (moment of inertia, mass etc.), physical modelling of this component has crucial importance.

As shown in Figures 4.2 (a) and (b), the robot body is roughly composed of two rectangular prisms, which are stacked together, and combined with a fixed and rigid joint. From now on, we will call the upper portion as the *bulk*, and the lower one as the *base*. While the base is responsible from the working of the locomotion system (since the wheels are attached to it), all other modalities reside on the bulk. Thus, both cameras, the laser scanner, the metallic frame which carries the notebook PC, and the simulated crane arm as well, are all mounted on top of the *bulk*. It also houses the infrared proximity sensors at its periphery.

The bulk and the base have similar geometrical shapes, and they are modelled as a rectangular prisms with rounded corners. Modelling these components with simple, ready-to-use rectangular prisms with sharp corners is not sufficient, since this would decrease the fidelity of the simulated interactions with the environment. Thus, these portions are simulated as composite bodies, which are composed of several basic geometries, two rectangles and four cylinders in our case. As shown in Figures 4.2 (c) and (d), two overlapping rectangular prisms with the same height but with different width and depths are combined in a plus shape, and then the cylinders are placed to the empty corners, resulting in a rectangular prism with smooth corners. Although the corners of the real Kurt3D's rectangular body is not exactly spherical shaped, we did not consider other alternatives such as meshes, since they are computationally too expensive.



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

Figure 4.2: The construction of simulated robot body. In (a) and (b), modeling of the Kurt3D base body by combining upper and lower parts are demonstrated. (c) and (d) illustrates the construction of the these elongated prisms, where former one shows top view of the sketch of this construction, and latter one gives a dismantled view in MACSim's 3D world.

4.2.2 Wheel Systems

Kurt3D is a differential drive robotic platform, with three wheels on the left and three on the right side. All wheels have identical shape and size, as discussed in detail in Section 3.1. The wheels are modeled using ODE's basic collision primitive, cylinder, and they are connected to the *base* body using motorized hinge joints[6]. Figure 4.3 shows a snapshot where textured wheels are attached to the robot body.

The real wheel is a solid cylinder with hollow parts, and surrounded by the shell of a torus. Since ODE does not support torus geometrical primitive, the wheel is modelled by simply a cylinder. Using other alternatives such as meshes would slow down the simulation unnecessarily.

After each wheel is created, a hinge joint with a horizontal rotational axis (vertical rotation plane) is generated between the wheel and the base, thus it is connected to the body with the ability to rotate 360 degrees in both clockwise and counter-clockwise directions. ODE provides the capability to motorize these joints, a desired velocity with a maximum force could be set for each wheel. However, since the real wheels on the same side are connected by the same belt drive, and driven by the same motor, it is not necessary to control each wheel individually. Therefore, similar to the interface functions that control real robot, three wheels on the same side are viewed as a *wheel system*, and they are controlled together (ie. when a desired velocity for the left wheel system is set, all three wheels on the left side are set to that desired

[6] Hinge joints connects the articulated bodies, in such a way that the bodies are allowed to rotate only around a certain axis, just like the function of the hinges of doors.

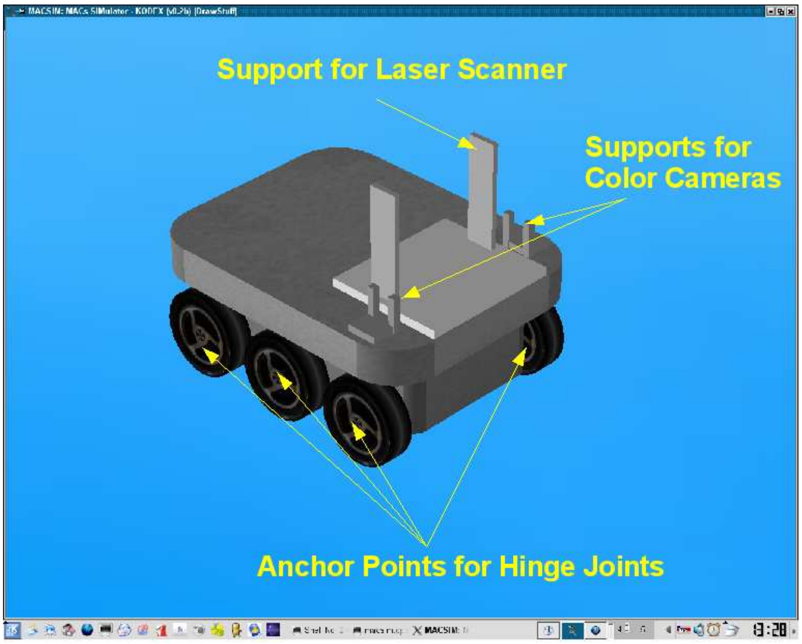


Figure 4.3: The simulated robot body and wheel system. In (a), the body of the simulated robot, after wheels are attached is demonstrated and the locations of the junction points are roughly shown. Additionally, the structures which are used to mount the laser scanner and cameras are illustrated.

velocity).

Since the motors have an output power of 90 W., real robot and its simulated counterpart are able to reach the specified speed in a short time.

4.2.3 3D Laser Scanner

As demonstrated in Figure 4.4, scanner is roughly a box-shaped object. The simulated counterpart is generated by combining a number of box and cylinders, for illustration purposes. Like its real counterpart, it is mounted on a supporting body (which is glued to the robot *bulk*) with a hinge joint that allows rotation only around horizontal axis. Similar to the wheels, the hinge joint of the laser scanner is also virtually motorized, enabling the user to set a vertical angle (range). The virtual motor mechanism rotates the sensor, making a 3D scan. However, unlike the wheel joints, which are not constrained, the laser scanner's motion is limited to an angular range $[-90°, +90°]$ in order to prevent any collision with robot body during rotation.

ODE's ray collision primitives are used to simulate the laser beams. In the simulation, instead of time of travel, direct distance is obtained using low level functions.

(a) (b) (c)

Figure 4.4: Modelling of the laser scanner. The front(a) and side views(b) of the laser scanner, and (c)its simulated counterpart.

To do this, the emission points and directions of the laser beams are computed during the simulation, and from that point, 8 meter long rays are generated. When the ray collides with an object, it's first contact point is taken from ODE, and the distance between emission and contact point is computed.

Unlike the real laser scanner, where one laser beam is transmitted to different directions by a rotating mirror, in MACSim, the 2D scan is acquired in one shot. This is performed by generating all rays in a plane, which is perpendicular to front surface of the laser scanner, as shown in Figure 4.5. However, after this level, the procedure is same: In both real and simulated robot, the scanner is rotated, and in each step, a slice of the environment is taken. Later, these 2D scans are combined to perceive the world in 3D.

In order to model the noise in the scanner, a Gaussian noise with a zero mean and 0.25 cm of standard deviation was added to each distance.

4.3 Environment Modelling

Modeling of the environment has great importance because the aim is the direct transfer of robot control code, from simulator to real world. Both the quality in display and reality of physical interactions in the simulator depend on many factors, like illumination conditions, the details on object surfaces, material types, etcetera.

The environmental objects are designed mainly to study different types of affordances, thus many different objects are included into the simulation. Some objects are attached to the environment, and might be used for navigation experiments.

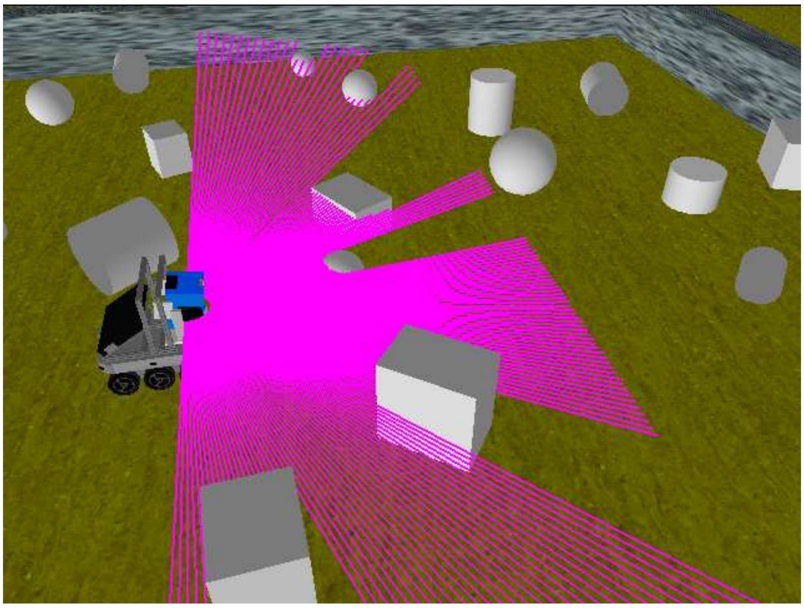


Figure 4.5: A snapshot of how simulated 2D scan is performed to acquire the range values of a planar slice.

Pushability or liftability affordances might be studied with portable objects around. Additionally, since our environment does not consist of only simple geometries, like boxes or cylinders, composite objects such as chairs and tables are incorporated into the environment. These objects, contrary to simple ones, are constructed from many different basic (collision) geometries. From now on, we will call the objects with one basic geometry as *basic objects*, and the others with more than one geometry as *composite objects*. In composite objects, the connections among the components are rigid, and the composite behaves as a single body with single mass, and center of inertia. Figure 4.6 show some *basic* and *composite* object samples, that are used during experiments.

During learning experiments, access to exact environment state will be required. Additionally, in successive trials, one would need to change the objects around and their properties on-fly. Working in information-rich and diverse environments is especially important in studying affordances of the objects. For example, in order to learn the traversibility affordances in the environment, learning module should be fed by objects, whose numbers, shapes and dimensions are changed between trials. MACSim gives its user the ability to insert new objects into the environment, delete existing ones, and change the properties, all on-the-fly.

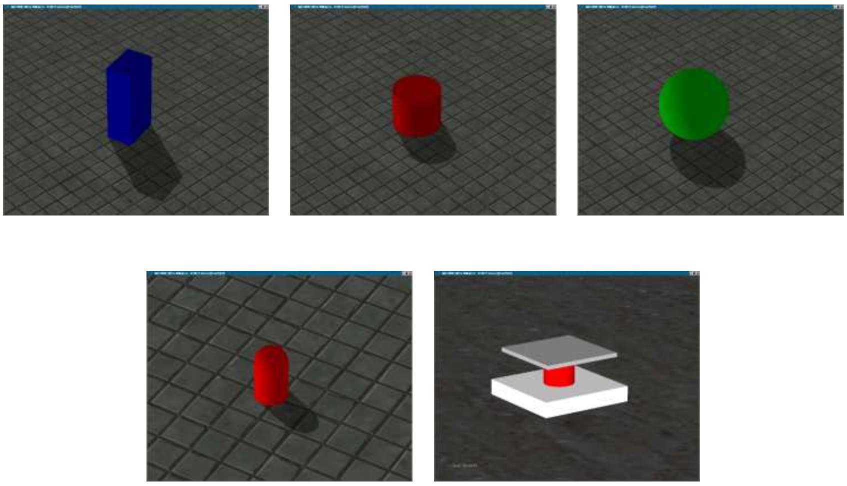


Figure 4.6: Some basic and composite environment objects that can be used in affordance test scenarios, are demonstrated.

4.3.1 Loading World Configuration Files

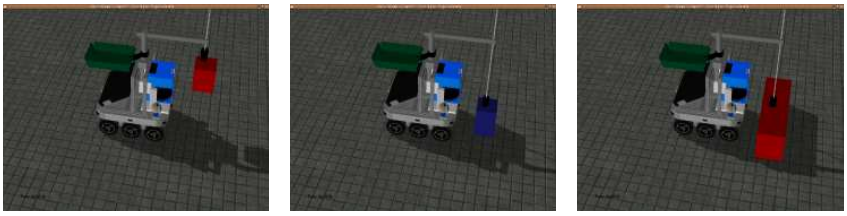
Environment is constructed using *XML file loading module*. The accepted format of the XML structure is given Appendix A and all XML descriptions which obey this format will be loaded in the virtual world.

4.3.2 Objects

As described in previous section, general purpose objects may be either *basic* or *composite*, depending on their geometries. In principle, there is no distinction between these two classes, and the interface functions applied to them. Below, you will find the functionalities that MACSim provides.

Insertion to/deletion from the environment is possible for both object types. While all *basic* objects can be created and destroyed during simulation (on-fly), only means of creating *composite* objects are through XML files. Manipulation with objects is also possible through appropriate interface functions, by obtaining the pointer of the object, and calling various functions that are described in the following paragraphs.

Color/texture information of objects are extremely useful when camera modality is used during perception. Thus, it is important to set new colors to objects, and cover with different textures. MACSim provides methods to learn the unique texture id



(a) (b) (c)

Figure 4.7: Sample scenarios for liftability. The density and weight are important object features for certain affordances.

(and its corresponding .bmp filename) of any texture, or to change it with available textures during runtime.

Changing position/orientation of any object is supported in MACSim. However, no additional check is performed regarding to possible unstable conditions. Relocated objects may appear inside other objects or in penetration with them. The behavior of the dynamics engine in such situations is unpredictable, and whole program might collapse in an instant. Thus, users should make their own collision check, prior to the use of these interface control functions.

Dynamic dimension change in objects is another supported functionality for both basic and composite bodies in MACSim. The former case is straightforward, since basic objects include only one collision geometry. In order to access or change any dimension, user should check the geometry first, and then call appropriate function specialized to that geometry type. Box, cylinder and sphere bodies are all handled in this way. Setting dimensions of composite objects is a more complex process, since they are constructed from many objects with different collision geometries. To access and change dimension of any component, users should find the geometry type of that component, and call the appropriate method for that geometry. Besides its complexity, since relative positions of components cannot be changed in the current version of MACSim, dimension changing might not used as conveniently as in basic objects' case. The general shape of the body does not remain constant after dimensions of the components are modified.

Weight and density is realistically modeled by underlying physics engine in MACSim. These properties might appear as invariant features for various affordances, like Pushability and liftability. The heavy objects are difficult to push, if it is not circular in rotation direction, and the friction with surface is not so low. Likewise, although magnetizable, some objects may be so heavy that magnetic arm could not lift them. Figure 4.7 shows a hypothetical scenario, where blue objects are too heavy and learning module should extract the color as invariant feature. Setting mass of the objects without changing their volumes will refer to direct changes in densities, and vice versa. Providing the ability to change the weight and density of objects gives users the ability to model the environment more realistically, since different real world objects have different densities. In the current version of this simulator, although direct setting of mass is correctly modeled, mass computation is not performed in high fidelity when densities of composite objects are changed.

Force and torque that accumulated over any object can be extracted by using appropriate functions.

Velocity of any object can be obtained or changed by the given interface functions. When absolute linear velocity for a particular body is set, the object will try to move in the given direction with constant speed. The collision situations are not tested, and most probably instabilities will occur, if this function is used carelessly.

Fixed objects

Obstacles are modeled as fixed objects in MACSim. They are attached to the environment, thus it is impossible to relocate them during simulations, based on physical laws. They can only be replaced by using appropriate *simulation control interface* functions, which corresponds to human intervention to the environment in real world conditions. Obstacles might be in form of both basic geometric shapes, like boxes or cylinders, or they might be constructed as composite objects. In Figure 4.8, some fixed objects are demonstrated. They might be used as circular small obstacles to be used in navigation experiments.

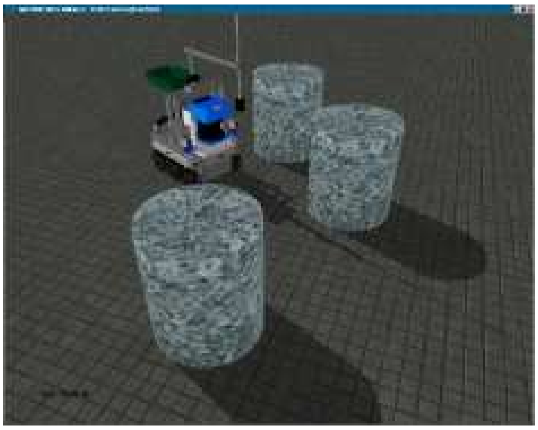


Figure 4.8: Interactions with fixed circular obstacles.

Portable objects

These objects are not attached to any body or surface, thus free to move around. Their behaviors are determined according to physical laws when any interaction with other bodies occurs. As in the case of fixed objects, they can be constructed by one basic object, or a number of basic geometries which are combined to form one solid composite portable object. Different perception and action experiments regarding to the provided affordances in the environment are possible with portable objects in various positions, orientations and shapes.

CHAPTER 5

AFFORDANCE BASED CONTROL OF THE ROBOT

In this chapter, the architecture of the affordance-based robot control will be provided. First of all, how the robot acts to and perceives its environment, with the description of the general characteristics of the world will be described. Then, how the robot learns the traversibility affordance of the environment with respect to its actions will be described. Lastly, how robot selects its behaviors based on the learned affordances will be presented.

Currently learning is not integrated into the online control of the robot, thus Kurt3D has two modes of operation: It could be either in *exploration mode*, or in *execution mode*. In *exploration mode*, the robot moves in an environment full of objects, and it tries to learn the traversibility affordance of the environment. In different trials, it performs various behaviors, and by physically interacting with the objects around, it learns a mapping between the environmental situations and the results of its actions. After this exploration phase, it is placed in a room cluttered with various objects (some of them not seen before). A "high-level" motivation module gives the robot a preferred movement direction. The robot tries to go in that direction by selecting appropriate behaviors, that are afforded by the immediate environment.

In this chapter, the behavior repertoire of the robot will be described first. Then, a description of its world, and how Kurt3D perceives its environment will be provided. Later, the setup for robot *exploration* will be given, and how learning is performed in this setup will be described. In the last section, the control of the robot in *execution* mode will be given.

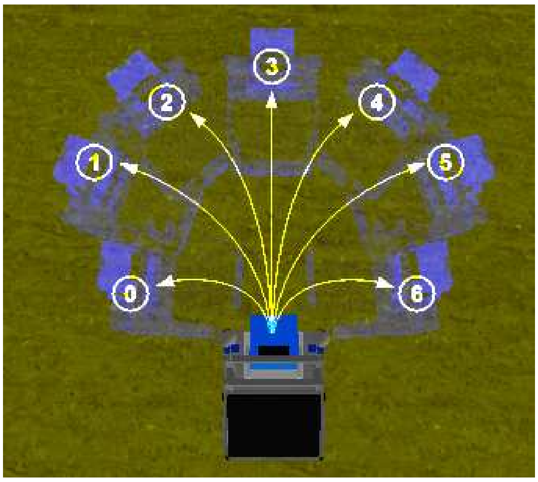


Figure 5.1: Illustration of the behaviors. The trajectory for each behavior, and the length of the paths are shown where 3^{rd} movement corresponds to *move-forward* behavior. The 0 and 6 numbered movements, which correspond to *turn-sharp-left* and *turn-sharp-right* behaviors, have a smaller movement range than others. See Table 5.1 for details.

5.1 Behavioral Repertoire

The robot is provided with a set of pre-coded move behaviors, to go in certain directions, like *go-to-left*, *go-forward* etc. Each behavior takes control of the robot for a certain amount of time. As a result, the command that are sent the wheel actuators will remain same for that period, independent of any response from the environment. Left and right wheels are set to fixed speeds for each behavior, restricting the robot to follow a circular arc during any behavior (or forward linear path). Thus the overall movement of the robot is the combination of these discrete behaviors, where the traversed path will be the composition of individual arc segments in the end. Since the robot is desired to move in certain directions on approximately linear paths, the execution times are kept small. Such an open-loop control and discrete behaviors are utilized to ease the design of the affordance based system.

Figure 5.1 illustrates seven simple hand-coded behaviors, which result in movement in seven different directions. As shown, there are three left, three right, and one forward move behaviors. From now on, these will be called as *turn-sharp-left*(0), *turn-left*(1), *turn-smooth-left*(2), *move-forward*(3), *turn-smooth-right*(4), *turn-right*(5), and *turn-sharp-right*(6) from left to right. The behaviors, numbered between $[0 - 6]$, are designed such that the total displacement is same in all of them. However, since a

Table 5.1: The actuator parameters and success criteria for each behavior.

Action	V_l	V_r	t	d_1	d_2
Turn-Sharp-Left	$0.06m/s$	$0.3m/s$	30 steps	$0.5m$	$-0.33m$
Turn-Left	$0.15m/s$	$0.3m/s$	40 steps	$0.47m$	$-0.49m$
Turn-Smooth-Left	$0.21m/s$	$0.3m/s$	40 steps	$0.72m$	$-0.36m$
Move-Forward	$0.25m/s$	$0.25m/s$	40 steps	$0.8m$	$0m$
Turn-Smooth-Right	$0.3m/s$	$0.21m/s$	40 steps	$0.72m$	$0.36m$
Turn-Left	$0.3m/s$	$0.15m/s$	40 steps	$0.47m$	$0.49m$
Turn-Sharp-Right	$0.3m/s$	$0.06m/s$	30 steps	$0.5m$	$0.33m$

- V_l and V_l : left and right wheel speeds,
- t: execution time,
- d_1 and d_2: required displacements in longitudinal and lateral axis for a successful behavior.
- One step of simulator corresponds to 80 ms in real world.

turn-sharp-X behavior, which is executed for a long time, would result in a circular path (instead of an approximately linear path like others), it's execution time (and length of the traversed path) is smaller when compared to others. The parameters that are used in the simulator are summarized in Table 5.1, together with the traversed distances for all behaviors. Along with each behavior, a success criteria is also provided in terms of the displacement the robot made.

5.2 The Environment and Its Perception

The environment is said to be traversible in a certain direction, if the robot (moving in that direction) is not enforced to stop as a result of contact with an obstacle. Thus, if the robot can push an object (ie. by rolling it away), that environment would be traversible even if the object is on robot's path, and a collision with it occurs. This point of view is quite different from classical object avoidance approaches where any collision with any object is avoided. The physical properties of the objects become important in our case, and as a result, several objects with various geometries and dimensions should be included in the world.

In our environment, the objects might have one of the following three geometrical shapes:

- rectangular boxes () that are not-traversible,

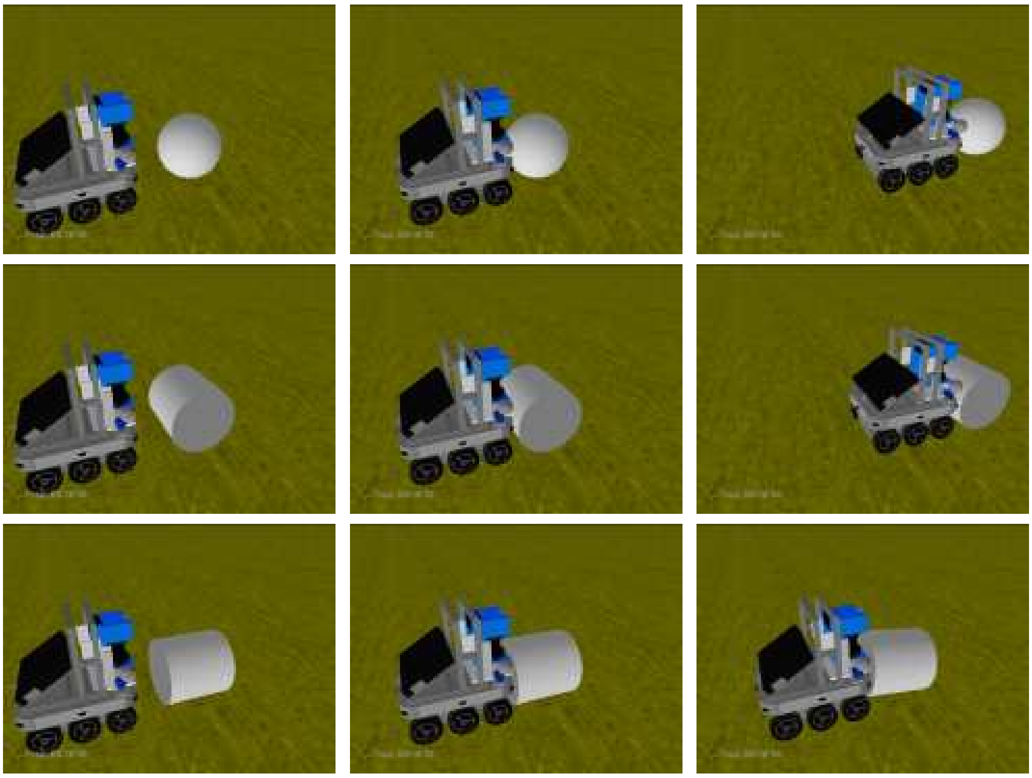


Figure 5.2: Sample interactions with objects. The snapshots in each row corresponds to interaction with a different object. The left-most and right-most columns show the initial and final positions of the robot and objects for the *move-forward* behavior. The first contact during behaviors are demonstrated in the middle column. As seen, top-most two objects afford traversibility. On the contrary, the robot is not able to roll the object in the bottom, even it is a cylinder.

- spherical objects ( ) that could roll in all directions,
- cylindrical objects
  - cylindrical objects that are placed in upright position ( ), thus non-traversible, and
  - cylindrical objects that lie on the ground ( ), and can roll in one axis. Unlike above ones, these objects do not have a fixed affordance. Their affordances depend on the orientation of the object relative to the robot's collision direction.

Figure 5.2 shows some possible interactions with various objects.

Traversibility affordance of the environment highly depends on distance and shape of the objects around, and laser scanner readings suit well for this task. Thus, al-

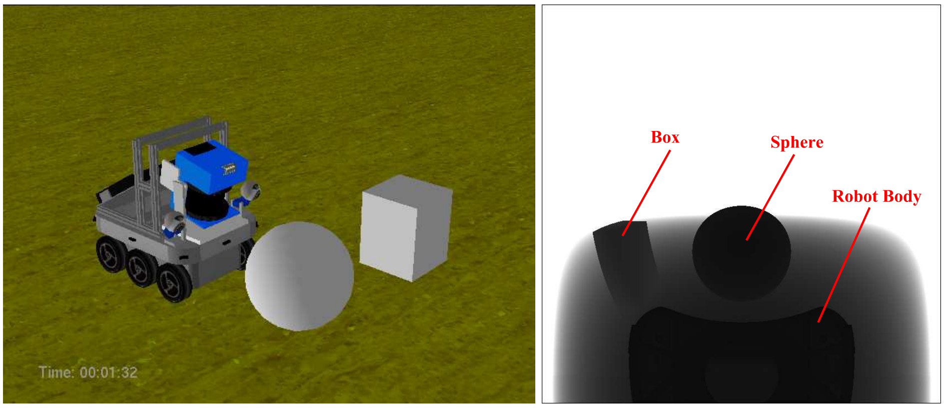


Figure 5.3: Illustration of a 3D laser scan using range images. Range value in each laser beam direction is coded as gray-scale values in the image (closer points are darker, and further ones are whiter). Each row in this image corresponds to one planar slice of the environment, read by the 2D laser scanner in a specific horizontal angle (step). Thus, each pixel (x, y) corresponds to one direction, a (α, β) angle pair more concretely.

though Kurt3D is equipped with many other sensor modalities, only the 3D laser scanner is utilized in our experiments. In detecting the traversibility affordance, the robot makes a full 3D scan of the environment, and decides in which direction it is able to move. From this scan, the robot should extract a set of features, and make its decisions based on these features. Figure 5.3 shows the result of a laser scan, which is represented as a gray-scale range image. As seen from the figure, there are two objects in front of the robot, one spherical object in its front, and one box on the left of this object. In such an environment, the movement directions except *left-forward*, would be afforded, based on the extracted distance and shape features. Next section describes how the laser scanner data is processed in order to extract these features.

Feature Set

In extracting the features, the range image is processed locally, thus no object or background detection phase is involved. The raw range data is processed in three sequential steps, where $i)$ the range image is first down-scaled, $ii)$ this lower resolution image is divided into rectangular grids, and $iii)$ some features that include distance and shape related information for these grids are computed.

Figure 5.4 shows a 720×720 range image, and its down-scaled counterpart, where

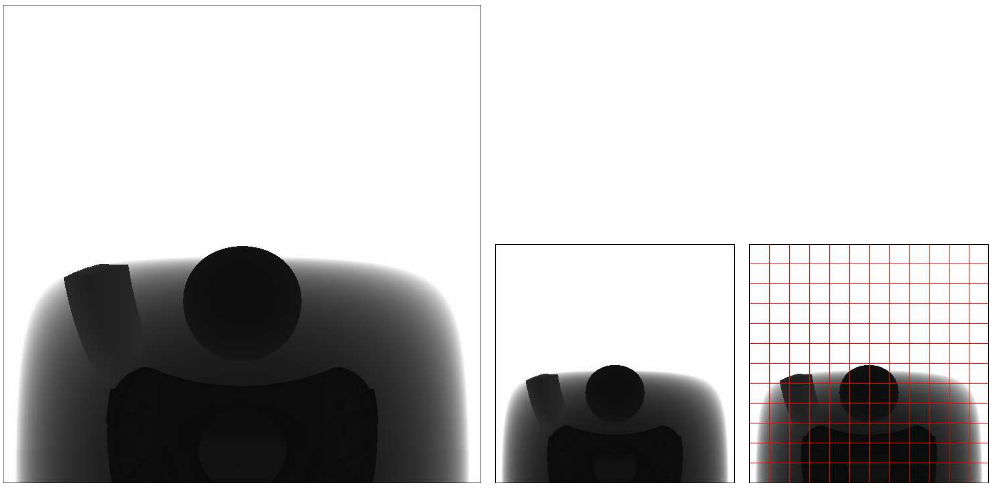


Figure 5.4: Down-scaling the image, and dividing it with rectangular grids.

four pixels in the first image are averaged into one. The motivation behind averaging is to decrease the noise in individual laser readings.

The obtained range image is later split into uniform size rectangular grids. As illustrated in Figure 5.4, each rectangular grid corresponds to one part of the whole 3D scan, which includes distance of the points in that local grid. These rectangular grids are then studied independent to each other, and from each grid a number of distance and shape related features are extracted.

Computing features from rectangular grids

There are two feature sets that are obtained for each grid, distance related ones, and shape related ones. Since, the data explicitly provides the distance information, the computation of the first feature set is straightforward: i) the closest point in the grid is found, and its range value is assigned as *minimum*, ii) the distance value of the farthest point is assigned as *maximum*, and iii) the *average* of all the range value in the grid is the third distance related feature.

In order to obtain the second feature set, which includes shape information , the surface characteristics of corresponding grid, are employed. The local shape of any object could be represented by the normal vector of the surface in that point. In our case, in order to extract some shape features from the rectangular grid, the distribution of the normal vectors over this segment is utilized (Figure 5.6). Thus, for each

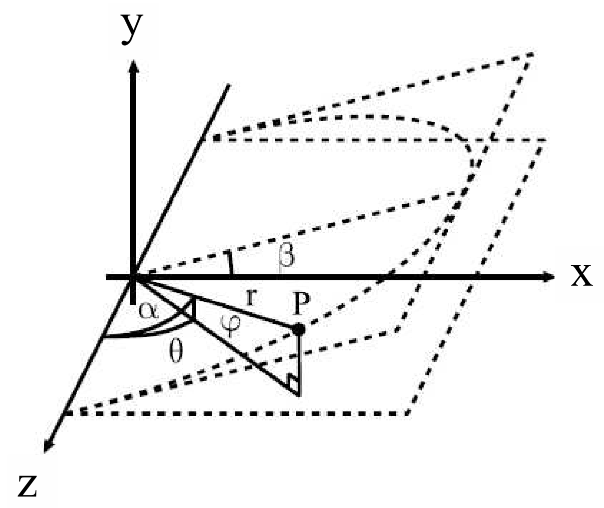


Figure 5.5: The coordinate system of 3D scanning. The laser beam (reflected from P) and its corresponding angles in the coordinate system. (The figure is reproduced with permission from Stefan Gächter [38].)

point, a normal vector is found, and the distribution of these normals represent the overall shape of the local grid.

Computing the normal vector For each point in the range image, a normal vector is computed using the parameters (angles) of the corresponding laser beam. In Figure 5.5, **P** is such a point. In this figure, β corresponds to the vertical angle of the physical rotation of the scanner, and α is the angle of the rotating mirror that redirects the beam in 2D mode. In order to obtain the normal vector, the $\boldsymbol{p}$ vector should be computed first:

$$\boldsymbol{p}_x = r.sin(\alpha).cos(\beta)$$

$$\boldsymbol{p}_y = r.sin(\alpha).sin(\beta)$$

$$\boldsymbol{p}_z = r.cos(\alpha)$$

where r is the range value, obtained for that point.

We compute the normal vector over three points, two more , $\boldsymbol{p}_1$,$\boldsymbol{p}_2$ for each **P** are selected in 3×3 neighborhood. The normal vector for each point is computed by cross-producting vectors $\boldsymbol{p}_1 - \boldsymbol{p}$ and $\boldsymbol{p}_2 - \boldsymbol{p}$ [1]:

$$\boldsymbol{N}_p = (\boldsymbol{p}_1 - \boldsymbol{p}) \times (\boldsymbol{p}_2 - \boldsymbol{p})$$

[1] The points p_1 and p_2 are selected such that the $p_1 \rightarrow p \rightarrow p_2$ are traversed in counter-clockwise direction

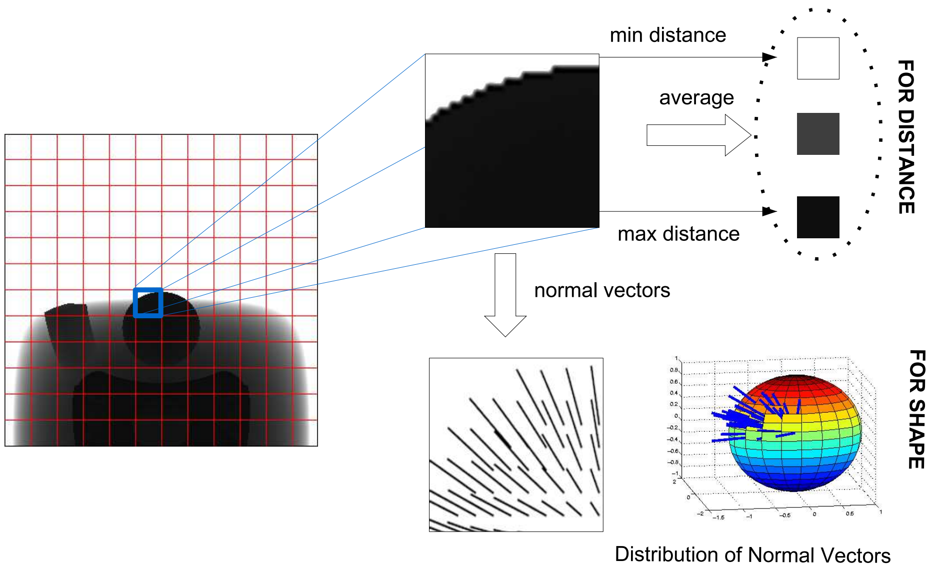


Figure 5.6: Surface normal vectors and distance features. The block arrows demonstrate that the features are extracted from the whole of the image, where features shown with thin arrows, are obtained from individual pixels (points).

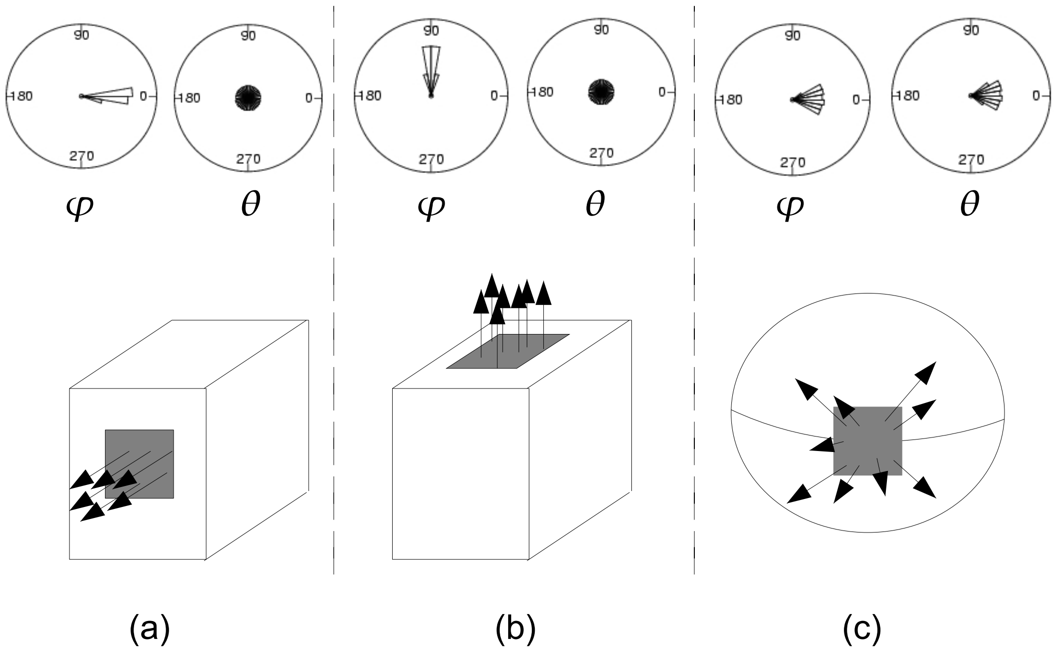


Figure 5.7: Sample angular histograms in channels θ and φ. (a) and (b) represents the grids on vertical and horizontal planar surfaces, (c) shows a grid which covers a part of a spherical surface.

At the end, for each point, 4 different normals are computed (from 4 different $< p_1, p_2 >$ pairs), and they are averaged to obtain the normal vector for that point.

Since the normal vector could be represented by two angles, θ and φ (Figure 5.5), two angular histograms (one for θ and one for φ channels) are computed using the normal vectors in the corresponding grid. Figure 5.7 shows some hypothetical surfaces and their corresponding angular histograms. As illustrated, a spherical, a horizontally planar, and a vertically planar surface could be differentiated using these histograms. For example, the histograms of planar surfaces are more compact, and the histograms for spherical surfaces are more scattered. Moreover, the horizontal and vertical grids in the figure, could be differentiated in their φ channel, since in this channel, there is a 90° degree difference.

After the angular histograms in both channels are computed for each grid, the frequency values (for each degree interval) will be provided as features. Thus, if the interval size of the histogram is h, $2 \times h$ shape features will be computed for each grid, where 2 refers the two angle channels. As a result, if there are p grids, the size of whole feature vector would be:

$$d = p \times (3 + 2 \times h)$$

where 3 corresponds to the three distance values (minimum, maximum, and mean). Figure 5.8 shows how the feature vector is formed from these grids.

The histogram is divided into 18 intervals ($h = 18$), and the range image if split into 30×30 grids ($p = 900$) so that total number of features computed over a down-scaled range image of 360×360 is:

$$900 \times (3 + 2 \times 18) = 35100$$

After the features are computed from the 3D laser range scan, the mapping between these features and traversibility affordances will be learned, as will be described in the next section.

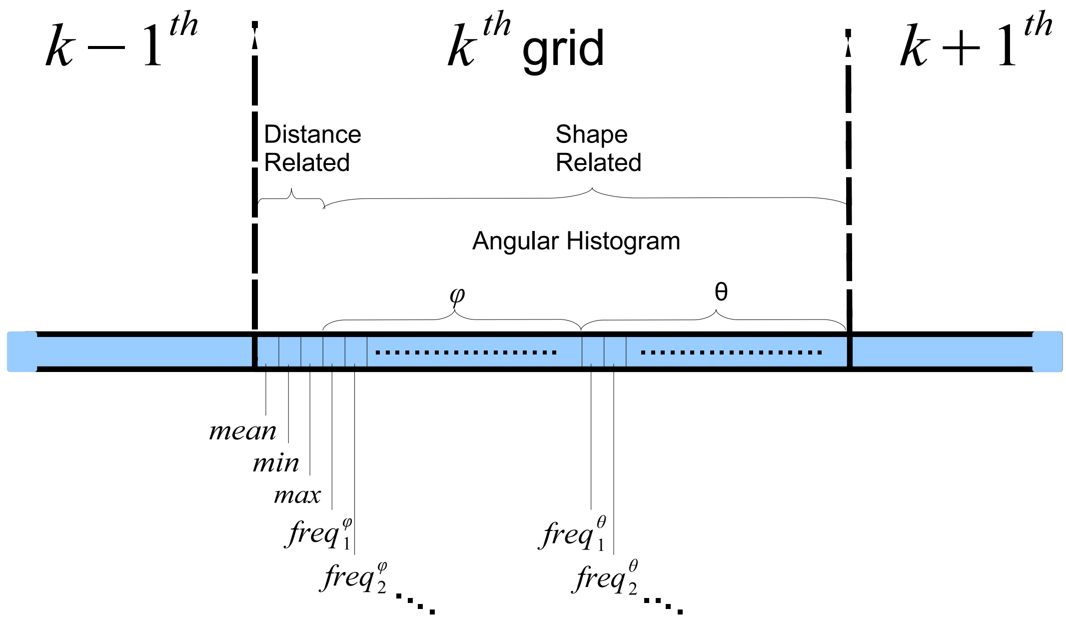
5.3 Exploration Mode

In the beginning of its *exploration mode*, the robot has no prior-knowledge about the world, it does not even know that movement over a flat ground surface is possible. By executing its behaviors, and observing success/fail, it gradually learns which environmental situations afford which behaviors.

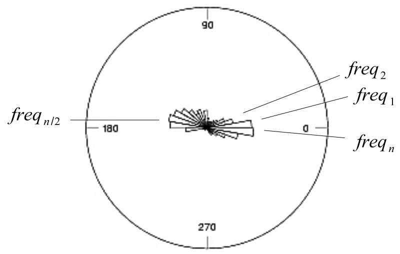
Algorithm 1 Learning by exploration

{Exploration phase}
for each trial k (from 1 to m) **do**
  Put the robot in a randomly constructed environment.
  Make a 3D scan
  Compute feature vector, $\boldsymbol{f_k}$
  for each behavior b^i **do**
    Perform b^i
    Find result of behavior, r_k^i.
    Put $< b^i, \boldsymbol{f_k}, r_k^i >$ into repository.
    Reset robot and object positions.
  end for
end for

{Learning phase}
for each behavior b^i **do**
  Fetch samples $< \boldsymbol{f_k}, r_k^i >$ from repository for behavior b^i.
  Find a set of relevant features $\mathcal{F}^i$.
  Train the SVM model, M^i, with relevant features.
  Store $\mathcal{F}^i$ and M^i for perception of affordances in *execution mode*.
end for



(a)



(b)

Figure 5.8: Construction of the feature vector.

Algorithm 1 describes how the robot is trained by the means of exploration trials and a batch learning phase. In the rest of this section, the modules of the learning architecture (Figure 5.9) will be provided.

In each trial of the learning phase, the robot is placed in a different environment, where objects are distributed around, with random positions and orientations. Figure 5.10 shows a sample learning setup. Prior to its action, the robot performs a 3D scan, and the features are computed using the method in the previous section. Then, the behavior is executed for a certain amount of time, and the result of this action, in terms of *success* or *failure* is found. In order to find the success of the behavior, the

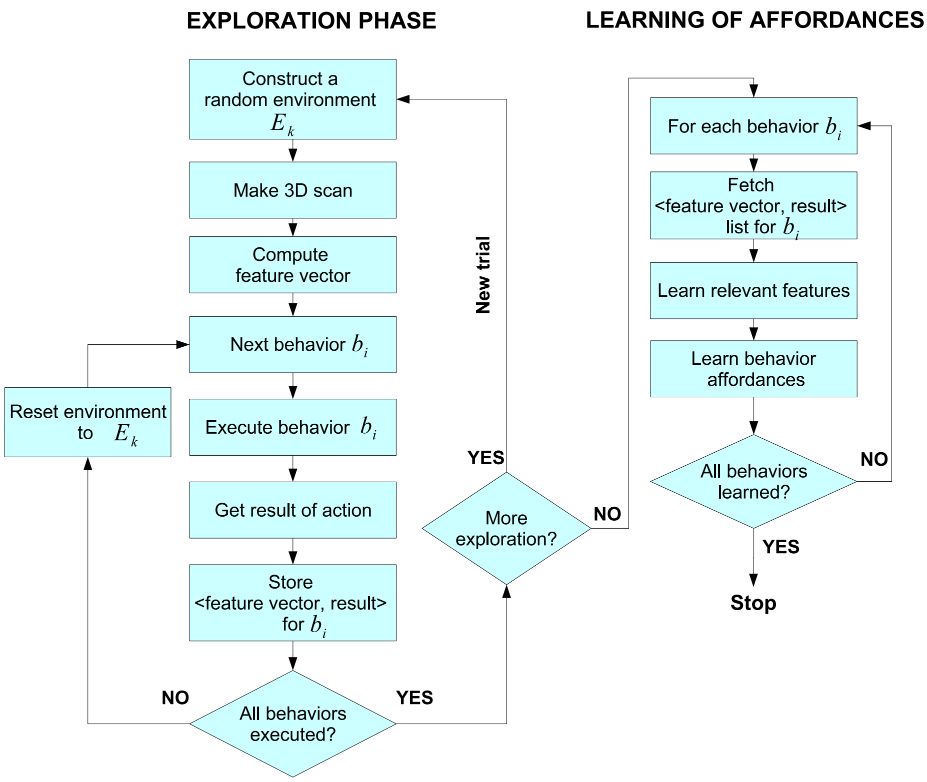


Figure 5.9: Diagram of exploration and learning of affordances.

displacement of the robot is extracted from the simulator, and it is compared with a pre-defined success criteria specified by d_1 and d_2 in Table 5.1. A more straightforward alternative, the collision check, would not give the true success of the behavior, since the robot could collide with an object, roll it, and proceed on its path. Figure 5.10 illustrates some snapshots from intermediate steps of these trials[2].

5.3.1 Selection of Relevant Features

When the feature vector is formed using the method described earlier, the size of this vector would be on orders of tens of thousands. Most of the features in this vector would be irrelevant for a particular affordance, thus do not require to be further processed. For example, if the robot looks for the affordance to move in forward direction, only the grids, which are horizontally centered in the range image would be relevant. The features, which are computed from other grids do not need to be processed, and should be filtered out. In the following section, the feature selection

[2] The corresponding movie can be downloaded from `http://kovan.ceng.metu.edu.tr/~emre/setup.mpg`

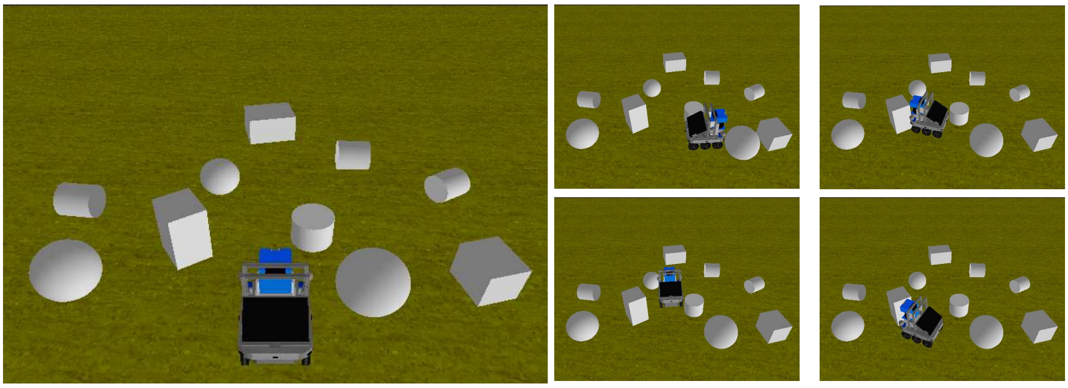


Figure 5.10: A sample trial. The left-most snapshot shows the initial position of the robot. The two snapshots in the middle shows the successful traversing trials when *move-forward* and *turn-sharp-right* are applied. The figures on the right corresponds to unsuccessful trials, where robot performs *turn-left* and *turn-smooth-left* behaviors.

method, which automatically selects the relevant features, will be described.

ReliefF Method

Originally proposed by Kira and Rendell [39], ReliefF [40] algorithm aims to estimate the quality of each feature in a feature set, based on its impact on the target category of particular samples of this set. In [41] the family of ReliefF algorithms are systematically analyzed, and proved to be robust in the face of noise and incomplete data. Unlike many other heuristic methods, which work with the assumption of conditional independence of the feature, ReliefF Method is able to deal with the conditional dependencies between features, thus is very popular in the literature. Furthermore, it is successfully applied to huge datasets with feature numbers on the order of thousands, for example in order to select the relevant pixels in images that distinguish between cats and dogs [42].

In ReliefF algorithm, if a feature is important, then it should be able to distinguish very similar samples with different target values (categories). Thus, the weight of any feature is increased, if it has similar values for the samples in same categories, and it has distinct values for samples in different categories, especially for samples that seem to be very similar to each other. The ReliefF algorithm, which is the two-category case of Kononenko's extended version [40], is presented in Algorithm 2. We employed this modified version because it is able to tackle with noise in the data, by computing the weights over several (K) similar samples.

Suppose $\mathcal{S}$ contains n samples, $\boldsymbol{s}_1, ..\boldsymbol{s}_n$, where each sample corresponds to a feature vector in d-dimensional space. The target values (categories) of these samples, namely traversibility and non-traversibility, are given in vector $\boldsymbol{c}$, where $\boldsymbol{c}_k$ corresponds to the category of the sample k. Given the set of feature vectors, and their target values, ReliefF algorithm first computes the weights of each feature, $\boldsymbol{w}_1, ..\boldsymbol{w}_d$. Then, it filters out some of the features, if their weight (or relevance) is smaller than a pre-determined threshold, τ.

Algorithm 2 ReliefF

$\boldsymbol{w}_f \leftarrow 0$, where $1 \leq f \leq d$ (initialize weights)
for $i = 0$ to m **do**
  Select a random sample feature vector $\boldsymbol{s}_r$ from $\mathcal{S}$
  Compute distance of $\boldsymbol{s}_r$ to all other samples in $\mathcal{S}$
  Find k nearest samples, which are categorized as $\boldsymbol{c}_r$, and put them into set of nearest hits, $\mathcal{H}$. ($\mathcal{H}$ = $\{\boldsymbol{h}_1, ..\boldsymbol{h}_k\}$)
  Find k nearest samples, whose categories are different from $\boldsymbol{c}_r$, and put them into set of nearest misses, $\mathcal{M}$. ($\mathcal{M}$ = $\{\boldsymbol{m}_1, ..\boldsymbol{m}_k\}$)
  for $f = 0$ to d **do**
    $\boldsymbol{w}_f \leftarrow \boldsymbol{w}_f - \frac{1}{m \cdot k} \sum_{j=1}^{k} \mid s_{r_f} - h_{j_f} \mid + \frac{1}{m \cdot k} \sum_{j=1}^{k} \mid s_{r_f} - m_{j_f} \mid$
  end for
end for

As described in Algorithm 2, if the value of a feature is different in its nearest misses, then this feature is thought to have impact on the target value differences, and its weight is increased. However, if a feature's value is same in the samples, whose categories are different, then, this feature's weight is decreased, since it has no effect on the target value.

There are two main parameters which affect the results of this algorithm. k, which is used to deal with noise, is one of these parameters. 10 is found to be a reasonable value for k, as suggested in [40] (increasing and decreasing of this value do not change the performance much in our experiments). However, the threshold value, τ, directly determines which features to be filtered out. Selection of a wrong τ could filter out very important features. Therefore, we optimized this parameter for each sample set (behavior type). Figure 5.11 illustrates our approach.

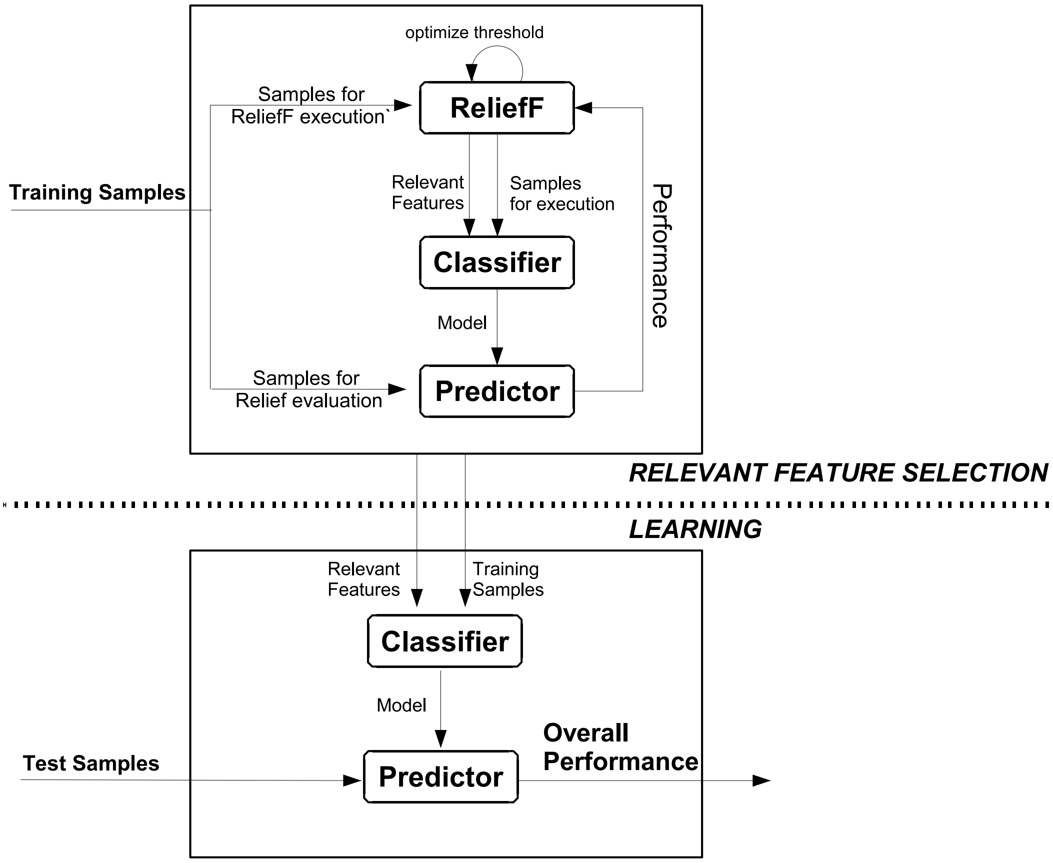


Figure 5.11: Optimization of ReliefF threshold.

5.3.2 Classification of Features

In this study, Linear Support Vector Machines (SVMs) are used to label the relevant features as traversible and non-traversible. SVMs, which are introduced by Vladimir Vapnik in 1998 [43], are powerful supervised learning tools used in classification and regression problems. They are very robust when the input data is noisy, and able to deal with large datasets and input spaces. In SVMs, the input space is converted into a high-dimensional feature space, where a hyperplane that best separates the two classes, is computed for a training sample set. Then, the prediction is performed over this hyperplane. An external library [44] is utilized in learning from the training samples that are obtained during exploration trials.

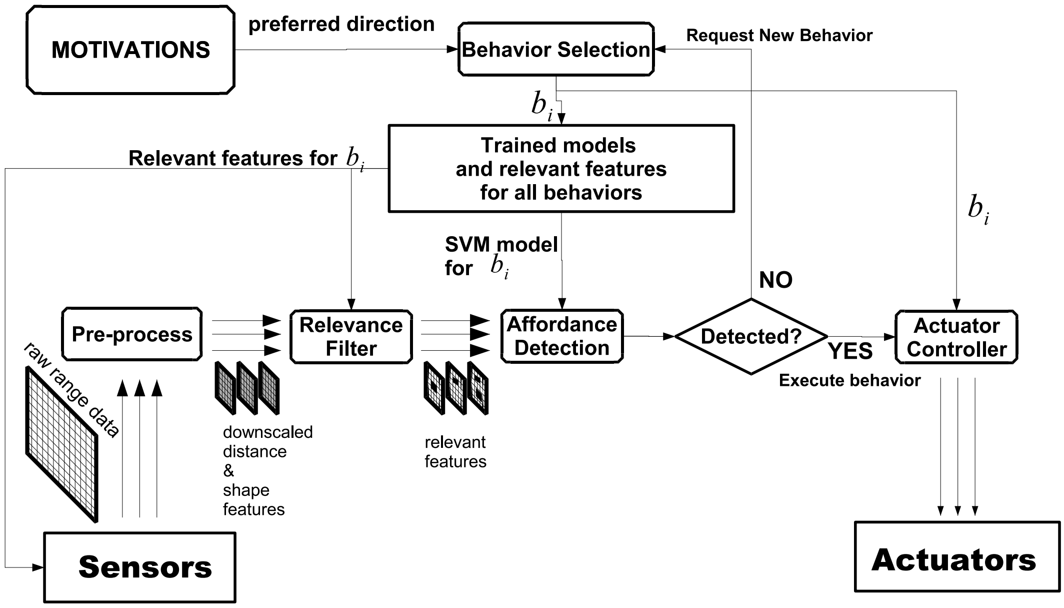


Figure 5.12: Robot control architecture in *execution* mode.

5.4 Execution Mode

In this mode, the robot is governed by a "high-level" *motivation* module. The behaviors, whose affordances will be searched, are selected based on a *preferred direction*, that is sent from this module. After the desired behavior is selected, the features which are relevant to this behavior (identified during exploration trials), are fetched from the repository. Later, these features are used in determining (predicting) whether the behavior is afforded or not. If it is afforded in robot's immediate environment, the behavior will be performed. If not, a new behavior, which would move the robot close to the *preferred direction*, is requested. The control architecture, during the *execution mode* is provided in Figure 5.12.

The set of relevant features are utilized in three different levels, $i)$ in parameterization of physical sensors, $ii)$ in identifying relevant rectangular grids, thus only computing features over these grids, and $i)$ in filtering out the individual irrelevant features on the relevant grids, while predicting based on the learned SVM and detecting the affordances.

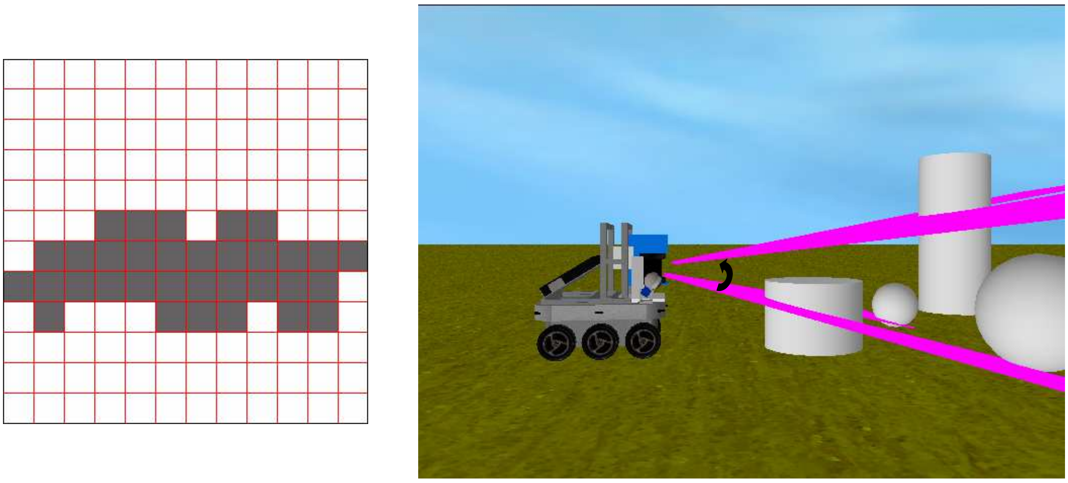


Figure 5.13: Parametrization of the laser scanner. Parametrization is based on the relevant grids of range image. The left figure illustrates a hypothetical relevant feature set, and right figure shows the horizontal range of laser scanning.

5.4.1 Parameterization of the laser scanner

Some parts of the environment would affect the affordance of a behavior, and some parts have no relation with that behavior. Therefore, in order to save the physical resources and time, these parts should not be scanned. In our case, the range of scanning angles could be constrained based on the relevant angular segments. Figure 5.13 demonstrates a hypothetical relevant feature set, where grids, that locate in vertically middle region of the range image, are relevant. As a result, the horizontal scanning range is not limited, but the laser scanner is modulated to scan a constrained vertical angular range.

5.4.2 Filtering Out Irrelevant Features

After the physical sensing is constrained and only a constrained part of the range image is acquired, the remaining irrelevant features are filtered out. This is done in two stages:

1. The grids that contain relevant features are found, and shape and distance related features are computed over only these grids.

2. Since some grids may include relevant and irrelevant features, the later ones should be filtered out.

5.5 Discussion

As discussed in Chapter 3, laser scanners become very slow when they are used in 3D mode. For example, our sensor make a full 3D scan in 45 seconds. However, the world part, which have relations with the desired behavior, are some behaviors might not require such a full scan. As illustrated in Figure 5.13

CHAPTER 6

EXPERIMENTAL RESULTS

In this chapter we first describe the experiments conducted towards the learning of traversibility of the KURT3D robot from range images, and present the results. In the rest, first, the two experimental environments in which the training of the robot is carried out is described. Then, the results of the method used for learning relevant features is presented and described. The learned knowledge and the generalization performance of the method is systematically evaluated, using different training cases and evaluating the learned knowledge against different novel objects and surfaces. Finally the results of the learning, which is carried out in MACSim, is tested on the real KURT3D robot on a number of experiments.

6.1 Experimental Setup

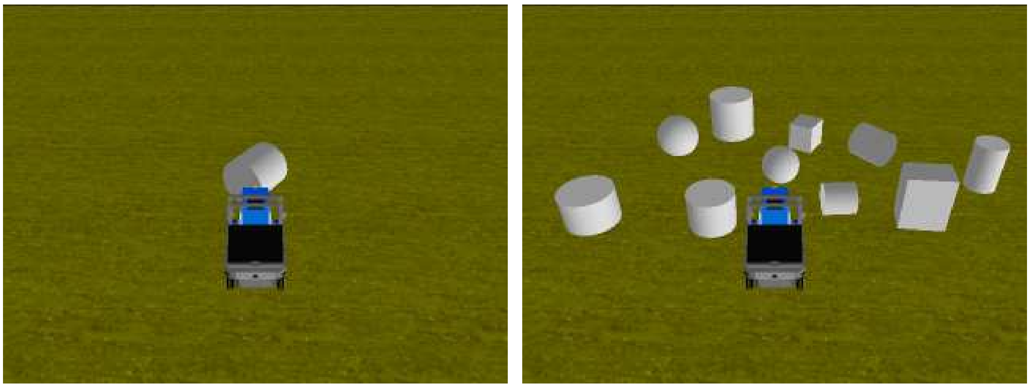
The learning and a number of systematic evaluations of learning is carried out only in MACSim. Two types environments called, *simple* and *complex*, are used for the experiments. In the *simple environment*, a single random object is randomly placed in front of the robot whereas in the *complex environment*, a number of random objects are randomly placed in front of the robot.

In the *simple environment*, the robot tests the existence of traversibility affordance for only *move-forward* behavior, whereas in the *complex environment* all the behaviors in robot's repertoire are tested. A summary of the parameters that are used for the construction of these environments are presented in Table 6.1 and snapshots from the two types of environments are shown in Fig. 6.1.

In order to learn the perception of traversibility affordances of a complex environment in different instances, the robot should be provided with a crowded world

Table 6.1: The parameters of environment construction.

	Simple Environment	**Complex Environment**
Angular Range	$0°$ (fixed)	$[-90°, +90°]$ of robot's frontal area
Orientation	random rotation in vertical axis	random rotation in vertical axis
Distance	$40cm$ from robot body (fixed)	$[20cm - 170cm]$ from robot body
Object number	1 object	10 objects
Shapes	randomly select among (, , ,)	randomly select among (, , ,)
Dimensions [1]	$[20cm - 40cm]$	$[20cm - 40cm]$



(a) Simple Environment

(b) Complex Environment

Figure 6.1: Snapshots from training environments. In the particular sample environment, demonstrated in (a), a cylinder is placed in front of the robot. However since the object category is randomly selected, it could have been any of the (, , ,).

during its exploration trials. The *complex* environment is designed for this purpose.

Contrary to its complex counterpart, *simple* environment set is specially designed to show the generalization capability of the affordance based approach. In this set, one of the four different types of objects is inserted in an instance. Thus, we can train our model, using only a subset of these objects, and later test the model by other subset. In this way, we can show that, if possible, the robot is able to generalize the learned model, to novel objects.

[1]Dimensions refer to diameter, width, height, and depth for different geometries.

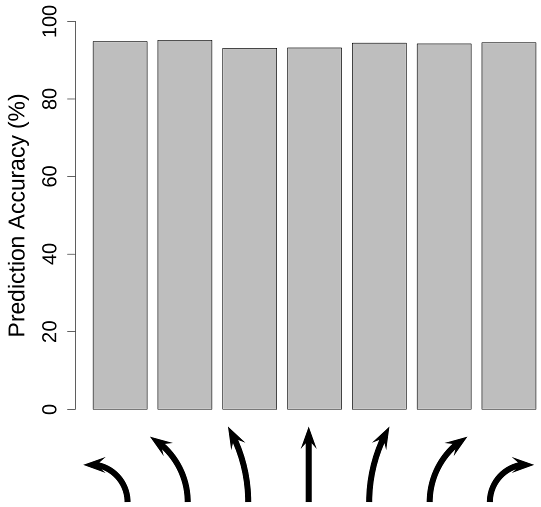


Figure 6.2: The accuracy in predicting affordances.

6.2 Learning in Complex Environments

3000 different setups are used in this experiment, 2000 of them are utilized for training of the model, and 1000 for testing the performance. As shown in Figure 6.2, the prediction accuracy of the robot for all different behaviors is around 94%.

Learned Relevant Features

In this section, the relevant features that are obtained by optimizing the threshold parameter of ReliefF algorithm, will be presented. In Figure 6.4, the performance change during this optimization phase is demonstrated. Additionally, the mapping between relevant feature numbers and their corresponding performances are provided. As shown in the figures, between $100 - 400$ features among 35100 features are found to be relevant for perception of the traversibility affordance for various behaviors. In other words, at most, 1.1% of the whole feature set is found to be relevant in detection of affordance of any behavior.

Table 6.2 provides the optimized threshold values and the number of relevant features. Similar values for symmetric actions might have been expected, however it did not appear in the table. Since, the size of the training set is not high relative to the space it represents, we think that it becomes more similar if training set size is increased.

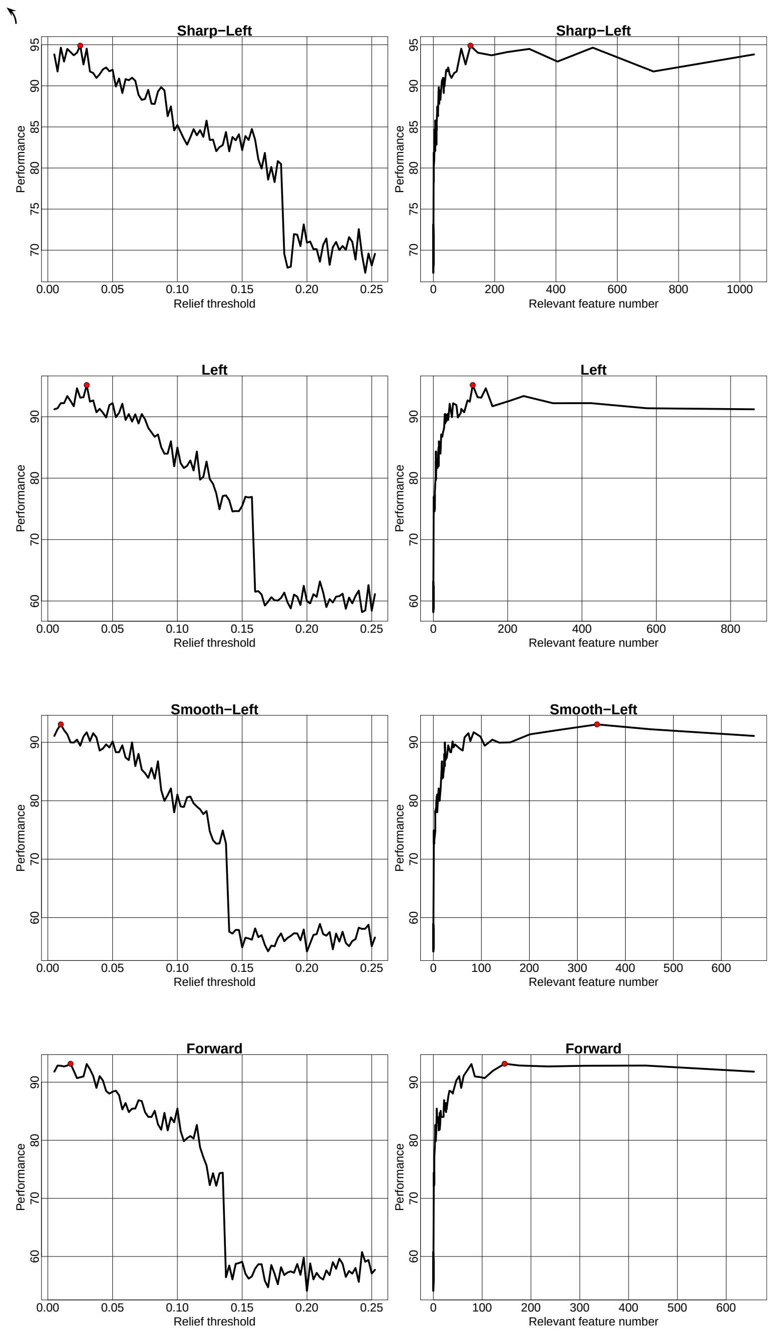


Figure 6.3: The change in relevant feature count and performances during optimization of threshold value of ReliefF for behaviors in forward and left.

Learned Relevant Grids

In order to identify the important regions in the images for different behaviors, the relevancy in rectangular grids will be examined. The grids, which include relevant features are marked as *relevant*. In Figure 6.5, the relevant grids for all behaviors are

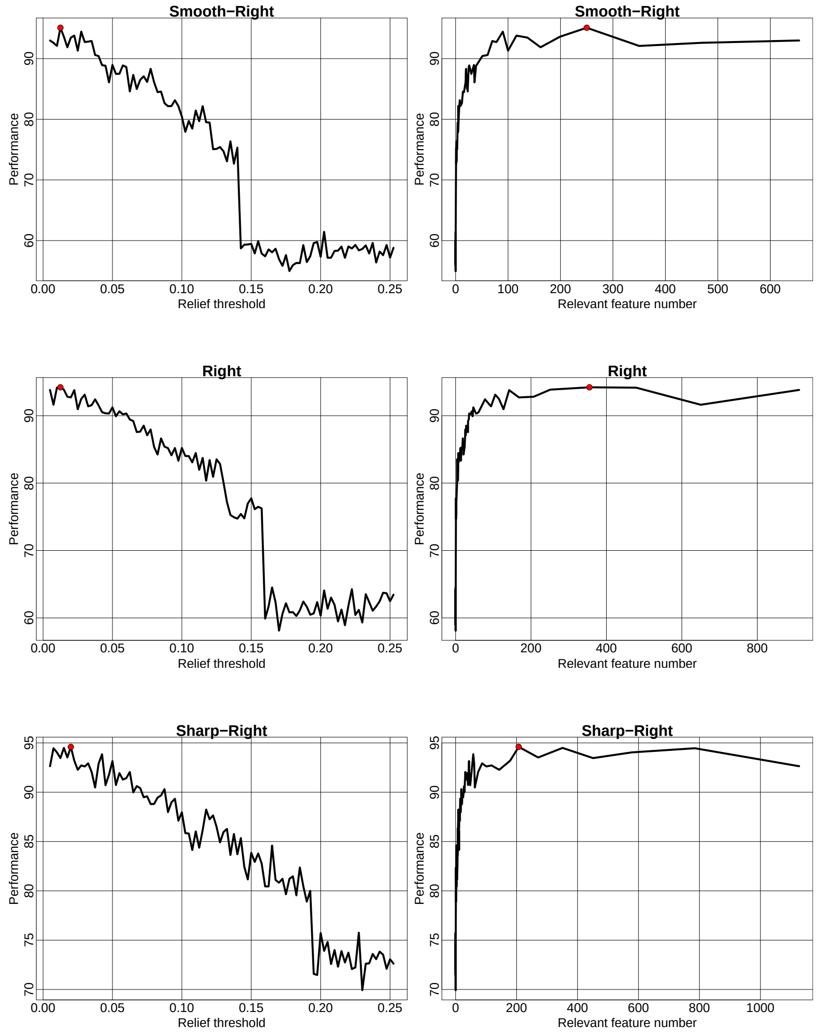


Figure 6.4: The change in relevant feature count and performances during optimization of threshold value of ReliefF for behaviors in right.

shown over a two-dimensional image, where the darker areas correspond to relevant grids, and lighter areas correspond to irrelevant grids. As shown, only one region of the whole image is found to be relevant for each behavior. Moreover, the relevant grids shift from left to right, when the movement directions vary from left to right. For example, in *forward* movement, only the grids which locate in the middle of the range image are relevant, and none of these grids are relevant for a *turn-sharp-X* behavior.

Table 6.2: The optimized threshold values and the number of relevant features.

Behavior	Optimized threshold	Number of relevant features
	0.025	121
	0.03	106
	0.01	341
	0.0175	146
	0.0125	250
	0.0125	355
	0.02	207

Learned Relevant Features

In order to determine exactly which features are relevant, the relevancy is examined by grouping features as $i)$ distance related ones, $ii)$ shape related ones in φ channel, and $iii)$ shape related ones in θ channel. For the *move-forward* behavior, 12% of the relevant features are distance related ones, 66% are the frequencies from φ channel, and 22% are the frequencies from θ channel. The relevant intervals are $[80°, 180°]$ for φ and $[-20°, +20°]$ for θ channels. This can be interpreted as the vertical shape of the objects are more important than horizontal shapes in determining their traversibility affordances. This agrees with the physical properties of the objects and their relation to affordances. and have different shapes in horizontal, but same affordances. But the shape in vertical distinguishes the traversibilities of the objects , , and .

Efficiency in affordances perception

The economy in affordance perception is acquired in three levels:

- **Sensor level:** The 3D laser scanner is parameterized based on the vertical range of relevant grids. The bottom and top regions, which correspond to the base body of the robot, and sky (or ceiling) are found to be irrelevant for traversing

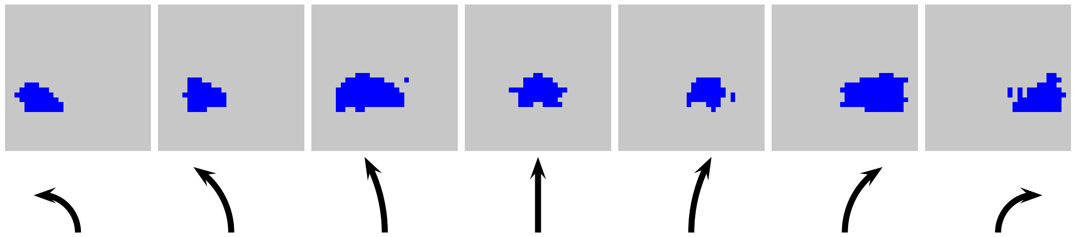


Figure 6.5: The relevant grids in the range image for each behavior. From left to right: *Turn-Sharp-left*, *Turn-Left*, *Turn-Smooth-Left*, *Move-Forward*, *Turn-Smooth-Right*, *Turn-Right*, *Turn-Sharp-Right*.

over the ground (Figure 6.6). In order to obtain the horizontal band in the range image, it is adequate to make a vertical scan between angles $[50° - 90°]$, saving 76.6% scanning time.

- **Feature extraction level:** Since 2D scanning of the laser scanner could not be adjusted (ie. by defining a horizontal angular range), many irrelevant grids are also read. For example, 45 grids are relevant for *move-forward* behavior, and a total of $8 \times 30 = 240$ grids are scanned. Approximately 81% of these grids are not required to be processed to extract distance and shape features.

- **Affordance Prediction level:** After the features are computed over the relevant grids, only a small amount of them are used in predicting affordances. For example, 146 features among $45 \times 39 = 1755$ features are relevant for *move-forward* behavior. (39 is the number of features for each grid, and 45 is the number of relevant grids, mentioned in previous item). Thus, 91% of them are only employed in prediction level.

6.3 Learning in Simple Environment (for Evaluation of the Generalization Performance)

A total of 1000 samples are used in training and test trials. In each sample, only one randomly selected object is placed in front of the robot. Thus, the whole set is the union of samples taken for each object category:  ,  ,  , and . Table 6.3 summarizes how the objects are distributed in the whole sample set, and the success rate of the *move-forward* actions within the sets of object categories. As seen, while  object is always traversible,  and  objects do not afford traversibility for the

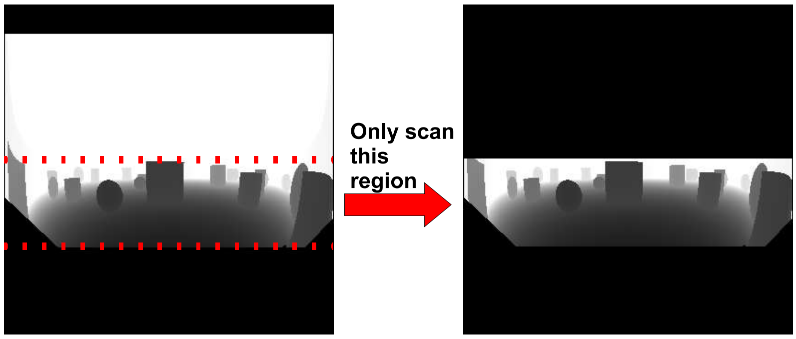


Figure 6.6: The irrelevant grids need not to be scanned. The left figure is the range image of a full scan and the right figure is the range image when the 3D scanner is adjusted to vertically scan only relevant grid region. The black rectangular and triangular shaped regions on the bottom of the left image are produced by the components of the robot. The limitation in rotation of the 3D laser scanner is the reason of the black band on the top of the left image.

Table 6.3: The distribution of the samples for *simple* environment

	Count	Success rate
	276	0%
	233	100%
	238	0%
	253	53%

simple environments, and is successfully traversed approximately half of the time.

As presented in Table 6.4, 16 different set of object categories are used to train 16 different models. This provides us a means of constraining the learning space to certain object types. For example, in case 0, no object is included into the training set, case 2 is trained only with box objects, case 6 with box and sphere, and in case 16, all objects are included while training. Next, each model is tested with all object categories one by one, and the prediction accuracy regarding to the traversibility affordances for that object categories' are computed. No sample is placed both in train and test sets in any situation. When there is overlapping in object categories in training and test sets, same sample is not allowed to be placed in both sets. Thus, in such situations, the samples of the same category are randomly selected for training and test sets.

When the training set includes only one category of affordances, the model predicts same affordance on all objects:

- In cases 2,4, and 7 training is performed with only not-traversible object, thus all test objects are perceived as not-traversible (completely false predictions for  and ).

- In case 3 only traversible objects are included into training set, thus all test objects are perceived as traversible. (completely false predictions for  and , and partly false for ).

As presented in the table, the method is able to predict the affordances of the novel objects when the training set includes samples from traversed and non-traversed situations:

- Although in case is trained with only  object, it is able to predict the affordances of all other objects that are not included in training set with high accuracy. Cases 8, 10, 11, 13, 14, 15, and 16 are similarly successful in prediction of novel objects because  is included in the training of these cases.

- The remaining cases 6, 9, and 12 which do not include  object in their training set are found to be successful in predicting the affordances of novel objects include  object.

The generalization ability of the model that is trained in case 5 is especially very successful. Although only one object, , was included in the training set, all other object affordances (, , ) are correctly predicted.

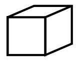
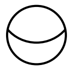
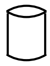
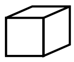
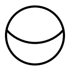
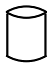
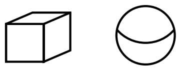
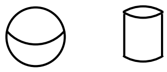
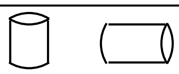
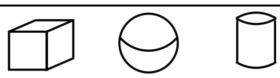
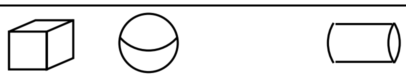
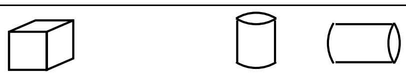
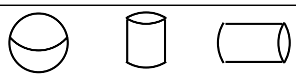
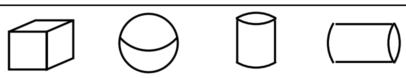
Case 6 also provides similar results. In this case,  and  are included in the training set, where traversibility is never afforded and always afforded, respectively. Since this training set includes samples for both success and fail, the affordances of novel objects ( and ) are correctly predicted.

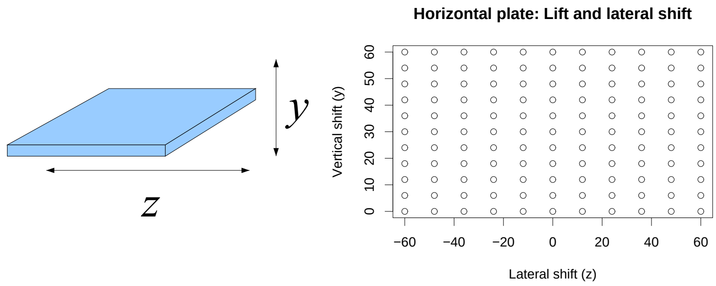
6.4 Systematic Analysis of the Learned Model

In this section, a number of experiments are conducted to systematically analyze "what is really learned" during exploration trials. There are three sets of experiments conducted for this purpose:

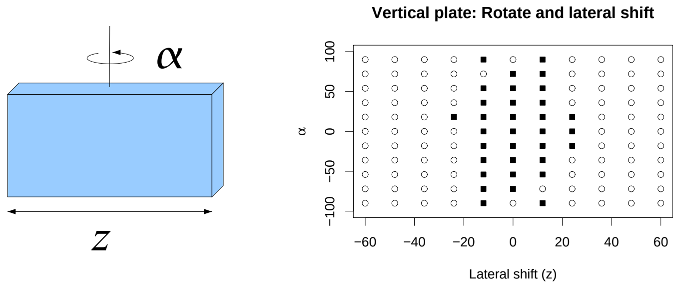
1. A very thin rectangular plate is used to analyze which surfaces of boxes were learned to be important in affordance perception.

Table 6.4: Generalization performance of the learned model. The left-most two columns show case number and the set of objects in the environment, where the corresponding model is trained. The second row shows which objects are included into the test sample set, where each set contains only one object category. For each of the given training set, and test object, the accuracy of the learned model's predictions are given in the rest of the table.

Case	Training objects	Accuracy in prediction			
					
1		100	0	100	53.4
2		100	0	100	53.4
3		0	100	0	46.6
4		100	0	100	53.3
5		100	83.8	100	94.7
6		100	100	100	86.4
7		100	0	100	53.4
8		100	83.8	100	95.6
9		99.2	100	100	85.9
10		100	100	100	93.8
11		100	83.8	100	94.7
12		100	100	100	86.4
13		100	100	100	95.6
14		100	83.8	100	95.6
15		100	100	100	94.7
16		100	100	100	94.7



(a) Parallel to the ground



(b) Perpendicular to the ground

Figure 6.7: The results of the experiments, which are conducted with a thin plate. Filled squares for not afforded situations, and circles for afforded ones.

2. All (, , , and) objects are shifted in two different axis, and the change in their affordances are analyzed.

3. The role of the horizontal cylinder's () orientation with respect to the robot, will be inspected.

For all experiments, the prediction of the traversibility for *move-forward* behavior is studied. These objects are placed, either shifted or rotated in various amounts. The robot then applies its standard procedure to perceive their affordances: i) a 3D scan of the environment is done, ii) irrelevant features for *move-forward* behavior are filtered out, iii) the affordance is predicted based on the learned SVM model. Next, the result of these experiments will be discussed.

Experiments with the thin plate Box objects have many surfaces. These experiments are conducted to identify which surfaces are important in perception of the *not-traversibility* affordance for a box object. Thus, we employed a very thin plate, which is not included in learning trials. In this way, only one surface is perceived and used in detection of the traversibility, since there is only one plane. This plate is placed parallel and perpendicular to the ground surface, respectively, as shown in Figure 6.7.

The parallel plane is shifted in two axis, one movement makes a vertical shift and the other is in lateral axis with respect to the robot. The object is lifted at most 60 cm, and it is shifted in $[-60, 60]$ cm range. Both extremes make the object disappear from the relevant region for *move-forward* behavior, since the robot has a width of 35 cm. However, the parallel plate, independent of its vertical or lateral position, always predicted to afford traversibility in the forward direction. We think that the similarity between the ground and parallel plate affected this result. During the exploration, the ground was learned to provide traversibility affordance.

In the second set of experiments, the thin plate is placed perpendicular to the ground surface. It is rotated around its vertical axis in $[-90°, +90°]$ range, and shifted in lateral direction. As shown in the Figure 6.7 (b), although the robot has not encountered with such a thin object before, the affordance of this object are correctly predicted. In this figure, the middle columns correspond to the objects that are placed in front of the robot, such that some parts of these objects impede the forward action. When inspected in detail, one point in the 4^{th} column and three points in the 8^{th} column do not afford traversibility, unlike the majority in these columns. Since the orientation angle of the object in these points is approximately $0°$, and this angle corresponds to a situation where the face of the object is fully exposed to the movement path of the robot, the forward action is not afforded. Same discussion is valid for the minorities in 5^{th}, 6^{th}, and 7^{th} columns. In these situations, the object is hardly detected by the 3D laser scanner, since the plane becomes almost perpendicular to the robot in these cases. The two objects in Figure 6.8 (a) and (b) afford traversibility even they are on the path of the robot. In these cases, because of the orientations of the objects (with $\alpha = 90°$ and $\alpha = 72°$), they are simply not detected.

(a) $\alpha = 90°$, $z = 0$ (b) $\alpha = 72°$, $z = -0.12$ (c) $\alpha = 90°$, $z = -0.12$

Figure 6.8: The effect of the orientation of thin boxes. Since the number of laser beams that are reflected from the two objects (a) and (b) is very small, affordances of these objects are not correctly predicted. However, although the object in the (c) is rotated in the same amount with (a), it is detected by the laser scanner, since the relative orientations of the objects are important.

Experiments with standard objects, which are gradually shifted in different directions The motivation behind these experiments is to analyze $i)$ how the traversibility affordances are affected with the distances to the objects, and $ii)$ how the system is affected when never seen scenes (the lifted objects) are provided. In this respect, all object in different categories are separately placed in front of the robot. The four set of experiments for these objects are shown in Figure 6.9, where the vertical distance of these objects from the ground is in the range $[0 - 150]$ cm, and they are placed in $[30 - 150]$ cm range from the robot in longitudinal axis. In these experiments, the orientations of the objects with respect to robot, and on their own axis are fixed.

As seen from the figure, the cylinders and spheres afford the *move-forward* behavior, independent of their distances or vertical positions. The predictions are correct since spheres are always traversible. Likewise, in this setup where orientation is fixed, robot will push the cylinder from the same surface, and roll it, thus are also traversible. The effect of the cylinder's orientation on affordance perception will be inspected in the next experiment set. However, we should first examine the results obtained from other two objects.

Although they have different surface characteristics, the boxes and the cylindrical objects provide similar results for this experiment. When they are placed close to the robot, the environment does not afford the traversibility, and when they are out of the robot's movement range, environment becomes traversible. The triangular region of non-traversibility in both figures seem interesting. The objects, which are

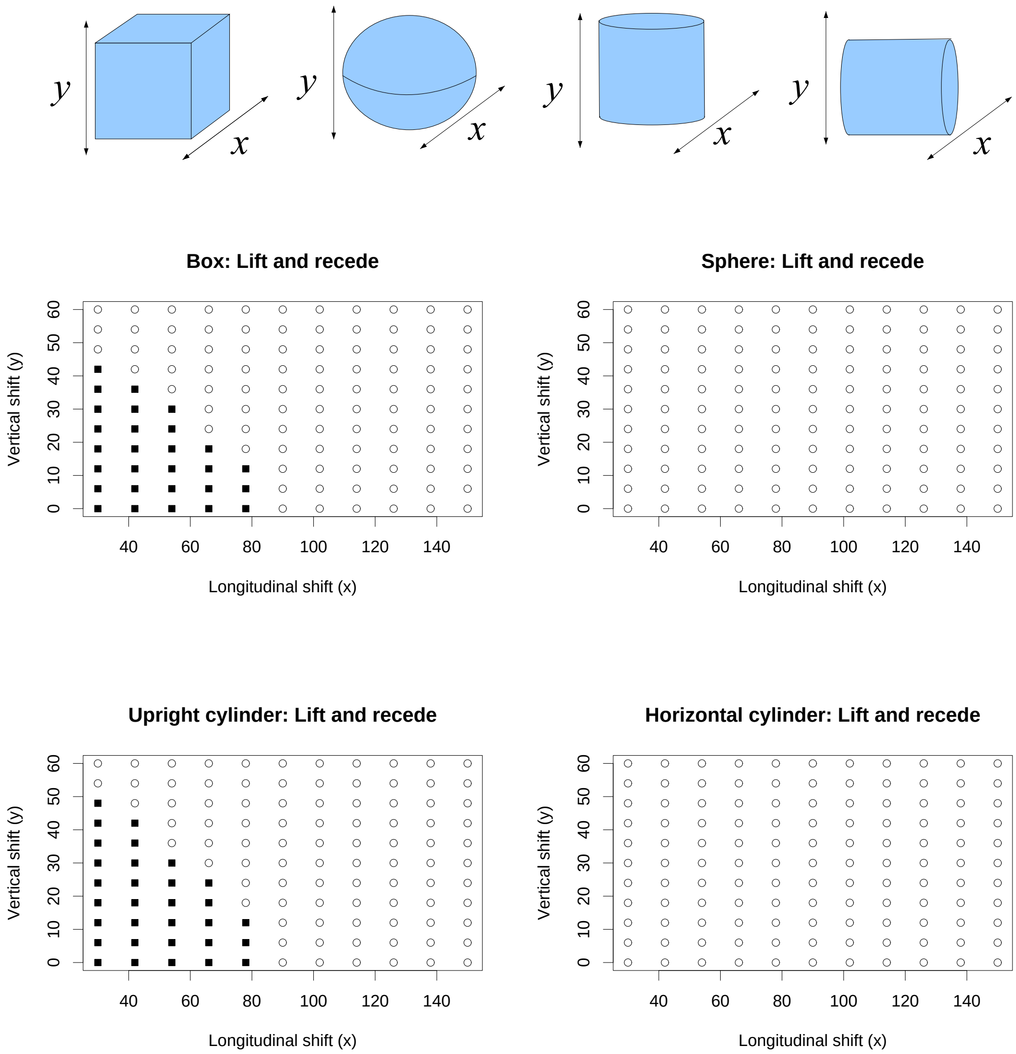


Figure 6.9: Affordances of standard objects in various positions. Filled squares for not afforded situations, and circles for afforded ones.

lifted same amount (in any column), afford traversibility when they are distant, and do not afford the action when they are close to the robot. Since, exploration trials do not contain any object that hang in the air, the system make wrong (and contradictory) predictions with these objects. Any object, that could be traversible when close to the robot, should also be traversible when it is a little further away, and vice versa.

We think that the wrong predictions (in such unrealistic environments where objects are hung in the air) have roots in robot's 3D scanning mechanism. Since it scans the environment in angular steps, the closer objects correspond to larger regions in the range image, and further objects generate smaller regions. Suppose that the *close*

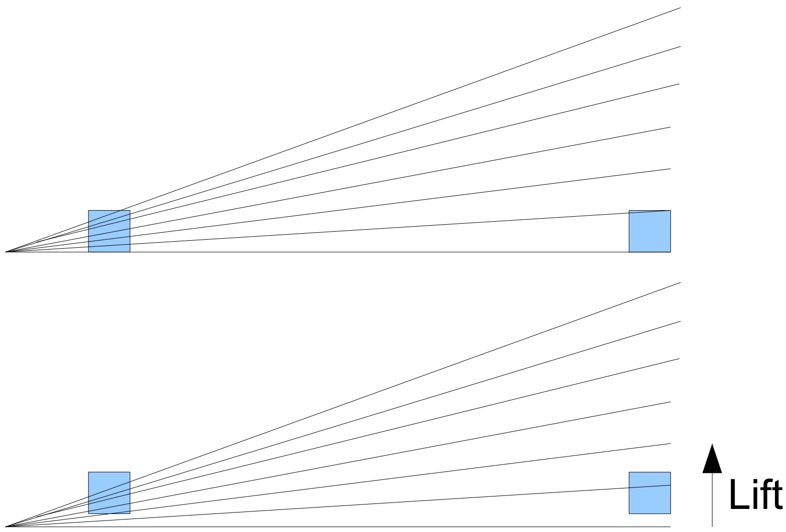
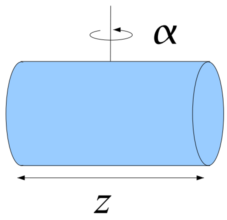


Figure 6.10: The effect of radial scanning. A sketch of how the perspective in laser scanning, affects the number of laser rays and grids that represent the *close* and *distant* objects.

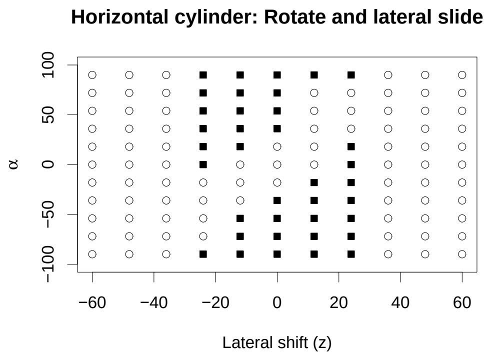
object is covered with n grids, and the distance object with only 1 grid in the most extreme case. This means that the perception of affordances of these objects are performed over n and 1 grids, respectively. When the *distant* object is lifted a small amount, it will be covered by n' grids, where $n > n'$ and $n' > 0$. As a result, some grids (n') and their representative feature values will remain same in *close* object, not degrading its prediction capability. However, when the *distant* objects are lifted in the same amount, since there was 1 grid before, the none of the feature values will remain same. In short, there is redundancy in the perception of *close* objects, and change in grid number did not affect the system. However, for the *distant* objects, the prediction highly depends on small number of grids (which are changed when lifting occurs).

Figure 6.10 demonstrates the geometrical sketch, where prior to lifting the object, *close* one have a region of vertically 6 grids width, and *distant* one have 1 grid on the range image. After they are lifted, 66% of these grids and their values would remain same in *close* object, and the 100% of the grid values, corresponding to the *distant* object will be changed.

The effect of horizontal cylinder orientation in affordance prediction The horizontal cylinder (), based on its orientation, might provide both traversibility and



(a)



(b)

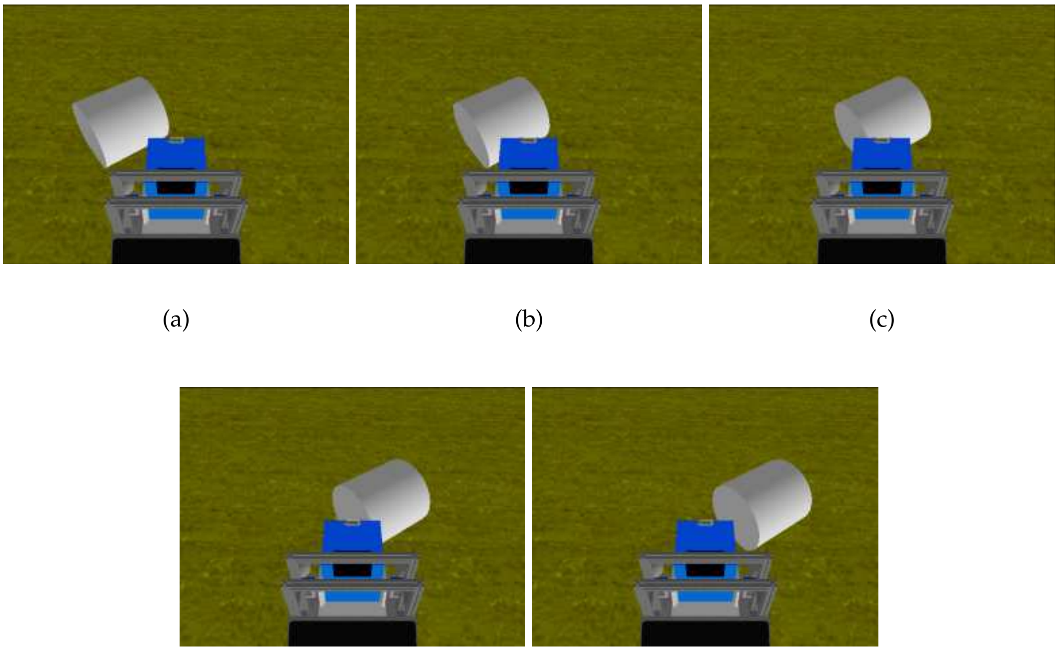
Figure 6.11: The affordances of cylindrical objects The objects are placed in various orientations with respect to the robot. Filled squares for not afforded situations, and circles for afforded ones.

not-traversibility affordances in different circumstances. The aim of these experiments is to examine whether the dynamics of the system is correctly learned. To do this, a cylindrical object, which lies over its circular surface, is placed in front of the robot. Later it is shifted in robot's lateral axis and rotated around its own vertical axis, in certain amounts. The predictions of the traversibility affordance for *move-forward* behavior are demonstrated in Figure 6.11.

As shown, the *move-forward* action is afforded when the object is not in the path of the robot. However, when it is on robot's movement path, the perception of the affordance changes based on object's orientation and position. When the cylindrical surfaces of the cylinders are in the robot's collision path, it becomes traversible. Figure 6.12 shows $36°$ rotated cylinders, which are shifted in the $[-0.24, +0.24]$. Since the robot makes contact with different sides of the cylinder, the traversibility affordance gradually disappears from (a) to (e).

6.5 Traversibility in a cluttered environment in MACSim

In all the previous experiments, the robot was placed in a fixed position, the objects around the robot are changed randomly, and the robot performs its actions only from this position. In this section, the robot which was trained in a complex environment, is placed in a virtual room, full of obstacles in different sizes and types. Then, in its



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

Figure 6.12: The effect of orientations of cylinders. By shifting the object in lateral axis, its orientation with respect to robot is changed.

execution mode, which was described in Section 5.4, the robot

- perceives the traversibility affordances of the environment,
- selects an afforded behavior where the priority is on "more " forward movements,
- performs its behavior for the fixed amount of time,
- stops and perceives the traversibility affordance for the changed environment

The trajectory of the robot in such a room, with 40 objects included, is shown in Figure 6.13. As shown, it successfully predicts the affordances of the spherical and lying cylindrical objects by driving towards them, and the boxes and upright cylindrical objects by avoiding them. The four affordance perception instances are identified in Figure 6.13 and their snapshots are demonstrated in Figure 6.14. [1].

[1] The movie of the robot navigation can be downloaded from `http://kovan.ceng.metu.edu.tr/~emre/virtual_room.mpg`

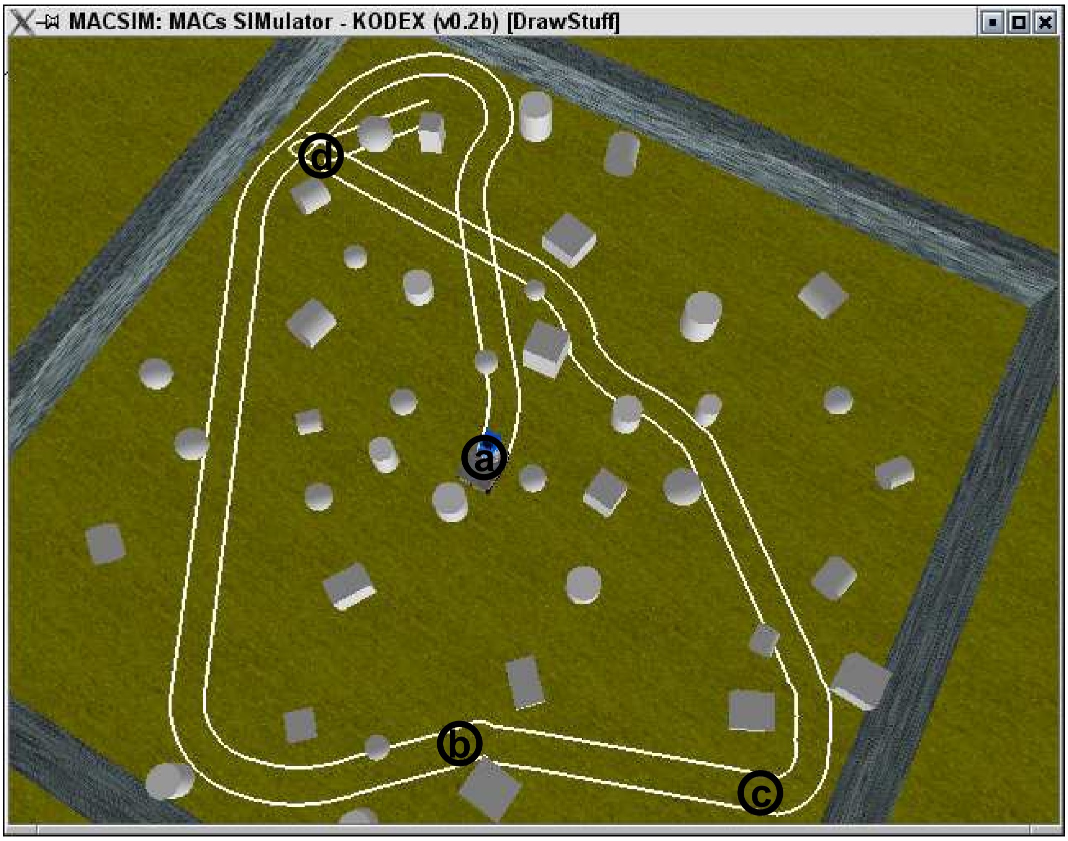
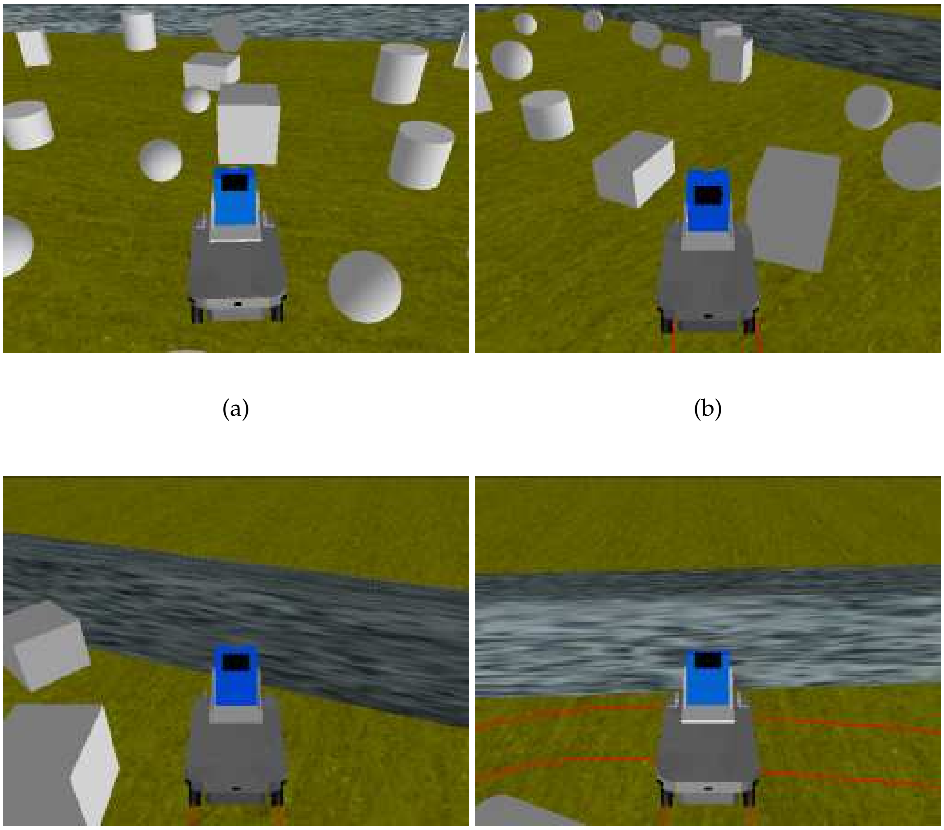


Figure 6.13: The execution of the robot is demonstrated The robot has a motivation to go forward as most as possible. In the figure, the trails of the wheels are demonstrated. The robot is able to successfully perceive the traversibility affordance of the environment, and selects behaviors accordingly.

When inspected in detail, the robot is generally able to avoid from the fixed objects. However there is a minor difference between interactions with cylinders and boxes. While the robot is able to pass the cylinders without any scratch, the performance decreases for box objects: The robot in many situations comes into contact with the edges of the boxes. We think that, the curved and sharp edges could not be differentiated by the robot with this feature set and learning setup. During learning trials, such small collisions that did not change the robot's path in large amounts, were labelled as successful trials. Likewise, as seen from the trials, the robot was always able to traverse the objects, and did not stuck in any point.



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

Figure 6.14: Encountered situations for cluttered room experiments. In (a) the *turn-left* behavior is afforded, and the robot drove towards the spherical object. In (b), although the robot made a contact with the right wall, it selected *move-forward* action. In (c), the only action that is afforded was turning left sharply, and *turn-sharp-left* is performed without any scratch. In (d), none of the behaviors were afforded, so the robot made a random turn. Note the slight difference between (c) and (d), where robot was able to find out the small open-space on the left in (c).

6.6 Perceiving Traversibility on the Real Robot Kurt3D

The Kurt3D robot is used to validate the results in the real world where the robot was trained in MACSim complex environment. Various objects, including simple geometrical ones, and office environment object like trash bins, and PC cases are placed in the frontal area of the robot. Two sets of experiments are conducted where boxes are used in the first set and cylinders are employed in the second one.

As shown in Figure 6.15, the robot is able to correctly perceive the affordances of box shaped objects. The traversibility affordances of the objects, which are placed in

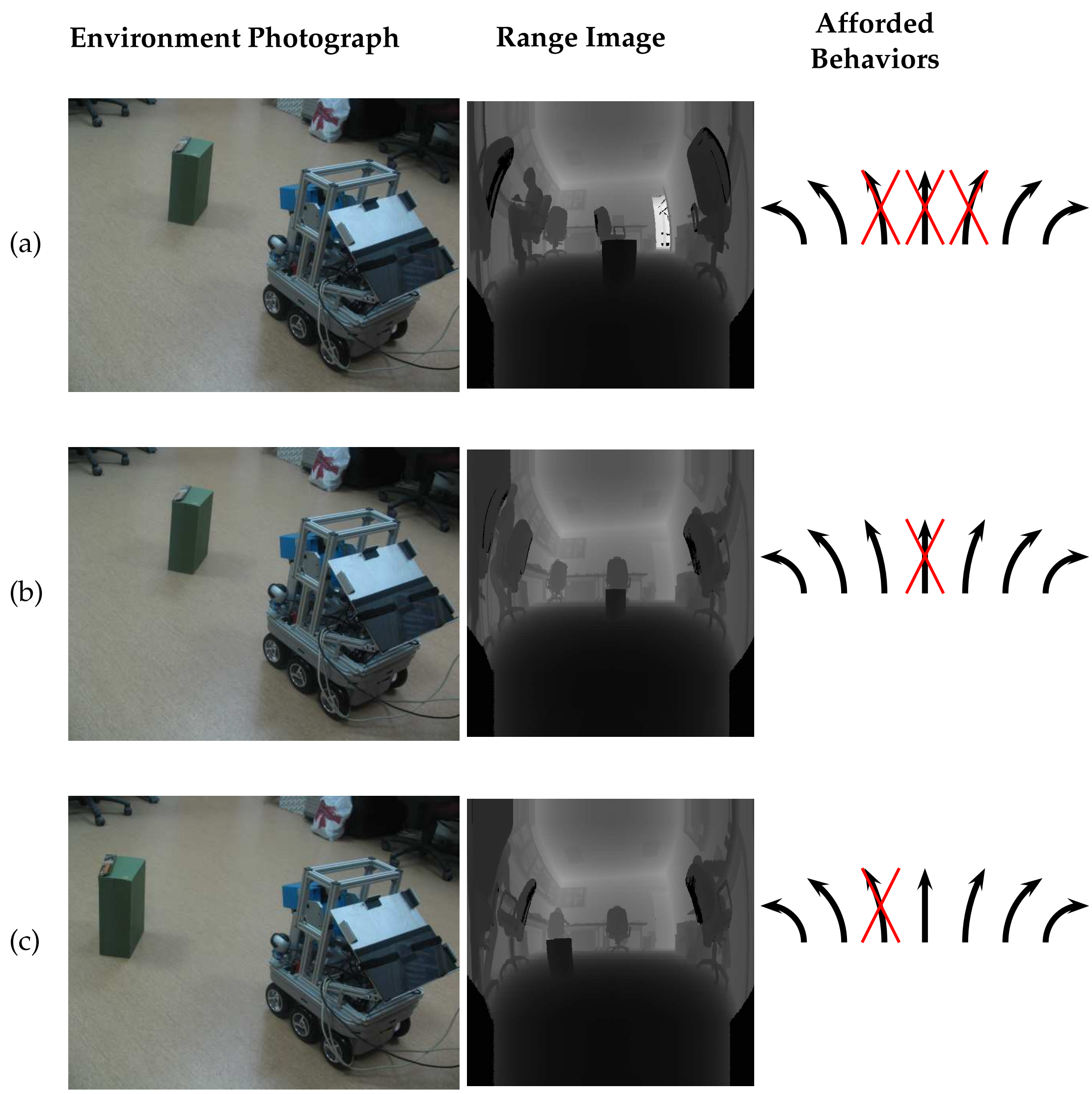


Figure 6.15: The real world experiments with box shaped object.

different distances and angles, are correctly perceived (a,b,c). Moreover, the performance of these predictions are not affected when more objects are included into the environment (d).

The effect of the rotation of cylinders are also studied in real world experiments (Figure 6.16). The robot is found to be successful in predicting the affordances in various orientations. Without training, the robot through simulated physical interactions, designing the affordances by hand would be very hard brittle especially for these situations.

Lastly, the perception of the affordances for different width apertures are studied.

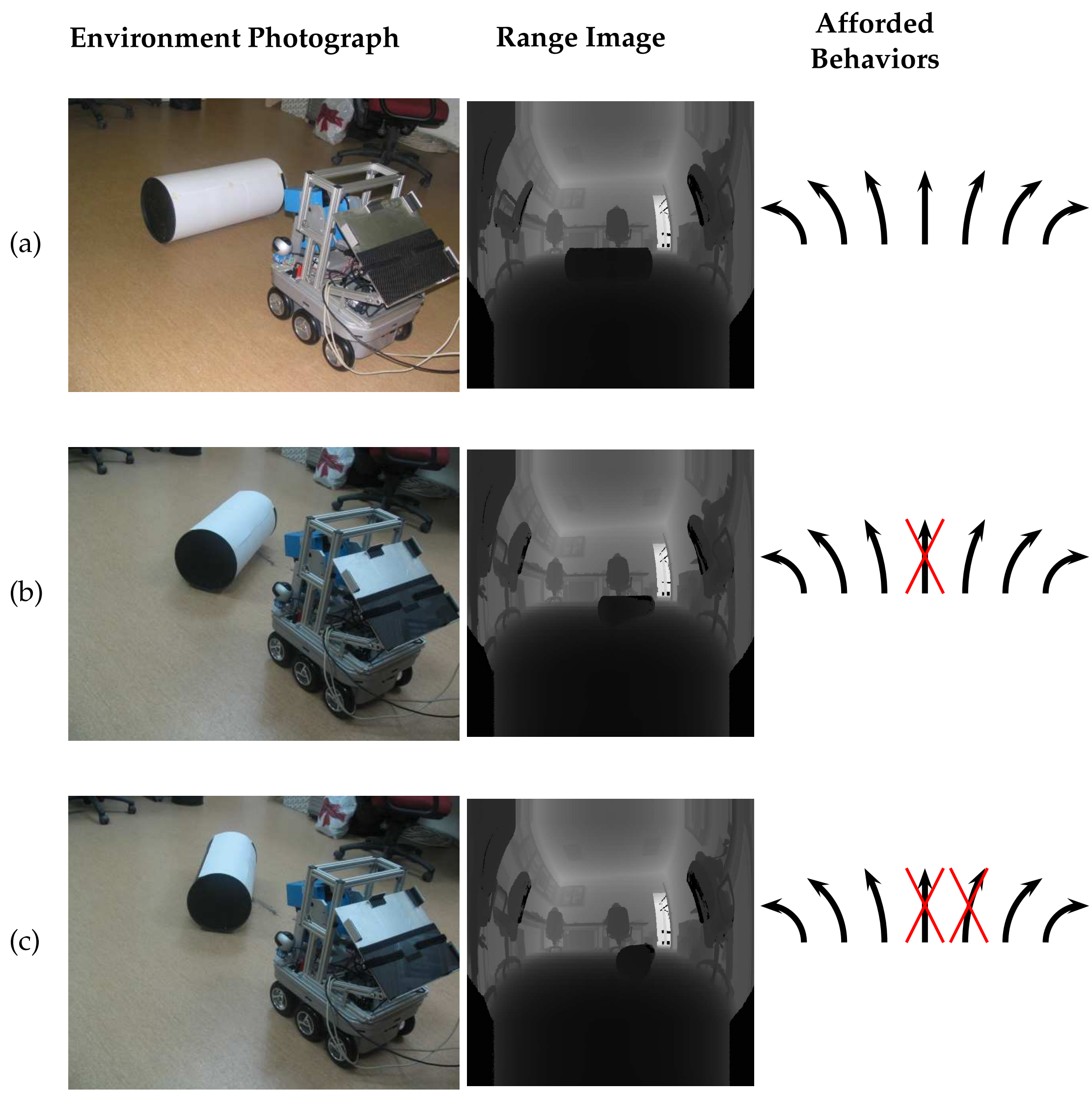


Figure 6.16: The real world experiments with cylindrical shaped objects.

The successful results obtained from these experiments (Figure 6.17) was unexpected. The robot has no notion of width, but it is able to perceive the traversible aperture widths.

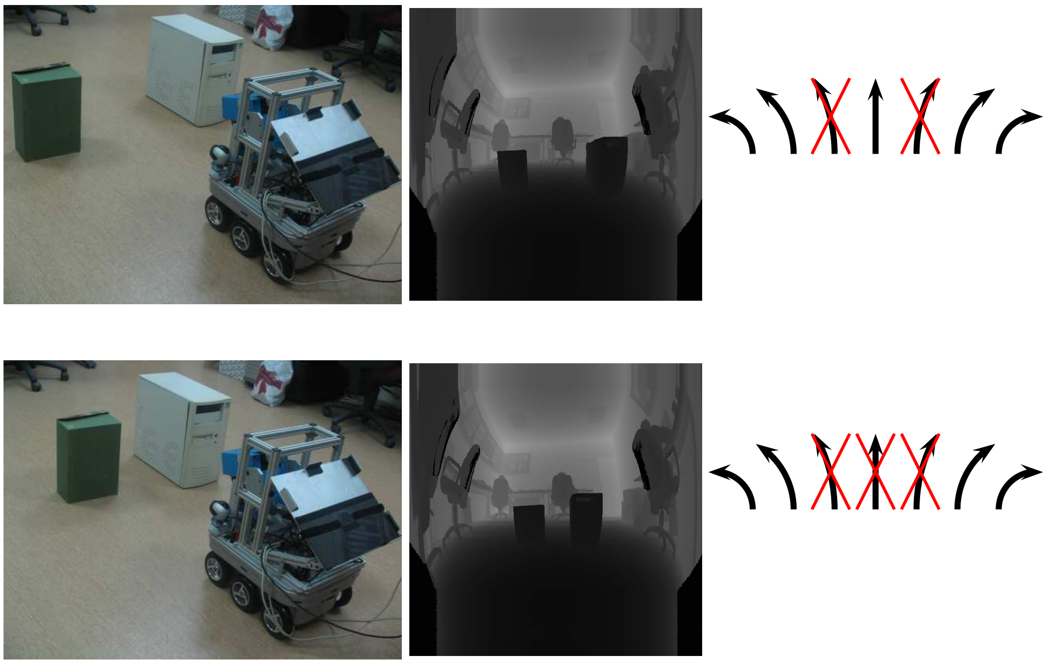


Figure 6.17: The real world experiments with two boxes. In this experiment, it is shown that when the aperture width is changed, the affordance of traversibility disappears. The robot has no notion of "width" of the objects, and it does not even know its own width. The robot is able to correctly predict its own aperture width and successfully perceive the affordances of apertures in various widths.

CHAPTER 7

CONCLUSION

In this thesis, traversability affordances of the environment for a mobile robot is learned through physical interactions in a physics based simulation environment. Since the traversability depends on the location of the objects and their geometrical properties, range images are used to perceive the physical affordances of the immediate environment. A simple perceptual representation is proposed, where intermediate high-level processes like object detection or world modeling are not utilized, thus favoring Gibsonian direct perception view. Since complex actions which require higher level processes are not in the scope of this study, perception is used solely to select a low-level behavior, and it might be utilized as the lower layer of a layered perceptual system in future.

Based on the low-level features that are perceived and the results of the interactions with the world, the robot is able to learn i) relevant features for different actions, and ii) the affordances provided. The prediction accuracy in perceiving the traversability affordances of the environment, which include several boxes, cylinders, and spheres is found to be around 95%. Furthermore, it is presented that the robot uses only 1.1% of the extracted features while perceiving the affordances. This in turn save the time 76.6% in scanning and 81% in feature processing, and J.J. Gibson's *perceptual economy* is obtained through learning to use relevant features.

After learning the affordances of the environment, the robot is tested in various setups. It is placed in a virtual cluttered room, and controlled with a simple motivation system. In this environment, the robot was able to traverse the environment, by successfully selecting its actions based on the perceived affordances. In the next experiment set, the generalization performance of the learned affordance based perception system is analyzed. It is shown that the robot was able to perceive the traversability affordances of the novel objects that it has never seen before. In the last set of experiments, the affordance-based action selection scheme that is learned in simulator is successfully transfered to real robot without any further modification. Although there is no concept of *object* or *width* in any representation level, and the robot has no awareness of its own body dimensions, it is able to perceive the traversability affordances of the apertures between the objects. In other words, the affordances of the apertures, which depend on the relation between the width of the apertures and the shoulder width of the robot, are *directly perceived* without recognizing them.

REFERENCES

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

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

[3] K. Jones, "What is an affordance?," *Ecological Psychology*, vol. 15, no. 2, pp. 107–114, 2003.

[4] E. Gibson, "Where is the information for affordances?," *Ecological Psychology*, vol. 12, no. 1, pp. 53–57, 2000.

[5] J. Gibson, *The senses considered as perceptual systems*. Boston: Houghton Mifflin, 1966.

[6] T. Stoffregen, "Affordances as properties of the animal environment system," *Ecological Psychology*, vol. 15, no. 2, pp. 115–134, 2003.

[7] A. Chemero, "An outline of a theory of affordances," *Ecological Psychology*, vol. 15, no. 2, pp. 181–195, 2003.

[8] W. Warren, "Perceiving affordances: Visual guidance of stair climbing," *Journal of Experimental Psychology*, vol. 105, no. 5, pp. 683–703, 1984.

[9] W. Warren and S. Whang, "Visual guidance of walking through apertures: body-scaled information for affordances," *Journal of Experimental Psychology*, vol. 13, no. 3, pp. 371–383, 1987.

[10] L. Mark, "Eyeheight-scaled information about affordances: A study of sitting and stair climbing," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 13, no. 3, pp. 361–370, 1987.

[11] E. Gibson, G. Riccio, M. Schmuckler, T. Stoffregen, D. Rosenberg, and J. Taromina, "Detection of the traversability of surfaces by crawling and walking infants," *Journal of Experimental Psychology*, vol. 13, no. 4, pp. 533–544, 1987.

[12] J. Kinsella-Shaw, B. Shaw, and M. Turvey, "Perceiving walk-on-able slopes," *Ecological Psychology*, vol. 4, no. 4, pp. 223–239, 1992.

[13] A. Chemero, "What events are," *Ecological Psychology*, vol. 12, no. 1, pp. 37–42, 2000.

[14] R. Oudejans, C. Michaels, B. vanDort, and E. Frissen, "To cross or not to cross: The effect of locomotion on street-crossing behavior," *Ecological Psychology*, vol. 8, no. 3, pp. 259–267, 1996.

[15] S. Cornus, G. Montagne, and M. Laurent, "Perception of a stepping-across affordance," *Ecological Psychology*, vol. 11, no. 4, pp. 249–267, 1999.

[16] A. Chemero, C. Klein, and W. Cordeiro, "Events as changes in the layout of affordances," *Ecological Psychology*, vol. 15, no. 1, pp. 19–28, 2003.

[17] A. Szokolszky, "An interview with Eleanor Gibson," *Ecological Psychology*, vol. 15, no. 4, pp. 271–281, 2003.

[18] E. Gibson, "Perceptual learning in development: Some basic concepts," *Ecological Psychology*, vol. 12, no. 4, pp. 295–302, 2000.

[19] U. Neisser, *Cognition and Reality: Principles and Implications of Cognitive Psychology*. 0716704773, W.H. Freeman and Co, 1976.

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

[21] J. Norman, "Ecological psychology and the two visual systems: Not to worry!," *Ecological Psychology*, vol. 13, no. 2, pp. 135–145, 2001.

[22] R. R. Murphy, *Introduction to AI Robotics*. MIT Press, 2000. ISBN:0262133830.

[23] R. Murphy, "Case studies of applying Gibson's Ecological Approach to Mobile Robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 29, no. 1, pp. 105–111, 1999.

[24] Y. Aloimonos, C. Fermüller, and A. Rosenfield, "Seeing and understanding: Representing the visual world," *ACM Computing Surveys*, vol. 27, no. 3, pp. 307–309, 1995.

[25] A. Duchon, W. Warren, and L. Kaelbling, "Ecological robotics," *Adaptive Behavior*, vol. 6, no. 3, pp. 473–507, 1998.

[26] R. Cooper and D. W. Glasspool, "Learning action affordances and action schemas," in *Connectionist Models of Learning, Development and Evolution* (R. M. French and J. P. Sougne, eds.), Perspectives in Neural Computing, (London), pp. 133–142, Sixth Neural Computation and Psychology Workshop, Springer-Verlag, 2001.

[27] I. Cos-Aguilera, L. Canamero, and G. 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.

[28] 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.

[29] K. MacDorman, "Responding to affordances: Learning and projecting a sensorimotor mapping," in *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, (San Fransisco, California, USA), pp. 3253–3259, 2000.

[30] 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*, pp. 3140–3145, 2003.

[31] A. Nüchter, K. Lingemann, J. Hertzberg, H. Surmann, K. Pervölz, M. Hennig, K. R. Tiruchinapalli, R. Worst, and T. Christaller, "Mapping of rescue environments with kurt3d," in *Proceedings of the International Workshop on Safety, Security, and Rescue Robotics SSRR 2005*, (Kobe, Japan), pp. 158–163, June 2005. ISBN: 0-7803-8946-8.

[32] J. M. Rodrigez, A. P. Ruz, S. Pedreza, and F. D. Fernandez, "Object following and obstacle avoidance using a laser scanner in the outdoor mobile robot auriga-a," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, (Canada), pp. 204–209, 1998.

[33] S. Roumeliotis and G. Bekey, "Segments: A layered, dual-kalman filter algorithm for indoor feature extraction," in *In Proc. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Takamatsu, Japan), pp. 454–461, 200.

[34] Y. D. Kwon and J. S. Lee, "A stochastic map building method for mobile robot using 2-d laser range finder.," *Auton. Robots*, vol. 7, no. 2, pp. 187–200, 1999.

[35] B. Giesler, R. Graf, R. Dillmann, and C. F. Weiman, "Fast mapping using the log-hough transformation," in *IEEE Conference on Intelligent Robots and Systems (IROS)*, 1998.

[36] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.

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

[38] S. Gächter, "Results on range image segmentation for service robots," Tech. Rep. EPFL-LSA-2005-01, Laboratoire de Systèmes Autonomes, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, 2005.

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

[40] I. Kononenko, "Estimating attributes: analysis and extensions of relief," in *ECML-94: Proceedings of the European conference on machine learning on Machine Learning*, (Secaucus, NJ, USA), pp. 171–182, Springer-Verlag New York, Inc., 1994.

[41] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Mach. Learn.*, vol. 53, no. 1-2, pp. 23–69, 2003.

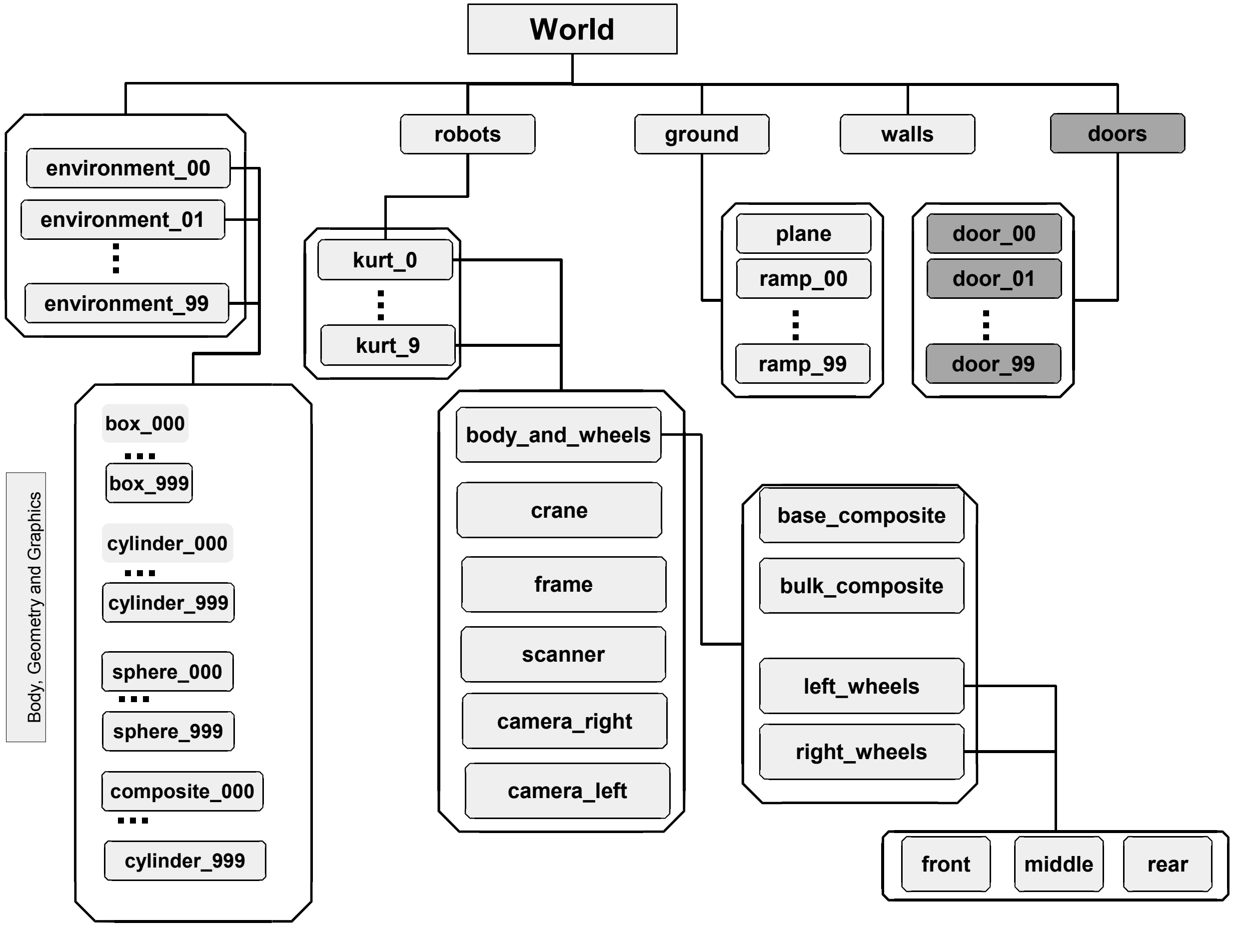
[42] J. Bins and B. A. Draper, "Feature selection from huge feature sets.," in *ICCV*, pp. 159–165, 2001.

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

[44] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

APPENDIX A

XML FILE HIERARCHY


World
environment_00
environment_01
environment_99
robots
kurt_0
kurt_9
ground
plane
ramp_00
ramp_99
walls
doors
door_00
door_01
door_99
box_000
box_999
cylinder_000
cylinder_999
sphere_000
sphere_999
composite_000
cylinder_999
body_and_wheels
crane
frame
scanner
camera_right
camera_left
base_composite
bulk_composite
left_wheels
right_wheels
front
middle
rear
Body, Geometry and Graphics