TOWARDS LEARNING AFFORDANCES:

DETECTION OF RELEVANT FEATURES AND CHARACTERISTICS FOR REACHABILITY

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SELDA EREN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF INFORMATION SYSTEMS

JANUARY 2006

Approval of the Graduate School of Informatics

____

Assoc. Prof. Dr. Nazife Baykal

Director

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

____

Assoc. Prof. Dr. Onur DEMİRÖRS

Head of Department

This is to certify that we have read this 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. Semih BİLGEN	(METU, EEE)	________________________
Assist. Prof Dr. Erol ŞAHİN	(METU,CENG)	________________________
Assoc. Prof. Dr. Yasemin YARDIMCI	(METU, II)	________________________
Assoc. Prof. Dr. Göktürk ÜÇOLUK	(METU, CENG)	________________________
Assist. Prof. Dr. Erkan MUMCUOĞLU	(METU, II)	________________________

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, Surname : Selda Eren

Signature :

ABSTRACT

TOWARDS LEARNING AFFORDANCES:
DETECTION OF RELEVANT FEATURES AND CHARACTERISTICS FOR REACHABILITY

Eren, Selda
M.S., Department of Information Systems
Supervisor: Assist. Prof. Dr. Erol Şahin

January 2006, 55 pages

In this thesis, we reviewed the affordance concept for autonomous robot control and proposed that invariant features of objects that support a specific affordance can be learned. We used a physics-based robot simulator to study the reachability affordance on the simulated KURT3D robot model. We proposed that, through training, the values of each feature can be split into strips, which can then be used to detect the relevant features and their characteristics. Our analysis showed that it is possible to achieve higher prediction accuracy on the affordance support of novel objects by using only the relevant features. This is an important gain, since failures can have high costs in robotics and better prediction accuracy is desired.

Keywords: Affordance, Autonomous Robotics, Cognitive Robotics, Machine Learning, Ecological Psychology

ÖZ

*SAĞLARLIK*LARI ÖĞRENMEYE DOĞRU:
ERİŞİLEBİLİRLİK İÇİN İLGİLİ ÖZNİTELİKLERİN VE AYIRICI ÖZELLİKLERİN
ORTAYA ÇIKARILMASI

Eren, Selda
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Yar. Doç. Dr. Erol Şahin

Ocak 2006, 55 sayfa

Bu tezde, otonom robot kontrolü için *sağlarlık* kavramını inceledik ve objelerin belirli bir *sağlarlığı* destekleyen değişmez özelliklerinin öğrenilebileceği önerdik. Simule edilmiş KURT3D robot modeli üzerinde 'erişilebilirlik' *sağlarlığını* incelemek için fizik tabanlı bir simulatör kullandık. Her bir özellik değerinin çalışma yoluyla bantlara ayrılabileceğini ve daha sonra bu bantların ilgili özellikleri ve karakteristiklerini ortaya çıkarmak için kullanılabileceğini önerdik. Analizimiz, yalnızca ilgili özellikleri kullanarak daha önce karşılaşılmamış objelerin *sağlarlığının* yüksek doğrulukla tahmin edilebileceğini gösterdi. Robot biliminde başarısızlığın maliyeti yüksek olabileceğinden ve daha isabetli tahminlere ihtiyaç duyulduğundan bu kazanım önemlidir.

Anahtar Kelimeler: Sağlarlık, Otonom Robotbilim, Bilişsel Robotbilim, Makine Öğrenmesi, Ekolojik Psikoloji

To my family

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Erol Şahin for introducing me to the world of Robotics, for letting me work within an excellent research group and for his guidance and care throughout the whole thesis preparation process.

I would like to present my thanks the Faculty of Informatics Institute. They provide a friendly and motivating environment, and it is a wonderful place to pursue an academic degree with the facilities they provide.

Throughout my studies, my friends in Informatics Institute provided support with their invaluable friendship. From the beginning of my Masters studies, we had many fruitful discussions with Baki Kocaballi. I would also like to thank my friends Birsen Gülden and Beray Gençsoy for their support and love. Special thanks to Berna Bakır, for not leaving me alone on the final week of thesis preparation.

I have also gained friends from the Kovan Research Lab, thanks to the MACS Project. I would like to thank all of them for being supportive and friendly. By the way, they are the loveliest engineers I have ever seen.

I would like to express my deepest gratitude to my parents for their love, encouragement, and support throughout my education.

This work is partially supported by the MACS (Multi-sensory Autonomous Cognitive Systems interacting with dynamic environments for perceiving and using Affordances) project supported by the European Commission under IST-2003-2.3.2.4 Cognitive Systems (Project Number: FP6-004381).

TABLE OF CONTENTS

PLAGIARISM ii
ABSTRACT iii
ÖZ iv
DEDICATION v
ACKNOWLEDGMENTS vi
TABLE OF CONTENTS vii
LIST OF TABLES ix
LIST OF FIGURES x
CHAPTER
1. INTRODUCTION 1
1.1. Autonomous robotics 2
1.2. Cognitive Robotics 4
2. AFFORDANCES 8
2.1. Theory of Affordances 8
2.2. Related Work 11
3. AFFORDANCE PERCEPTION 15
3.1. An Affordance Perception Method 17
4. EXPERIMENTAL FRAMEWORK 23
4.1. Experimental Framework 23
4.1.1. MACSIM – The simulator 24
4.1.2. Simulated Robot Model 26
4.2. Reachability affordance 26
4.3. Implementation 27
4.3.1. Scenario 27
4.3.2. Initial experimentation 28
5. EXPERIMENTAL RESULTS 34
5.1. Strip formation 35

5.2. Affordance Perception 37
5.2.1. Using relevant features without threshold (STRIP-PASS)........... 38
5.2.2. STRIP-PASS with threshold 38
5.2.3. WINNOW Algorithm 40
5.2.4. WINNOW Algorithm with threshold.................................... 44
5.2.5. Comparison of the four methods 46
6. CONCLUSION 49
REFERENCES 51

LIST OF TABLES

TABLE

1. Features used in the experiments. 29

2. Decision method for the reachability affordance. For the test objects, value of the relevant features are examined. 37

LIST OF FIGURES

FIGURE

1. An affordance is relative to the capabilities of the agent. 10
2. The experiments in studies of Fitzpatrick and Metta et. al. The robot learns direction maps for different hand positions by observing initial hand positions and direction of object movement. 11
3. A snapshot from the experiments conducted on simulation by Stoytchev. The robot learns tools affordances by applying a set of exploratory behaviors and observing the results. 12
4. There are relevant features among the other features in the stimulus that support certain affordances. In this figure, the objects afford blockage to the robot. The robot can obtain many features from the stimulus it receives, such as color, distance, area, shape, and etc. But there is an invariant feature among these features that gives information about blocking affordance. It is the height of the objects; no matter what the color and shape of the object is, an object of a specific height will afford blocking to the robot. 18
5. Feature values are calculated from the visual stimulus obtained from the real world entities. The values are normalized and sorted after the robot experiments with a number of entities. v_n^{min} represents the minimum feature value for the n^{th} feature. v_n^{max} represents the maximum feature value for the n^{th} feature. 19
6. Affordance values for each feature value. 21
7. An example of feature strip formation. The large strip in $Feature^2$ is a candidate for being a positively relevant strip and thus $Feature^2$ is a candidate for a positively relevant feature. Similarly, the large strip in $Feature^n$ is a candidate for being a negatively relevant strip and thus $Feature^n$ is a candidate for a negatively relevant feature. 22

8. KURT3D robot platform. KURT3D is a 6 wheel, differential drive mobile robot. It is equipped with a 3D laser scanner (the large cube at the front of the robot), two pan-tilt cameras (small spheres), and eight infrared proximity sensors around the robot body. A crane, which will allow the robot to handle objects, is already designed and will be installed. The robot is controlled by a piggy-back laptop. ... 24
9. A snapshot from the MACSIM simulator. The small windows on the upper left and right corners display the left and right camera views respectively. Note that, the simulated robot model also included the crane which was not shown on the picture of the KURT3D robot..... 25
10. The robot 25
11. Reach behavior in infants has been an attractive phenomenon for human development research. 27
12. (a) The range of random object placement (b) One object is presented at a time 28
13. Reach behavior (a) Detect position of the object (b) Orient crane arm (c) Lower the magnet 31
14. In (a) and (b) reach behavior is successful. (c) and (d) shows the failure case. (a) and (c) are left , (b) and (d) are right camera views. 32
15. Scatter plot diagram of some of the feature values for the success and failure cases. Black triangles represent successful cases and grey rounds represent fail cases. 35
16. Strips formed after training the robot with 100 objects. Black areas represent strips that are positively relevant to the reachability affordance. Grey areas represent negatively relevant features. Corresponding feature names are stated under each column. Features related with crane color, crane area, crane distance, rope tension and rope length are not shown in this figure since their values are constant for all objects. For space reasons, features that are not very relevant are omitted from the figure. 36
17. The effect of number of relevant features on the true positive and true negative percentages for the STRIP-PASS method. 39

18. The effect of number of relevant features on the number of true positives and true negatives for the STRIP-PASS method. 39
19. The effect of number of relevant features on the number of positive and negative responses for the STRIP-PASS method. 39
20. The effect of number of relevant features on the true positive and true negative percentages for the STRIP-PASS method with threshold. 41
21. The effect of number of relevant features on the number of true positives and true negatives for the STRIP-PASS method with threshold. 41
22. The effect of number of relevant features on the number of positive and negative responses for the STRIP-PASS method with threshold.. 41
23. The effect of number of relevant features on the true positive and true negative percentages for the WINNOW algorithm. 43
24. The effect of number of relevant features on the number of true positives and true negatives for the WINNOW algorithm. 43
25. The effect of number of relevant features on the number of positive and negative responses for the WINNOW algorithm. 43
26. The effect of number of relevant features on the true positive and true negative percentages for the WINNOW algorithm with threshold. 45
27. The effect of number of relevant features on the number of true positives and true negatives for the WINNOW algorithm with threshold. 45
28. The effect of number of relevant features on the number of positive and negative responses for the WINNOW algorithm with threshold. . 45
29. Comparison of relevant feature utilization methods with respect to the true positive and true negative ratios. 47
30. Comparison of relevant feature utilization methods with respect to the number of true positives and true negatives. 47
31. Comparison of relevant feature utilization methods with respect to the number of positive and negative responses. 48

CHAPTER 1

INTRODUCTION

The dream of building robots that can assist us in our lives is coming true day by day. Industrial robots have already been successfully deployed in factory environments to do tasks such as assembly and painting with high speed and precision. These robots, which can be considered as programmable universal manipulators, have already been essential elements of industrial production. However, these robots require highly structured operating environments and high-precision actuators, and therefore can not operate outside of factory floors.

Tele-operated robots, robots that are operated by a human operator from a distance, have also been capturing the headlines in their role for inter-planetary exploration[1] or mine removal[2]. With assistance from humans, these robots assist us to explore dangerous environments, or to rescue people under wrecks. These robots send sensor data obtained from the environment to a human operator. The operator, in return, sends commands to the robot making it act on the environment with its actuators. Such operations keep the humans away from dangerous environments. In tasks, where communication between the operator and the robot is slow, such as inter-planetary exploration case, the operator can send high-level commands to the robot, such as ‘go to that rock’, leaving the execution of the command to the robot. Such a mode of operation makes the robot more responsive to its environment, and makes the control easier.

Ideally, however, we wish to have fully autonomous robots, robots that can accomplish tasks in unstructured environment without human operators. The entrance of robots into our daily lives has been rather slow when compared to their success on the factory floors. Only recently, we began to hear news about 'robot cars' driving autonomously [3], 'robot dogs' playing football[4], or humanoid robots sharing our daily environments[5]. Despite the availability of autonomous 'robot' vacuum cleaners[6] to be used at homes, or the 'robot dogs' for 'home entertainment', we are not still satisfied with the current state of robots in our daily lives. The robots, such as the ones envisioned in the movie 'I Robot'[7], that can talk to people, do shopping for its owner, and take care of its life, have not come to the stage yet.

1.1. Autonomous robotics

Autonomous robots are robots that can sense their environment, make plans to accomplish their tasks, and execute these plans through their actuators without human intervention in unstructured environments. Therefore, most of the challenge in autonomous robotics lies in its control system, making it an excellent test-bed for artificial intelligence studies.

The studies on the development of control systems for autonomous robots evolved through three main paradigms: hierarchical, reactive and hybrid. One of the first examples of hierarchical control systems, was built on the mobile robot Shakey, during late 1960's [8]. Shakey was able to sense its environment with a camera, a range finder and bump sensors. It would create a world model using the sensory information, make plans to accomplish human-set goals within this world model and execute them. The world was represented in predicate logic and accomplishing a goal was equivalent to solving a theorem. Despite its accomplishments, Shakey was too slow and susceptible to noise in its sensors. The reason was not only the hardware limitations, but also the artificial intelligence approach of the time. Deliberative reasoning and planning lied at the core of the system, and the perception and action modules only translated the world into a symbolic representation space. Researchers believed that the decision making process of an intelligent agent should consist of sensing the environment, constructing a world model, and then planning for the next action. The type of processing isolated the robot from its environment during the planning step. As a result, the robot was able to operate only in very structured environments and was not very responsive to changes in the environment.

During the 80's, as a reaction to the hierarchical paradigm, the reactive paradigm was proposed[9] to make the robots be able respond to the demands of the environment that is changing and unstructured. The idea came from nature. The challenge of operating in unstructured environments has already been overcome by living organisms. Braitenberg's vehicles[10] demonstrated how simply wired sensor-motor connections result in complex-appearing behaviors such as fear and love. Ronald Arkin introduced 'behavior-based' robots[11], which are controlled by behaviors that provide sensor-actuator mappings. The common belief of these researchers was that the robots do not need to make deliberative planning to survive in complex environments. Instead, interaction between the environment and the robot is emphasized. Their robots were successful in responding to the changes in the environment such as an obstacle coming to the robot's way, and they were fast unlike their predecessors. However, they were unable to accomplish high-level goals such as localization, map building and object manipulation.

In order to survive in unstructured and dynamic environments, an autonomous robot must be both responsive to its environment and capable of accomplishing high level goals at the same time. As a result, the hybrid paradigm, combining the advantages of both reactive and hierarchical systems, emerged. In this paradigm, simple behaviors are implemented reactively and the higher level goals are accomplished deliberatively[12]. These hybrid systems generally consist of several layers. The arrangement of these layers differ form architecture to architecture. For example, in ATLANTIS architecture, there are three layers: controller, sequencer and deliberator[13]. Controller is the reactive layer that controls simple behaviors, sequencer is the layer that organizes these simple behaviors and deliberator is responsible for high-level planning. These hybrid systems proved to be better than reactive and hierarchical systems as anticipated[14].

Although the paradigms presented above provided an architectural framework for the design and implementation of control algorithms of autonomous robots, the problem of how to design controllers for a particular task is still left to the designer. This problem still poses a big challenge, since the designer often possesses the problem from his viewpoint and is likely to propose solutions that are very specific to a domain, and are not very suitable for the robot. For example, in order to deal with the variety of entities in the environment, considerable effort[15] has been made to classify objects in the environment, according to their form or functionality. However, the object classes had to be preimplemented in the robot, restricting the types of objects the robot can deal with.

In order to build systems that are not restricted to a specific domain, the ability of learning seems essential. Humans go through a rather long developmental phase before learning to successfully act in the environment. Babies learn to control their body parts through years of experimentation. In time, they not only discover their own abilities, such as reaching objects within their arm-distance proximity, but they also learn about the world, such hollow object can be used to hold water. These abilities, which mostly fall under Cognitive Science, are also of interest to autonomous robotics. Recently, some of the concepts and theories developed within Cognitive Science has been picked up by robotics researchers, creating a new sub-field that can be called Cognitive Robotics.

1.2. Cognitive Robotics

Cognitive Robotics lies at the intersection of cognitive science and autonomous robotics. From the cognitive scientist's view, the robots can act as a test-bed to implement and evaluate ideas from cognitive science on physically embodied and situated agents. This allows constructive testing of the concepts and theories proposed. From the autonomous roboticist's view, learning architectures, as good as humans, can be designed with contributions from cognitive science. Below, we will review some of the cognitive robotics studies in the literature.

In [16], Morales trains a robot torso with various objects to learn how to grasp them. A set of features is extracted from the camera data during interaction with the objects. Possible grasping configurations are computed and one configuration is chosen and tested on the object. The result of the grasping action is observed and recorded. For the new objects, the suitable grasp configurations are selected by using voting k-nearest neighborhood algorithm.

In [17] Broxvall et. al discuss the problem of associating the sensor data with the symbols representing the objects, namely, perceptual anchoring problem and suggest a symbolic planner to overcome perceptual ambiguities like noise in sensors. The symbolic planner evaluates the matches between symbols and percepts from sensor data according to object properties and properties required for the symbol. If some of the object properties are missing, the planner initiates a recovery plan where the robot tries to obtain these properties by a more detailed examination of the object. After the information is obtained, the planner decides the best matching object for a particular symbol.

In another study, Giacomo et. al incorporates Propositional Dynamic Logics(PDLs) and Description Logics(DLs) for utilizing information from the environment in reasoning[18]. They use PDLs for reasoning about actions and DLs for knowledge representation. Both of these studies show the effort on combining the powerful decision making benefit of deliberative planning with the useful information in the environment.

Some applications in cognitive robotics take a cognitive theory as the starting point, and try to implement principles of this theory on a robot. In a particular study by Buchsbaum et. al, Simulation Theory has been exploited for building "socially intelligent" robots[19]. According to the theory, a person can predict another person's intentions by thinking himself in the same current condition with the other person[20]. The agent in the study of Buchsbaum et. al observes the actions of another agent in order to decide his next own action accordingly and to learn new actions. First, it perceives the configuration of the other agent's body parts. Then, in his own action repertoire, it searches for the action that is initiated by the observed configuration. Thus, it understands which action the other agent is starting to do. With this information, it is able to make an action that is in accordance with the other agent's action.

Theory of affordances is one of these theories and utilized by many robotics researchers in recent years[21]. The primary significance of this theory for robotics among others is that it provides a basis for a complete cognitive system. It is not specific to a particular domain or task. It provides clues for perception, representation, reasoning and action. It offers a system where perception and action together constitute representation, and reasoning is assisted by the environment.

The work done in this thesis is conducted within the MACS project, funded by the European Commission within the Cognitive Systems call of Information Society Technologies programme of FP6. The project aims to take the affordance concept (described in detail in the next chapter) from cognitive science and develop a robot control architecture utilizing this concept. In short, an affordance is a property of an agent-object system that is defined by both agent properties and object properties and gives information to the agent about the possible actions that can be performed with this object if perceived. This thesis presents a study on the perception and learning of affordances on robots.

There are two objectives of this thesis. The first objective is to gain perceptual speed-up by perceiving affordances instead of separately recognizing objects and deciding which actions can be performed with these objects. The affordance concept provides the interface between the perceived properties of the objects and the actions that can be performed with this object. We will explain these useful properties of the affordance concept in the next chapter. The second objective is to increase the accuracy of prediction of action possibilities, i.e. affordances, by developing a mechanism that is robust in the presence of noise and with a short period of training.

The scope of this thesis is to clarify affordance perception issue by proposing a method to identify affordances in the environment. This method brings a solution to the problem of learning sensory invariants that are necessary to support a desired affordance. We explained the method on a specific affordance, reachability, and performed experiments using a simulator and presented the results of these experiments.

The rest of the thesis is organized as follows. In Chapter 2, the affordance concept is described and literature survey on the use of affordance theory in robotics is presented. The problem of affordance perception is stated and our approach to this problem is identified. In Chapter 3, the proposed affordance perception mechanism is introduced. Chapter 4 gives the framework for the experiments. The simulation environment and specifications of the simulated robot are presented. The object reachability problem is introduced and the nature of the initial experimental data is discussed. The results of the experiments and discussions on findings are stated in Chapter 5 and finally conclusions are presented in Chapter 6.

CHAPTER 2

AFFORDANCES

The theory of affordances was introduced by Gibson[22], the founder of ecological psychology discipline. Gibson has denied the classical view on perception, where physical objects create stimuli to activate particular object representations. According to his theory, humans perceive affordances in the environment. Perception is direct, explicit object recognition is not required. This provides competitive advantage by enabling fast response to the environment, and increases chances to survive.

"…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." [22]

In this chapter, we will explain the theory of affordances before implementing it in an autonomous robot. Following the theory, literature survey on the applications of affordance in robotics will be presented.

2.1. Theory of Affordances

An affordance is what an entity in the environment offers an agent. For example, a stone affords throwing. A doorway affords entry and exit. An obstacle affords collision and possible injury. It can be seen from the examples that an affordance can be beneficial or harmful to the agent. Thus, perceiving affordances provide many advantages to the agent. The agent can see what it should utilize and what it should avoid beforehand. By this way, it can obtain a useful entity before another agent takes it, or it can avoid a harmful entity before getting injury.

Affordances are determined not only by properties of the entities but also by the agent's properties. An entity may not offer the same affordance for different agents. For example, a chair affords sitting for an adult human as in Figure 1. However, it does not afford sitting for an elephant, an infant or a robot. For these agents, a chair may afford blockage or even an injury. The difference is due to the physical properties of the agents. The agent should have legs and leg height should be within a specific range for the chair to afford sitting. In addition, the agent should have a waist. A wheeled mobile robot has neither legs nor a waist, thus a chair would not afford sitting for it. An infant has legs, but he is not capable of using them yet. A chair would not afford sitting for him until he walks on foot and his leg reaches to a certain height. However, a smaller chair may afford sitting for a child although he has shorter legs. For the blockage affordance, the height and the area covered by the chair are important from the entity properties side. From the agent's part, body height matters. The agent can be short enough to pass under the chair. Otherwise, the chair would block the passage.

Perceiving affordances enables the agent to see the possible actions it can perform in the environment. It is up to the agent to perform these actions or not but these possibilities can be utilized as the building blocks of any plan to reach some goal. For example, an agent might have a goal of passing over a gap. From its previous experiences, it might have learnt that rigid, flat surfaces afford support. Thus, the agent searches for rigid and flat surfaces in the environment. A plank can be an ideal candidate for affording support. Using the plank, the agent can cross over the gap. The ability to "see" affordances of entities enables the agent to find quick solutions to emergent problems in the environment, without

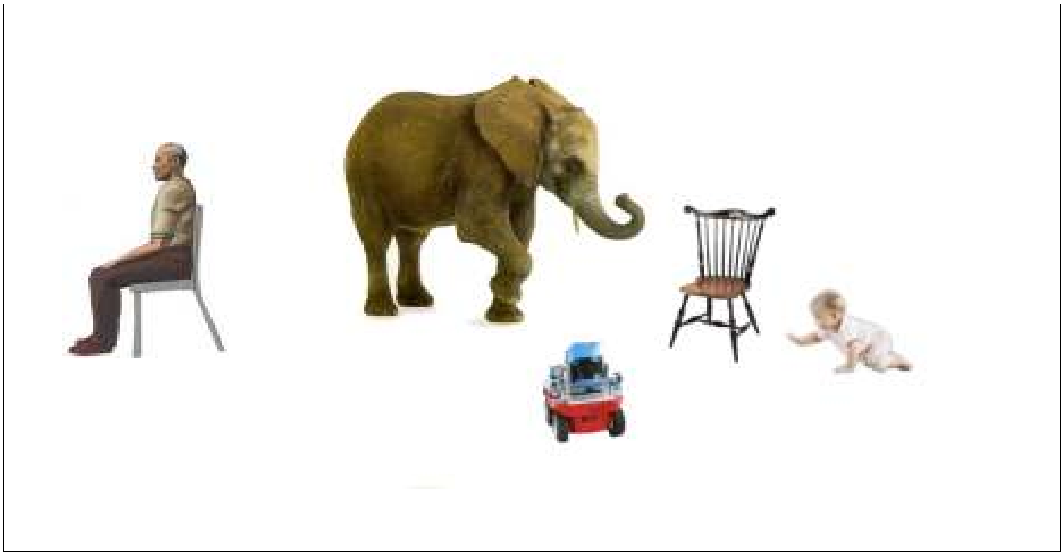

Figure 1 An affordance is relative to the capabilities of the agent.

elaborate decision making processes.

A single object may have many affordances. For example, a mug affords holding liquid substances inside and also affords throwing as a weapon. These affordances do exist at the same time for a single object. An agent may perceive the former affordance if it needs to keep water. The latter can be perceived if the agent needs to protect itself from an enemy. Thus, perceived affordances are context-dependent, although possibilities for action with the object persist.

According to Gibson, affordances are specified by the invariants in the optic array. Here, optic array implies the light received by the eyes. The visual system develops in time to be able to identify these invariants[23]. This development is the result of the interaction between the infant and the environment.

This study focuses on perception of affordances for robots. The robot will encounter the challenges that natural agents have encountered such as enormous flux of sensor data. From this flux, it will select what it needs, namely, affordances.

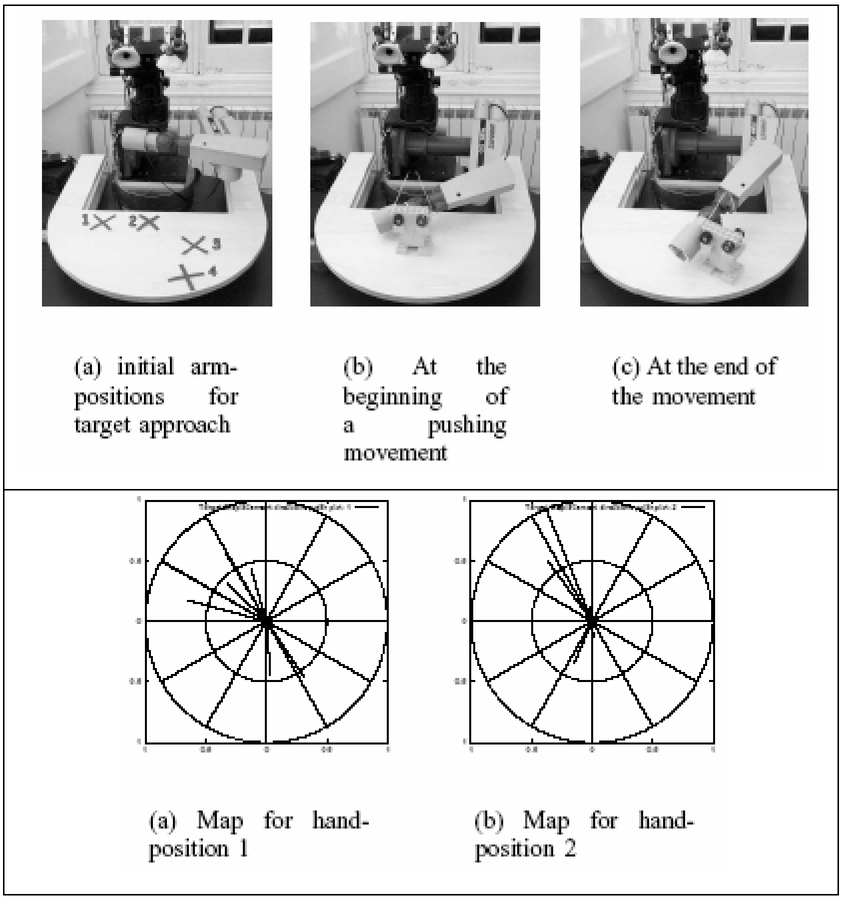

Figure 2 The experiments in studies of Fitzpatrick and Metta et. al. The robot learns direction maps for different hand positions by observing initial hand positions and direction of object movement[24].

2.2. Related Work

In this section, the use affordances in robotics are reviewed.

Fitzpatrick and Metta[24]. et. al define an affordance as "a visual characteristic of an object which can elicit an action without necessarily involving an object recognition stage". From this definition, it is possible to infer that by perceiving affordances, a robot can decide which action to perform on an object without identifying its class label. For perception of affordances, they suggest a

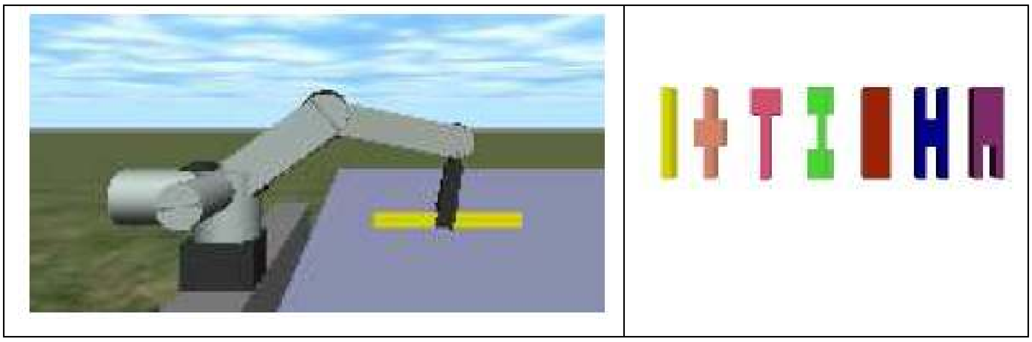

Figure 3 A snapshot from the experiments conducted on simulation by Stoytchev. The robot learns tools affordances by applying a set of exploratory behaviors and observing the results.

learning mechanism and conduct experiments on the object displacement task. The robot pushes various objects to discover the mapping between initial hand position and direction of object movement as depicted in Figure 2. After a set of experiments, the robot obtains a direction map. Through this direction map, the robot knows how to orient his arm to move an object to a desired location. Then, experimenting with objects of different shapes, the robot learns which action to perform on a particular shape. For example, a ball rolls, a cube drags. Thus, the robot can perceive displacement affordances of objects as well as the appropriate hand positions to realize these affordances.

A more recent study has been published by Stoytchev[25]. The aim of his study is to develop and evaluate a way of representing and learning affordances in autonomous robots. The primary motivation is that humans learn affordances by interacting with them. The most influent ideas in this study are from Gibson's affordances, Arkin's behavior-based approach and experimental results from animal studies on tool using task. Whenever a new object is encountered, the robot applies a set of exploratory behaviors and observes the results. The objects and the simulation environment are shown in Figure 3. In other terms, the representation of the tool is 'grounded' in the behavioral and perceptual capabilities of the robot[26]. This idea is compatible with theory of affordances where animals can utilize affordances of entities if their physical capabilities are compatible with the properties of these entities. The method is implemented both on simulation and on a real robot, and successful results are obtained in predicting the affordances of different tools.

Wünstel and Moratz use the affordance concept in object recognition[27]. Although they identify their approach as affordance-based, they only use affordances as class labels. Their approach is successful in identifying objects in unstructured complex environments. However, their method required an initial class description for the objects, which is not feasible for autonomous robots. Additionally, their approach is not compatible with Gibson's affordance concept, where the affordances are learnt by direct interaction with the objects, and learning affordances is not classifying objects.

Lewis argues that robot locomotion is not solely a control problem; rather it should be handled as a perceptual process[28]. He treats affordance as an example of a 'percept' and suggests a way of integrating percepts in the environment with motor actions. In his system, percepts in the optic flow are extracted, and the association between these percepts and the joint position information is learnt by a neural network. So the robot can predict what it will 'perceive' after generation of a self-movement. If the prediction fails, the system is fine-tuned to adapt to the new percepts. Although the system is not totally affordance-based, the direct link between perception and action and interaction with the environment makes it consistent with the theory of affordances. The future work offered by Lewis includes determining the appropriate set of percepts necessary to guide locomotion. This is similar to determining invariants for an affordance. His ideas are worth considering for an affordance-based system and his work can be extended by ideas from the theory of affordances.

In a previous work of Lewis, terrain affordances were learnt by a bipedal autonomous robot[29]. He points out to the benefits of biologically inspired robot design. Affordances are one aspect of his studies. His main emphasis is on predicting consequences of an action from visual information. He uses a neural method that involves a pattern associator which maps the novel sensory data to length of foot steps and is trained whenever a collision occurs. The novel data is determined by finding the difference between expected data and the current raw data. By this way, the robot can detect collisions before direct contact with the obstacles i.e. can perceive affordance of the terrain features.

Slocum et. al. aims designing agents which can visually judge pass ability of openings relative to their own body size, discriminate between visible parts of themselves and other objects in their environment, predict and remember the future location of objects in order to catch them blind, and switch their attention between multiple distal objects[30]. The robot is controlled by a continuous-time recurrent neural network, whose parameters are evolved by a genetic algorithm.

The studies on affordance perception in robotics show that the current research is directed towards building cognitive systems that can deal with the entities in the environment in a more flexible and efficient way than before. However, the solutions are proposed for specific tasks. The robot gains the ability to learn that task, and other solutions need to be developed for other tasks. In order to be an autonomous agent, a robot needs to have 'core' abilities that enable it to learn different tasks. We propose a method for affordance perception, which we believe is core ability for an autonomous robot. The next chapter presents our method in detail.

CHAPTER 3

AFFORDANCE PERCEPTION

Autonomous robots face an incoming sensory stream flow from a changing environment and they need to process these to act in real-time. Illumination changes from place to place as well as at different times of the day creating different images of even the same objects. The view of an object changes significantly as the robot moves around. As the robot moves around, the robot is likely to face with novel objects that it has not encountered in the past. It is not a exaggeration to say that, the robot never sees the same scene twice. Despite these, the robot still has to operate in its environment in real-time. Hand-made controllers are inadequate under these conditions, and robots are required to adapt to their operating environment based on their experiences with it.

Adaptation methods that rely on object recognition fail to provide a good generalization. Learning functionality of objects from the scene usually incorporate a predefined association between prototype forms and their functionalities and match the form of observed object with one of the prototypes. The problem with these methods is to find an appropriate representation for the objects. The representation should not be affected significantly by the change in illumination, point of observation and occlusions so that the new observed object can be matched to a prototype in the knowledge base properly. Besides, the representation should support multiple functionalities since one object can have much functionality. A crucial problem is the number of prototypes in the knowledge base. No matter how many prototypes are stored, they will not be enough to represent all the objects in the agent's environment. As a result, a robot which learned to roll an orange ball, after experimenting with it, is unlikely to generalize its knowledge to a red apple.

The affordance concept has important implications for autonomous robot that are expected to operate in unstructured, everyday environments. An affordance-based view of robot control indicates that the robot does not need to recognize an object in order to know what it can be used for, a.k.a. what it affords.

In this thesis, we propose that perception of affordances require both the learning of relevant features and the characteristics of these relevant features. Affordances in the environment are perceived by detecting the relevant information from the stimuli obtained through the sensors. This relevant information is learnt by the robot through its interactions with the environment. During these interactions, the robot acts on the environment and observes the results of its actions. At the same time, the robot has a perception from the environment. The stimulus obtained can be of any type: pixel values from camera, distance range from infrared or sonar sensors etc.

The question that we tackle in this thesis is: How can a robot learn the sensory invariants that are necessary to support a desired affordance? For instance, what are the sensory invariants that the robot must check for traversability? Although it may seem trivial, to solve this problem, the robot, through its experiences, must learn that the roughness of the ground, but not its color, is the relevant sensory stimulus that needs to be checked.

We propose a method for robots to perceive affordances of objects. The method is novel in terms of representation of the objects since the representation tells what the object affords. The robot does not need to make explicit processing in order to decide what the object is used for, it will 'see' the affordance when the object is in its field of view. The time for responding to the environment is reduced since there are no separate object recognition and action decision processes. This is important for an autonomous mobile robot since the environment is dynamic and any long decision making process can decrease its chance of survival.

We present the affordance perception method based on invariant extraction from the visual stimulus. The details of the method are explained in the following sections.

3.1. An Affordance Perception Method

The affordance perception method proposed in this study consists of three phases. In the first phase, the robot interacts with objects and tests them to see if they support a desired affordance or not. This phase is similar to the 'physical babbling' period of babies, where the baby explores his abilities. In the second phase, the robot, based on the experimental data obtained from the first phase, extracts (1) the relevant features, and (2) the characteristics of the featural invariants for that affordance. In the third phase, when the robot is presented with novel objects, the robot predicts whether it supports the specific affordance or not.

As the robot interacts with the objects, there are some structures in the stimulus that persist for a specific outcome of an interaction. For example, whenever a robot hits an object that affords blockage, namely an obstacle, the object is of a specific height and width, as in Figure 4. These persisting structures are called invariants, and these invariants specify the affordances. Note that an affordance is relative to the observer, thus the invariants specifying an affordance will differ from one robot to another.

In the proposed method, the robot obtains a number of features from the continuous flux of data from its sensors. These features can be of any complexity, from plain pixel values of regions to the shape of the object. The choice of the features to be calculated depends on the task to be performed, or the

Figure 4 There are relevant features among the other features in the stimulus that support certain affordances. In this figure, the objects afford blockage to the robot. The robot can obtain many features from the stimulus it receives, such as color, distance, area, shape, and etc. But there is an invariant feature among these features that gives information about blocking affordance. It is the height of the objects; no matter what the color and shape of the object is, an object of a specific height will afford blocking to the robot.

environment that the robot operates. Actually, the type of features to be calculated from the flux can be adaptive. The complexity of features can be increased in time, as the affordance perception of the robot gets more precise.

The robot records the values for each feature during the interaction with an object. The interaction will have an affect on the robot or it will satisfy a goal of the robot, i.e. the object will afford something to the robot. The robot will record this affordance together with the feature values. After several interactions with the objects in the environment, a set of feature values are obtained for an affordance. Some of these values belong to stimulus obtained from objects that have this affordance, and some values obtained from objects that does not have this affordance. For each feature, the values are normalized and sorted from the smallest to the largest (Figure 5).

Feature^1	Feature^2	.	.	.	Feature^n
v_1^{max}	v_2^{max}	.	.	.	v_n^{max}
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
v_1^{s}	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.		.	.	.	.
v_1^{min}	v_2^{min}	.	.	.	v_n^{min}

Figure 5 Feature values are calculated from the visual stimulus obtained from the real world entities. The values are normalized and sorted after the robot experiments with a number of entities. v_n^{min} represents the minimum feature value for the nth feature. v_n^{max} represents the maximum feature value for the nth feature.

The consequent values that have the same affordance are grouped together so that each feature has a set of groups of values that either belongs to an affording object or not (Figure 6). We will call each of these groups a strip. For example, considering the blockage affordance, the entity_height feature may consist of two strips such as one strip from 0-5 cm belonging to objects that do not afford blockage, and one strip from 5-50 cm belonging to objects that afford blockage.

The number of values that a strip has indicates the relevance of the strip to the affordance. If a strip has large number of points compared to other strips, we say that this strip is a relevant strip. Consequently, a feature that has a relevant strip is called a relevant feature.

For the blockage affordance example, consider the *entity_color* feature. If we sort the color values from the smallest to the largest, and form the strips according to their affordance, the number of values that falls into any strip will not be large, since there is no significant relationship between the object color and the blocking affordance. On the other hand, for the *entity_height* feature, a large number of values will gather around 5-50 cm strip that belongs to objects having the blockage affordance. Here, the 5-50 cm strip is a relevant strip, and the object_*height* feature is a relevant feature.

These relevant features are the information for affordances in the environment. Instead of recognizing the abstract objects or trying to match the novel objects to known prototypes, the robot simply identifies the relevant features. Thus, what it sees in the environment are the affordances.

The relevant strips that include values from stimulus obtained from objects that have the affordance A are referred to as positively relevant strips. Oppositely, the relevant strips that include values from stimulus obtained form objects that do not have the affordance A are called negatively relevant strips. A feature can be both positively and negatively relevant to an affordance A if it includes both the positively and negatively relevant strips.

The benefit of considering both the positively and negatively relevant features is to have judgments from two different sources and thus to have a more confident response. For example, an object of 30 cm height and 2 cm width may belong to a positively relevant strip of entity_height feature, and at the same time to a negatively relevant strip of entity_width feature. Through this conflicting information, the robot understands that there is something about this object that it has not discovered yet. It needs to obtain values from this object and update the existing strips accordingly.

A drawback of this method is that it assumes slightly accurate feature values. For example, if both of the cameras are shaking, the value for *object center y* feature might fall into a negatively relevant strip in one trial and into a positively relevant strip in another trial, for the same placement. In this case,

$Feature^1$	$Feature^2$	.	.	.	$Feature^n$
$a(v_1^{max})$	$a(v_2^{max})$	.	.	.	$a(v_n^{max})$
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
$a(v_1^{s})$	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.		.	.	.	.
$a(v_1^{min})$	$a(v_2^{min})$	.	.	.	$a(v_n^{min})$

$$a(v_n^x)= \begin{cases} 0 & \text{if } v_n^x \text{ is a value calculated from visual stimulus obtained from an entity that does not have affordance A} \\ 1 & \text{if } v_n^x \text{ is a value calculated from visual stimulus obtained from an entity that has affordance A} \end{cases}$$

$Feature^1$	$Feature^2$	.	.	.	$Feature^n$
1	0				1
0	0	.	.	.	1
1	0	.	.	.	1
1	0	.	.	.	0
0	1	.	.	.	0
0	1	.	.	.	1
1	1	.	.	.	1
0	0	.	.	.	0
0	0	.	.	.	1
1	1	.	.	.	0
1	1	.	.	.	1

Figure 6 Affordance values for each feature value.

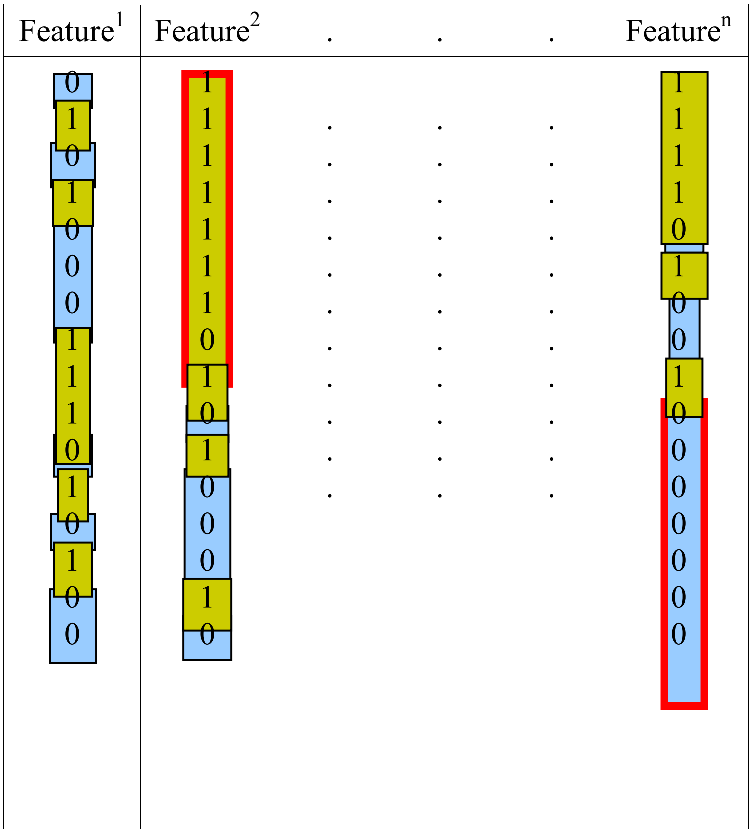

Figure 7 An example of feature strip formation. The large strip in Feature^2 is a candidate for being a positively relevant strip and thus Feature^2 is a candidate for a positively relevant feature. Similarly, the large strip in Feature^n is a candidate for being a negatively relevant strip and thus Feature^n is a candidate for a negatively relevant feature.

many small strips can be formed with positive and negative values, and thus object center y feature would not be considered as a relevant strip. This problem can be overcome by taking the camera positions into account while recording the feature values. In this study, we assume that camera, crane arm and robot body is stable and slightly accurate feature values can be obtained through sensors.

CHAPTER 4

EXPERIMENTAL FRAMEWORK

In this chapter, we describe our experimental framework and present the problem that we chose to study. First, we introduce the MACSIM simulator which simulates the KURT3D mobile robot platform (Figure 8). Then, we motivate and present the affordance of object reachability as our case study. Finally, we present an analysis of the sensor data which will provide evidence about its informative capability for the affordance under discussion.

Our robot experiments with various objects in order to learn perceiving the reachability of objects. Details of the experiments and the simulation environment will be explained in the following sections.

4.1. Experimental Framework

We conducted our experiments using the MACSIM simulator, developed at KOVAN Research Laboratory for the MACS Project[1].

---

[1] Multi-sensory Autonomous Cognitive Systems interacting with dynamic environments for perceiving and using Affordances (MACS) - Specifically Targeted Research Project (STReP), IST-2003-2.3.2.4 Cognitive Systems, Project Number: FP6-004381, http://www.macs-eu.org

Figure 8 KURT3D robot platform. KURT3D is a 6 wheel, differential drive mobile robot. It is equipped with a 3D laser scanner (the large cube at the front of the robot), two pan-tilt cameras (small spheres), and eight infrared proximity sensors around the robot body. A crane, which will allow the robot to handle objects, is already designed and will be installed. The robot is controlled by a piggy-back laptop.

4.1.1. MACSIM – The simulator

MACSIM is a physics-based simulation of the KURT3D mobile robot platform A snapshot is shown in Figure 9. MACSIM provides the following facilities for our study:

- Creating objects of different size, shape, position and orientation
- Obtaining online data stream from the sensors
- A set of behaviours such as reach, lift and drop

MACSIM is built using Open Dynamic Engine (ODE), a library for simulating articulated rigid body dynamics[31]. The robot model included a realistic model of actuators and sensors on the physical robot, and the physical simulation library handles the physical interactions between the objects in the world. As it is, MACSIM provided nice test-bed for testing an affordance based perceptual system.

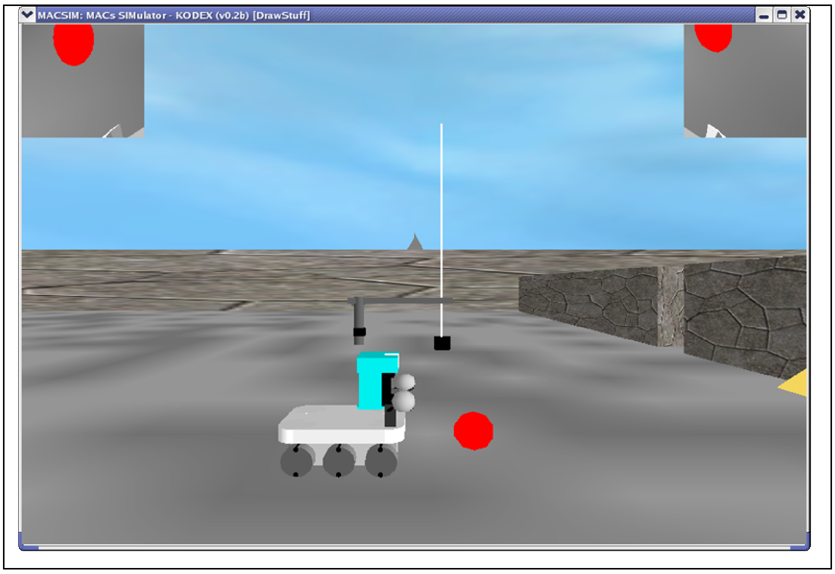

Figure 9 A snapshot from the MACSIM simulator. The small windows on the upper left and right corners display the left and right camera views respectively. Note that, the simulated robot model also included the crane which was not shown on the picture of the KURT3D robot.

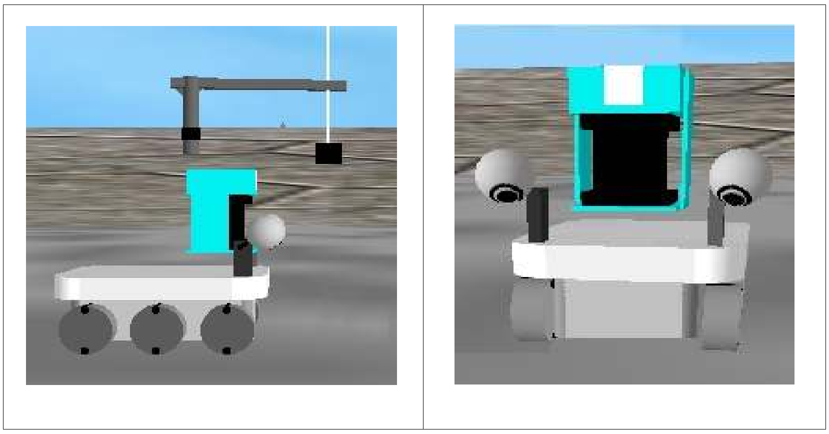

Figure 10 The robot

(a) Side view - the crane arm and the magnet attached to the arm via the rope (b) Front view - laser scanner and cameras

4.1.2. Simulated Robot Model

Since an affordance is determined by both the environmental entity and the agent, it is crucial to state what capabilities and constraints are offered by the robot.

Sensor configuration

The robot in the MACSIM simulator has two pan-tilt color cameras and a 3D-Laser scanner with 180° horizontal field of view. (see Figure 10)

Manipulators

The crane robot arm has 3 degrees of freedom. A rope with a magnet at one end is attached to the arm. The magnet can be lowered and raised by pulling and leaving the rope with a motor. Only magnetizable objects can be manipulated.

4.2. Reachability affordance

We have chosen 'reachability' as the affordance to study the learning of affordances, since reaching is the first stage of interacting with an object. It is a fundamental behavior that initiates exploration. Other behaviors such as grasping and lifting require reaching the object first. To realize this behavior, detecting reachability of objects is essential. It does not only affect decision about whether to reach the object or not, but also determining the appropriate body and arm position and configuration as well as the reaching trajectory[32].

Reaching behavior has gained special emphasis in infant development literature. Many studies have been conducted to see whether infants realize reachability of objects before trying to reach them[33, 34]. Researchers have also investigated reach behavior from the Gibson's point of view[35, 36].

Figure 11 Reach behavior in infants has been an attractive phenomenon for human development research.

A recent work of Rochat analyses the reaching behavior in infants to show that infants can detect whether an object is reachable or not before touching it[37]. He refers to Gibson's affordances and argues that reachability of an object is determined by both object properties and infant's physical capabilities.

Finally, reachability is of interest to the KURT3D robot, since it has a crane arm with a magnet attached via a rope to it.

4.3. Implementation

4.3.1. Scenario

For the case of reachability, the robot is kept at a fixed point. It can only move its arm in the horizontal plane. Objects are placed randomly within a specified area. One object is presented at a time (see Figure 12). For each object, the robot executes the reach behavior. If the magnet engages with the object, the

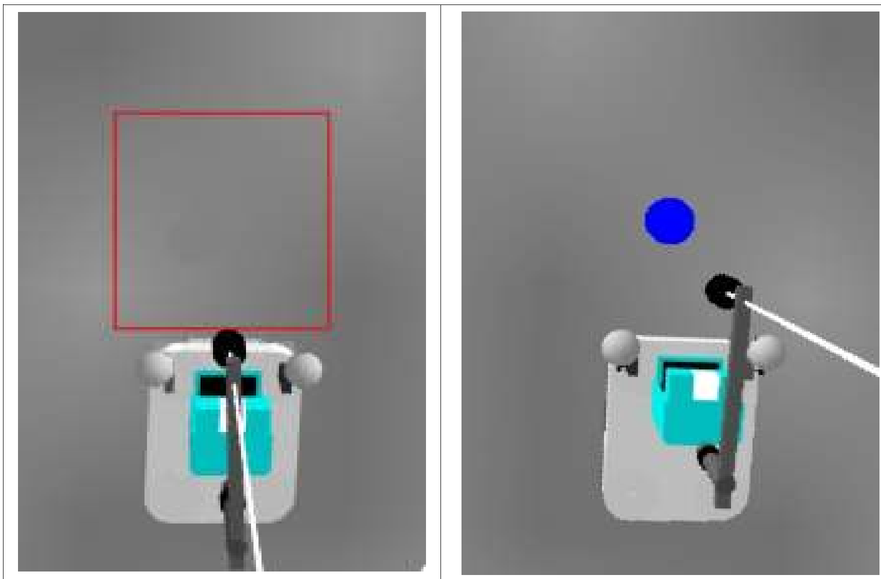

Figure 12 (a) The range of random object placement (b) One object is presented at a time

object is said to be reachable. The robot receives the sensor stream continuously throughout the simulation run from the two cameras.

4.3.2. Initial experimentation

We will explain implementation of our method within the simulator environment under the capabilities and constraints described in the previous sections. The robot is presented with a collection of objects randomly placed in front of it, one object at a time. Range of object placements is shown in Figure 13. Firstly, the robot extracts a feature vector representation of the object using a number of feature detectors listed below. Then, the robot executes the reach behavior on the object. Finally, the robot checks whether the reach behavior succeeded or not.

Table 1 Features used in the experiments.

feature	*explanation*	*values*	*category*
Object color red	Red value of the object pixels	0, 255	Camera
Object color green	Green value of the object pixels	0, 255	Camera
Object color blue	Blue value of the object pixels	0, 255	Camera
Crane color red	Red value of the crane pixels	0, 255	Camera
Crane color green	Green value of the crane pixels	0, 255	Camera
Crane color blue	Blue value of the crane pixels	0, 255	Camera
Object area right	Number of pixels occupied by the object in the right camera	0 - 640*640	Camera
Object area left	Number of pixels occupied by the object in the left camera	0 - 640*640	Camera
Object center x right	x pixel-coordinate of object center on right camera	0 - 640	Camera
Object center y right	y pixel-coordinate of object center on right camera	0 - 640	Camera
Object center x left	x pixel-coordinate of object center on left camera	0 - 640	Camera
Object center y left	y pixel-coordinate of object center on left camera	0 - 640	Camera
Crane area right	Number of pixels occupied by the crane on the right camera	0 - 640*640	Crane
Crane area left	Number of pixels occupied by the crane on the left camera	0 - 640*640	Crane
Crane center x right	x pixel-coordinate of crane center on the right camera	0 - 640	Crane
Crane center y right	y pixel-coordinate of crane center on the right camera	0 - 640	Crane
Crane center x left	x pixel-coordinate of crane center on the left camera	0 - 640	Crane
Crane center y left	x pixel-coordinate of crane center on the left camera	0 - 640	Crane
Rope tension	Tension on the crane rope	0 - 10	Crane
Rope length	Magnitude of the extension of the crane rope	0 - 4.5	Crane
Object distance	Distance between the robot body and the object	0 - 10	Range sensor

feature	*explanation*	*values*	*category*
Crane distance	Distance between the robot body and the crane	0 - 10	Crane
Object shape box	True if the object is a rectangular prism	0, 1	MACSIM
Object shape cylinder	True if the object is a cylinder	0, 1	MACSIM
Object shape sphere	True if the object is a sphere	0, 1	MACSIM

Creating feature vector representation of the object

Extracting the features from the raw sensor readings is a field of study itself. Our aim is not calculating precise values for the attributes of objects, but rather obtaining a set of structures that characterize the current sensory data. We have selected a set of features appropriate for using a two camera system. The features extracted can be roughly categorized into four:

- Features obtained from the left and right camera images. We assumed that the object is segmented from the background, and applied crude feature detectors to the segmented image. We also assumed that the crane head is tracked and features from that are also extracted.
- Feature obtained from the 3D range sensor. We used the smallest range reading from the 3D range sensor.
- Features obtained from the crane. These include values such as rope tension and rope length.
- Features obtained from MACSIM. Features such as the shape of an object are difficult to extract and fall beyond the focus of the thesis. We chose to extract the shape of the object from the data internal to the MACSIM simulator.

These features are listed in Table 1.

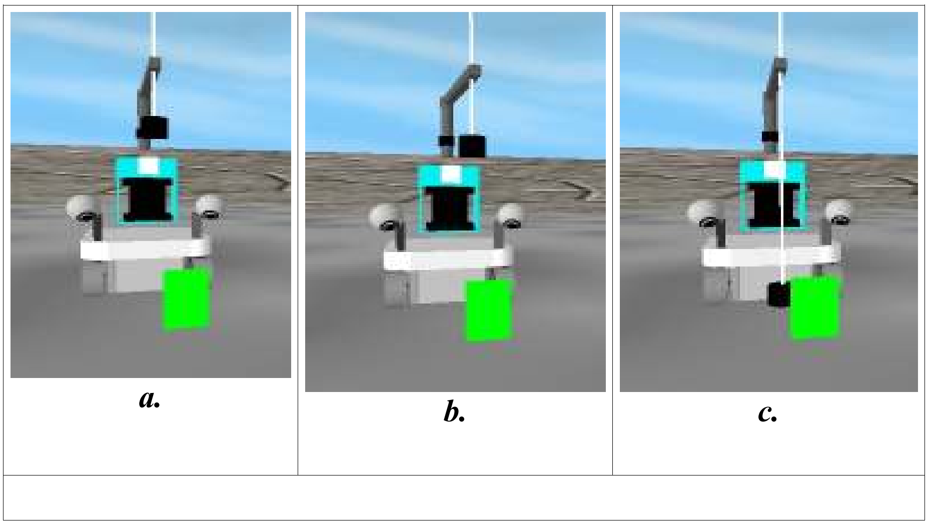

Figure 13 Reach behavior (a) Detect position of the object (b) Orient crane arm (c) Lower the magnet

Executing the reach behavior

Reach behavior is hand coded in MACSIM. It consists of the following steps. (1) Detect position of the closest object in the field of view, (2) Orient the crane arm towards the object, (3) Lower the magnet (see Figure 13)

Checking result of the behavior

The robot can decide whether the affordance is successful or not, by checking position of the magnet with respect to the object. Snapshots from successful and failure cases are shown in Figure 15.

Analysis of the feature values

In an experiment with 100 objects, we analyzed the feature values calculated from the cameras at the start of reach behavior for each object. We will compare values for the successful and failure cases.

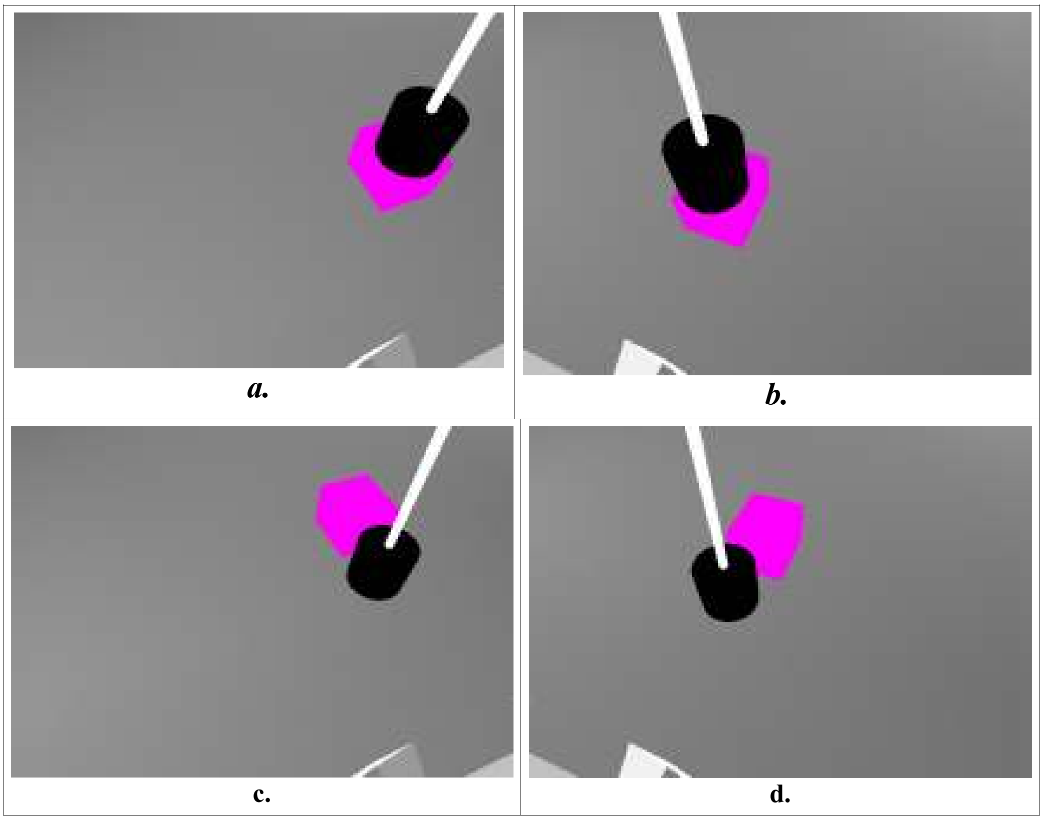

Figure 14 In (a) and (b) reach behavior is successful. (c) and (d) shows the failure case. (a) and (c) are left , (b) and (d) are right camera views.

For the object area right and object area left features, success points show a wider distribution whereas values resulting in failure accumulate in the 0-0.2 range. This shows that if the object is not reachable, it will probably have a small value for the object area right and object area left features. However, the opposite is not true. The value of object area is related with the object distance and object size. Small object area may mean that the object is small or it is far from the robot.

For the object center y right and object center y left features, success points gather between 0.4-0.7 whereas fail points accumulate between 0.7-0.9 values. So we can say that the smaller the value of the object center y, the more possible that the object is reachable. Value of these features is smaller as the object gets closer to the robot. Oppositely, it will get larger if the object goes further. Thus, these features give information about the object distance.

Object color red, object color green and object color blue features get binary values. Success and fail cases are distributed homogeneously so we can say that color features do not give information about reachability.

Values of object center x right and object center x left features are also distributed almost uniformly. Success and fail values do not produce clusters. Thus, these features do not provide information about reachability. They are related to the horizontal placement of the object and reachability is concerned with vertical placement.

For the object distance feature, the fail and success values are significantly separated. The values for the fail cases are accumulated between 0.6-0.9 and for the successful cases this range is 0.3-0.6. Thus, if the value of the distance feature for an object falls into 0.3-0.6 range, the object would probably be reachable.

As seen from the above discussion, some features are more informative about the reachability of objects than the others. The proposed mechanism makes use of this fact. In the next chapter, details of the mechanism are presented with the results of the experiments conducted on the perception of object reachability using the proposed affordance perception mechanism.

CHAPTER 5

EXPERIMENTAL RESULTS

In this chapter, we apply the proposed relevant feature extraction method to the the perception of the reachability affordance on the simulated KURT3D robot in MACSIM. Using the experimental framework described in the previous chapter, we presented the robot 200 different objects randomly placed in its front and tested the reachability affordance. We saved the features extracted from each test and the result of the test (afford/not afford) in a data file. We have created 50 different training and testing data sets, each containing 100 data points, and used them in the rest of this study. Note that, we envision our method to run in real-time on the robot and that the seemingly batch processing is only chosen to minimize the simulation costs and to maintain the replicability of our experiments.

Our experiments consisted of two steps. In the first step, using the training data, shown in Figure 15, we formed strips for each feature using the method described in Section 3.1. Here, we chose to work on each feature alone to form the strips, since, we would like to identify the relevant feature channels that are essential for our affordance. The strips are formed both for positive (afford) data points as well as negative (not afford) data points. In the second step, using the strips created, we applied four different methods to make predictions about the affordances of the objects included in the test data and reported the results.

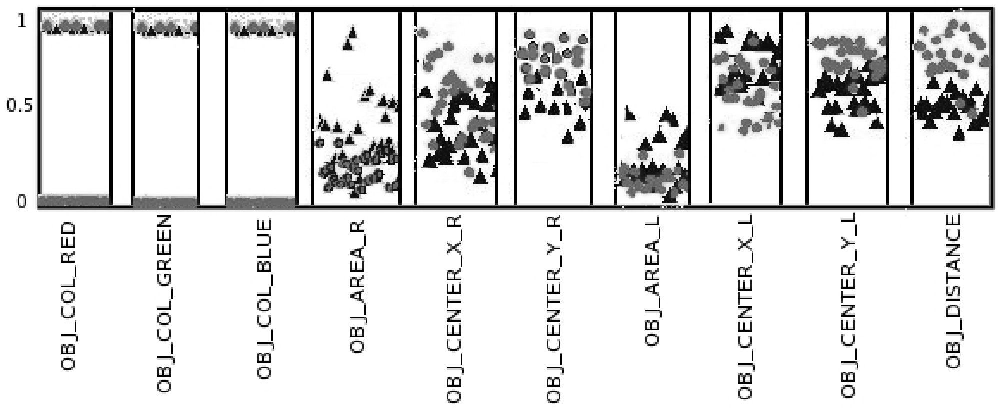

Figure 15 Scatter plot diagram of some of the feature values for the success and failure cases. Black triangles represent successful cases and grey rounds represent fail cases.

5.1. Strip formation

Strips reflect robot's experience with training objects. The first phase of the experiment consists of forming the strips, after the training is completed. Figure 16 demonstrates formed strips after a training phase with 100 objects. 25 features were calculated from the camera data. The values for color related features are gathered in 0 and 1 points, since the objects were either red, green or blue.

Features that have the largest strips are considered as relevant. In Figure 16, distance of the object from the robot, referred to as obj_dist, has the largest and most compact strip for both positively and negatively relevant strips. Thus, the distance feature is both positively and negatively relevant to the reachability affordance.

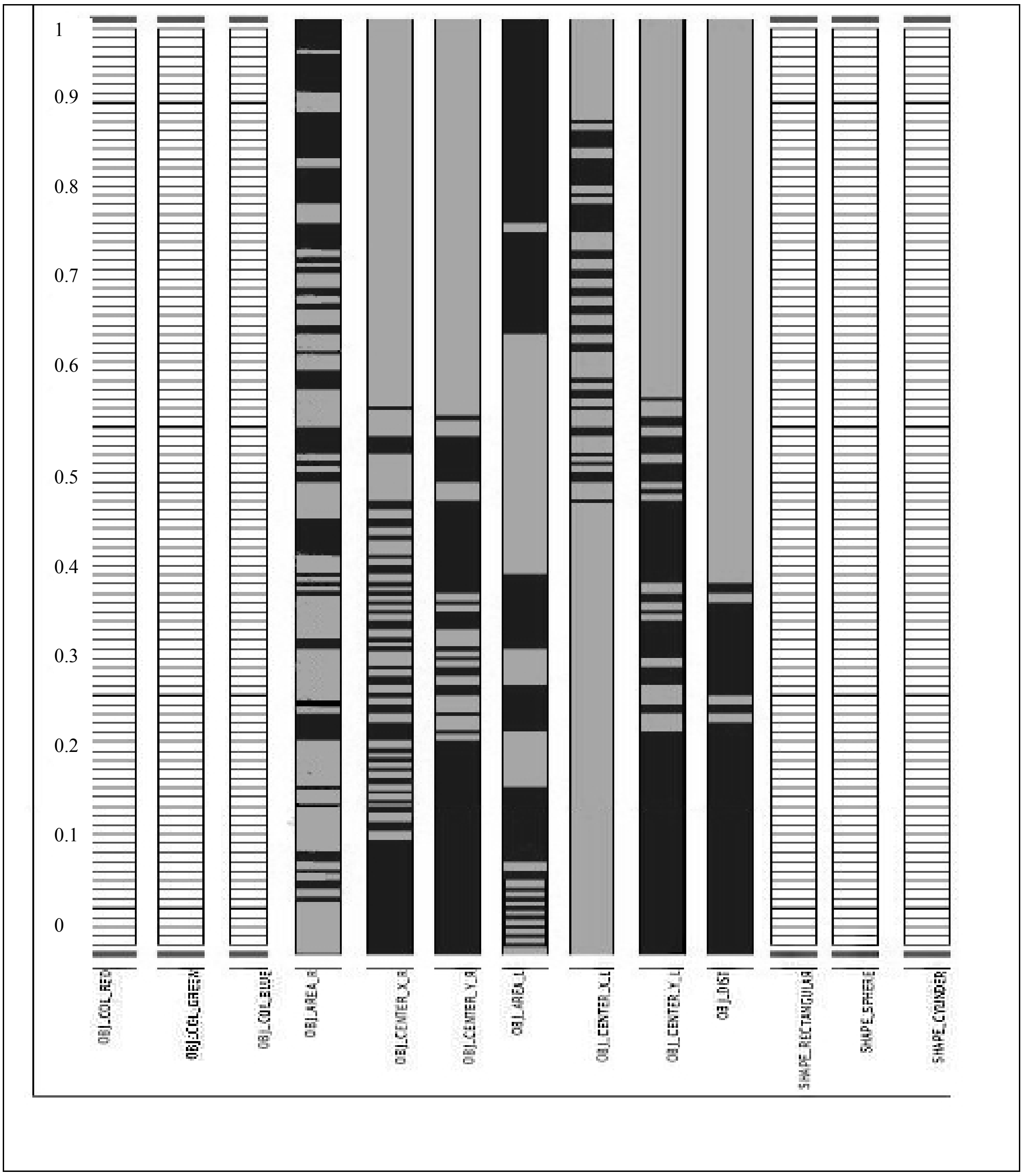

Figure 16 Strips formed after training the robot with 100 objects. Black areas represent strips that are positively relevant to the reachability affordance. Grey areas represent negatively relevant features. Corresponding feature names are stated under each column. Features related with crane color, crane area, crane distance, rope tension and rope length are not shown in this figure since their values are constant for all objects. For space reasons, features that are not very relevant are omitted from the figure.

Table 2 Decision method for the reachability affordance. For the test objects, values of the relevant features are examined.

Belongs to a positively relevant strip?	Belongs to a negatively relevant strip?	Final decision
Yes	No	Reachable
No	Yes	Not reachable
Yes	Yes	Undecided
No	No	Undecided

Object area and y-dimension of the object center in the left camera follow the distance feature, referred to as obj_area_l and obj_center_y_l. As the object gets closer to the robot, the area of the object in the camera view gets larger. For the y-dimension of the center of the object, there is a large strip containing the smaller values of that feature, meaning that the objects for which small y-dimension values were obtained are reachable.

Note that most of the large strips have many smaller neighbor strips. These smaller regions point to two problems that the robot encounters. The first one is the sensor noise, when incorrect pixel values are obtained. Second problem is the incorrect feedback received from the object. For example, the object may slip in the ground or the magnet cannot be precisely inserted onto the object. These noisy regions can be minimized as the number of experimented objects increase.

Thus, the relevant features are apparent after the formation of strips. The next step is to use these strips in perceiving reachability of new objects.

5.2. Affordance Perception

Using the strips created, we applied four different methods to detect and learn relevant features and their characteristics and evaluated their performances.

5.2.1. Using relevant features without threshold (STRIP-PASS)

This method involves taking a number of relevant features and checking whether the value of these features in the new object belongs to any of the strips in these features. If the object has the desired values for a specified number of features, the robot gives a positive response. If the value of these features in the new object is not in a positively relevant strip and is in a negatively relevant strip, the robot gives a negative response. The decisions of the positively and negatively relevant features are integrated for the final decision about the affordance of the object. This method is summarized in Table 2.

We have analyzed the effect of number of relevant features on the accuracy of predictions. The mean, maximum and minimum values are calculated for 100 trials each having 100 objects. The robot was 85% successful in its positive responses and 87% successful in its negative responses when the most relevant feature is used in the decisions(see Figure 17). The accuracy of the predictions increases as the number of relevant features increases. For 4 relevant features, the ratios rise to 92% and 96% for the true positive and true negative ratios respectively.

However, the number of responses decreases significantly as the number of relevant features increase. Positive responses decrease from 47 to 14 and negative responses decrease from 51 to 19 as seen in Figure 19.

5.2.2. STRIP-PASS with threshold

In this set of experiments, the robot did not check all the strips for the relevant features; only the strips whose relevancy exceeds a certain threshold were considered. The aim of these experiments is to see whether the robot can discard a large number of small strips so that it can detect the affordance of the objects in a shorter time.

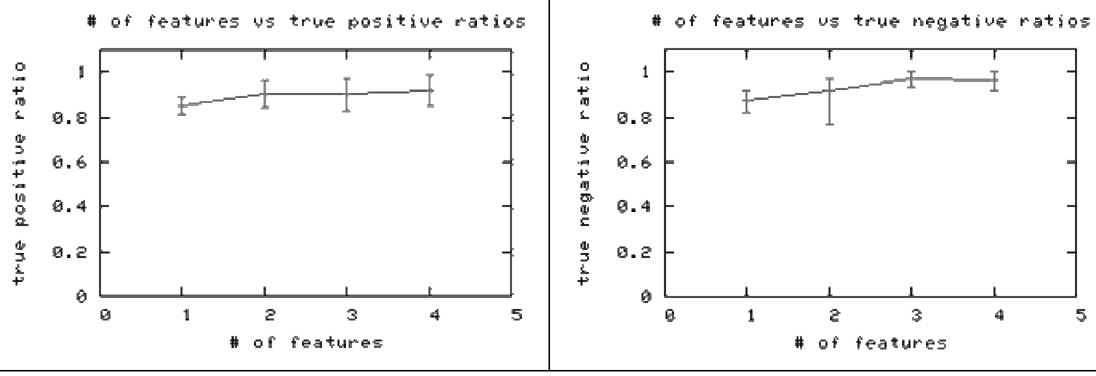

Figure 17 The effect of number of relevant features on the true positive and true negative percentages for the STRIP-PASS method.

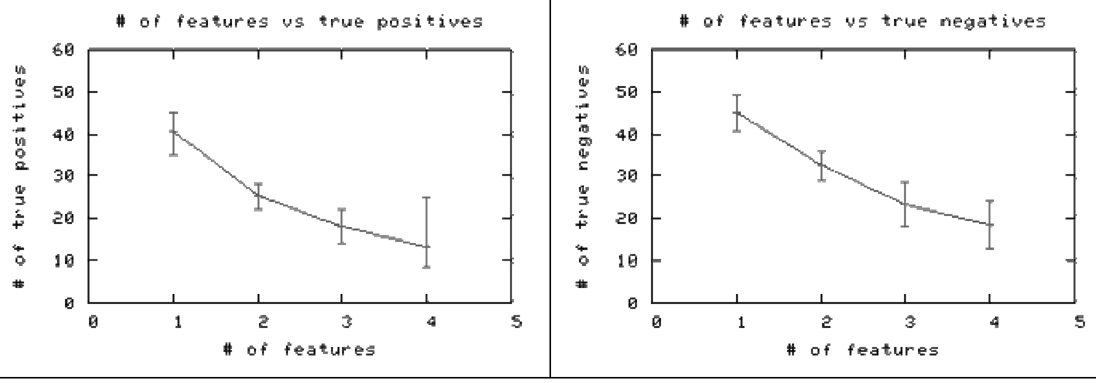

Figure 18 The effect of number of relevant features on the number of true positives and true negatives for the STRIP-PASS method.

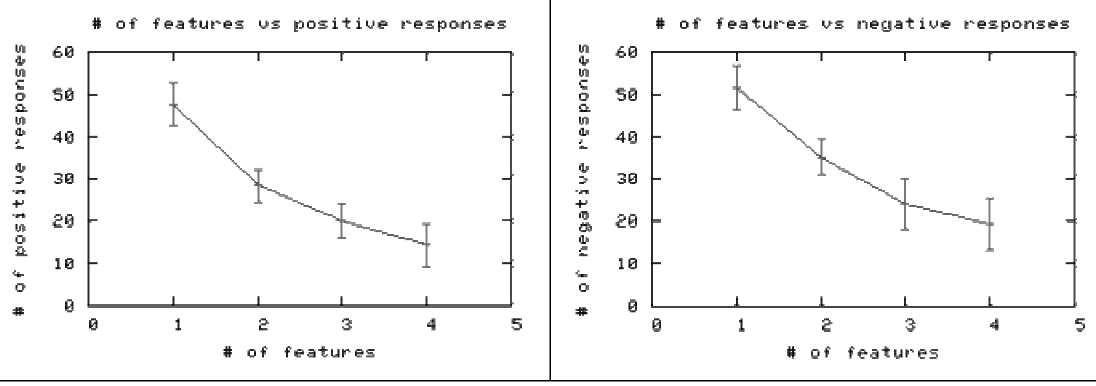

Figure 19 The effect of number of relevant features on the number of positive and negative responses for the STRIP-PASS method.

The performance of the system with threshold was analyzed for 100 trials each consisting of 100 objects. The mean, maximum and minimum values are shown in Figure 4. The accuracy of predictions now rose to 94% for the positive responses and to 98% for the negative responses. The decrease in the number of responses is also encountered for the threshold case(see Figure 20).

Compared to the no threshold case, the threshold case has better performance. In addition to better averages, the variances are significantly lower. However, the number of positive responses decreases from 48 to 25. The robot remains undecided for almost half of the reachable objects.

It can be inferred from the results that the number of relevant features affects success ratio positively, and number of responses negatively. The choice should depend on the precision that we expect from the robot. If the robot is required to be more responsive to the environment, a decrease in accuracy should be expected. If the robot is required to be as accurate as possible, a decrease in the number of responses should be expected. This may be the case when the robot's inaccurate decisions cause fatal outcomes. Another approach to this case can be to let the robot continue experimenting if it can not reach a solid decision. The robot will respond to more and more objects in time as it collects more information for the relevancy of the strips.

5.2.3. WINNOW Algorithm

The WINNOW algorithm is proven to be successful as a feature selection method when there exists a large number of irrelevant features[35]. Actually, it does not select the relevant features but assigns weights to each feature and update them after each observation. A downside of using Winnow algorithm is that it requires discrete values for the features. Thus, we have configured our system to better utilize Winnow for our case.

Positively and negatively relevant features were presented to the Winnow algorithm separately according to their relevancy determined during the strip

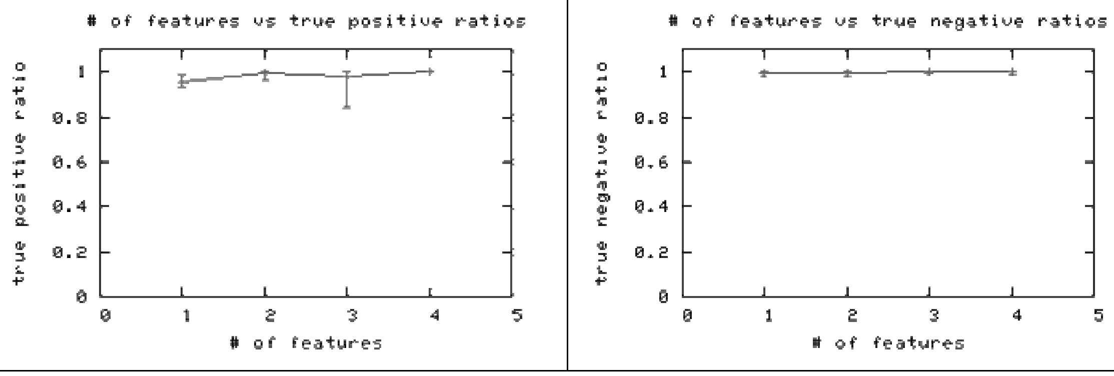

Figure 20 The effect of number of relevant features on the true positive and true negative percentages for the STRIP-PASS method with threshold.

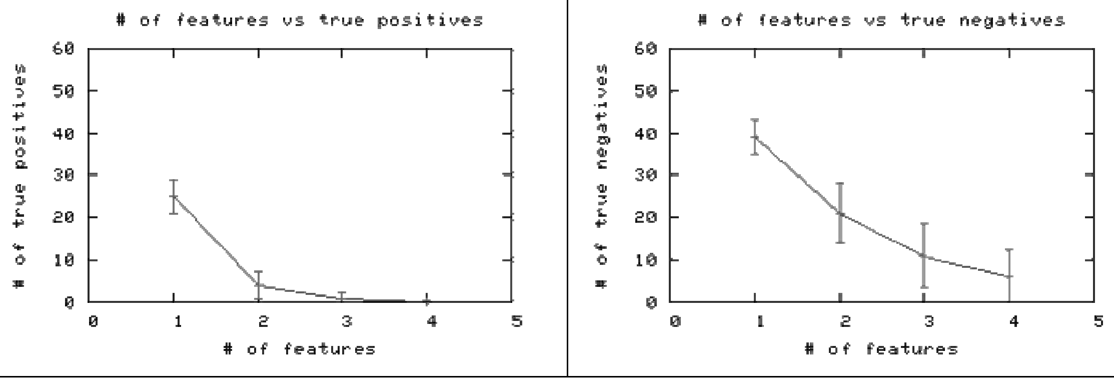

Figure 21 The effect of number of relevant features on the number of true positives and true negatives for the STRIP-PASS method with threshold.

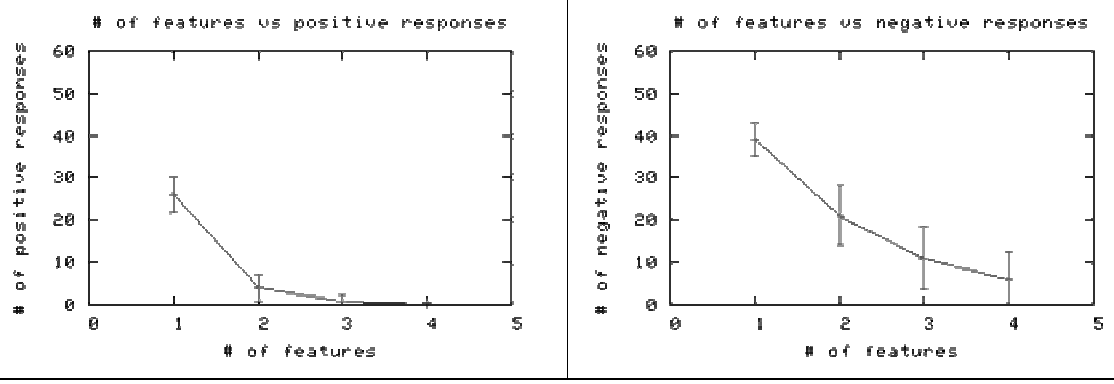

Figure 22 The effect of number of relevant features on the number of positive and negative responses for the STRIP-PASS method with threshold.

formation phase. When a new object is observed, the values of the relevant features in the new object are checked in the strips belonging to these features. If the value is found in a positively relevant strip, and not in a negatively relevant strip, the robot gave a positive answer. If the value is found in a negatively relevant strip, and not in a positively relevant strip, the robot gave a negative answer. In case of a wrong answer, the weights of the corresponding features were updated.

Results of experiments averaged over 100 trials each consisting of 100 objects are presented in Figure 23, 24 and 25. The true positive and true negative ratios did not change significantly with the change in number of relevant features, even when all the features were included. This is compatible with the argument in [36] telling that Winnow is robust considering the number of irrelevant features.

Although the performance is consistent when the true positive and negative ratios are considered, the number of positive and negative responses changed greatly for each trial. For some trials, the robot gave positive response for 90 objects.

The WINNOW algorithm[35]

1. Initialize the weights $w_1,\ldots, w_n$ of the features to 1.
2. Given an example $(x_1,\ldots, x_n)$, output 1 if $w_1 x_1+\ldots+ w_n x_n >= n$, and output 0 otherwise.
3. If the algorithm makes a mistake:
    a. If the algorithm predicts negative on a positive example, then for each x_i equal to 1, double the value of w_i.
    b. If the algorithm predicts positive on a negative example, then for each x_i equal to 1, cut the value of w_i in half.
4. Go to 2.

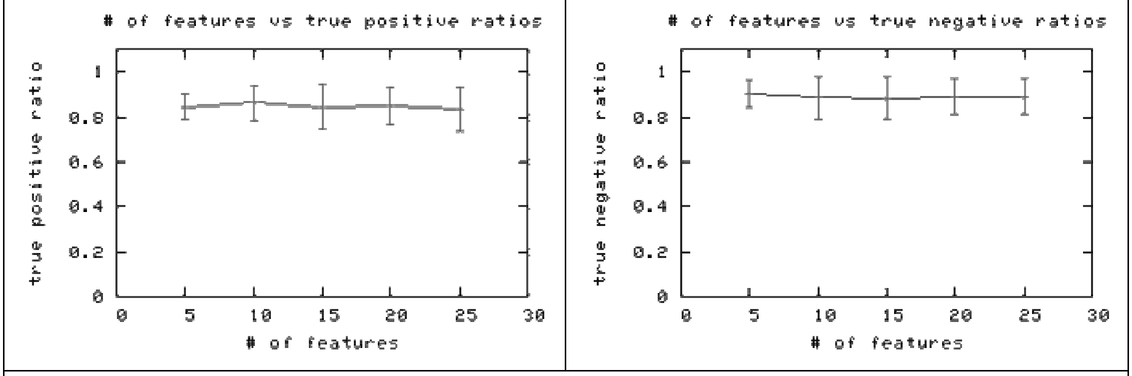

Figure 23 The effect of number of relevant features on the true positive and true negative percentages for the WINNOW algorithm.

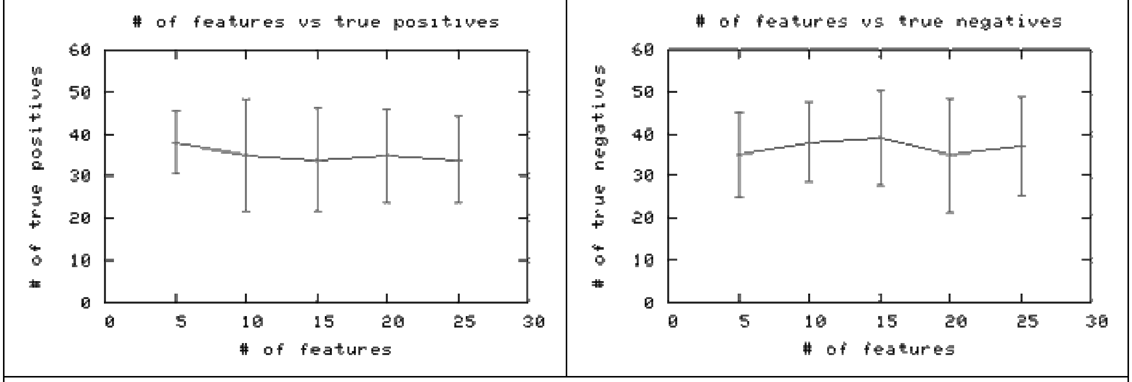

Figure 24 The effect of number of relevant features on the number of true positives and true negatives for the WINNOW algorithm.

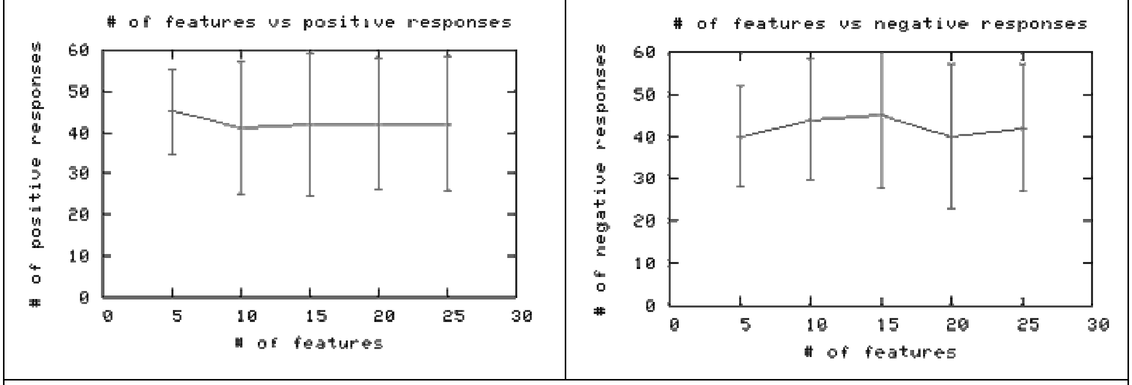

Figure 25 The effect of number of relevant features on the number of positive and negative responses for the WINNOW algorithm.

5.2.4. WINNOW Algorithm with threshold

Positively and negatively relevant features were presented to the Winnow algorithm separately according to their relevancy determined during the strip formation phase just like in the threshold case. The difference is that when the new object is observed, the values of the relevant features in the new object are checked only in the strips that pass a certain threshold. If the value is found in a positively relevant strip, and not in a negatively relevant strip, the robot gave a positive answer. If the value is found in a negatively relevant strip, and not in a positively relevant strip, the robot gave a negative answer. In case of a wrong answer, the weights of the corresponding features were updated.

The average success percentage of true positives is above 85% for any number of relevant features. Compared to the no threshold case, there is a 2% improvement in averages. There is much better improvement for the variances. For 20 relevant features, the worst performance is around 50% for the no threshold case and 82% for the threshold case.

There is also a 2% improvement in the true positive percentages. The improvement is more significant for the variance of the true negative percentages. Worst performance rises from 55% to 93% in the threshold case.

The high variance in positive and negative responses is observed also for the threshold case. However, for the case of 10 relevant features, the variance improves significantly, pulling the lowest value to 26 from 0 for positive responses.

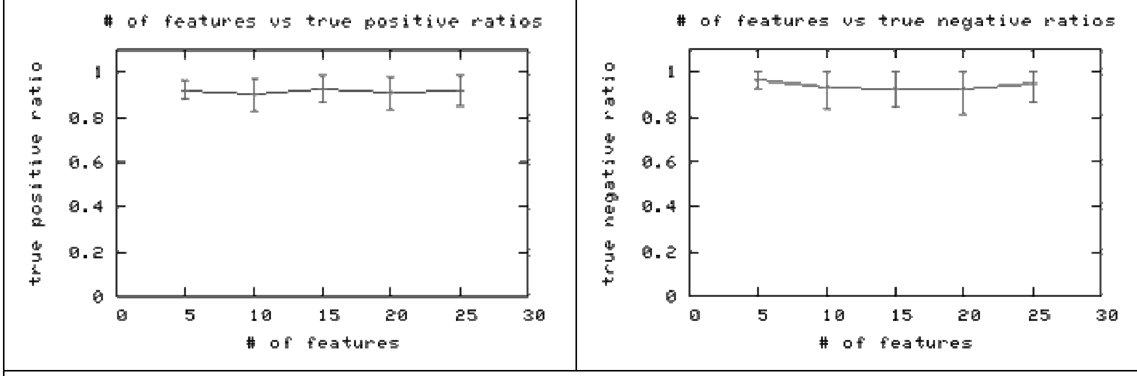

Figure 26 The effect of number of relevant features on the true positive and true negative percentages for the WINNOW algorithm with threshold.

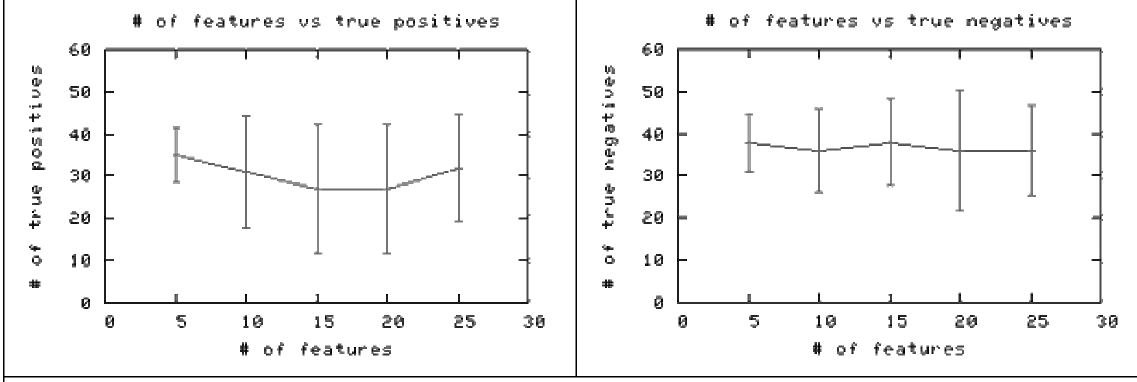

Figure 27 The effect of number of relevant features on the number of true positives and true negatives for the WINNOW algorithm with threshold.

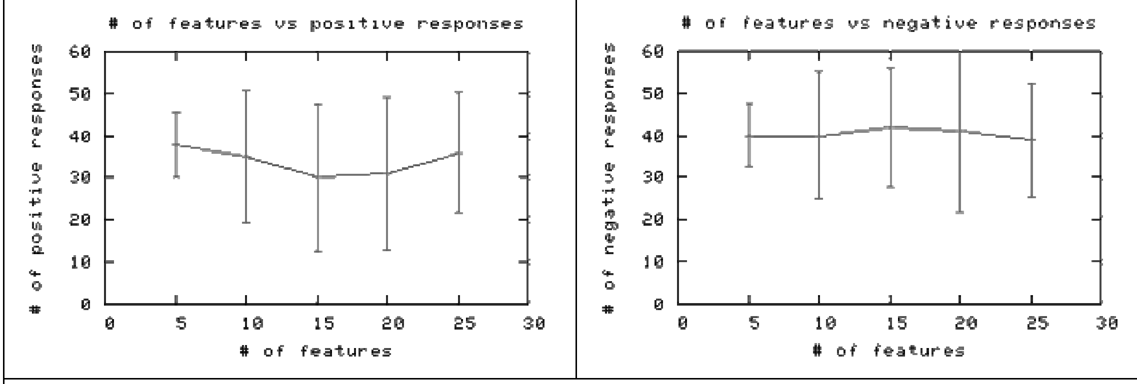

Figure 28 The effect of number of relevant features on the number of positive and negative responses for the WINNOW algorithm with threshold.

5.2.5. Comparison of the four methods

Results of experiment are summarized in Figures 29, 30 and 31. For each method, best cases are selected. Best case for the STRIP-PASS with threshold, no threshold and with example selection is when only one relevant feature is selected. Winnow’s best case is determined as when there are 10 relevant features.

For true positive and true negative ratios, STRIP-PASS method with threshold performs slightly better than other methods. WINNOW with threshold follows it.

Considering number of true positives, STRIP-PASS without threshold outperforms other methods. WINNOW with selection follows behind. Methods with threshold has lower true positive values since applying threshold restricts decisions considerably in order to get more accurate responses. For the number of true negatives, methods with threshold have higher values. The reason is the fact that there are many small negatively relevant strips, unlike positively relevant strips. When a threshold is applied, these small strips are discarded, but many relevant strips remain. So the number of responses does not decrease significantly and at the same time, the performance increases since more relevant features remain.

The number of positive and negative responses is higher for STRIP-PASS method than the other methods.

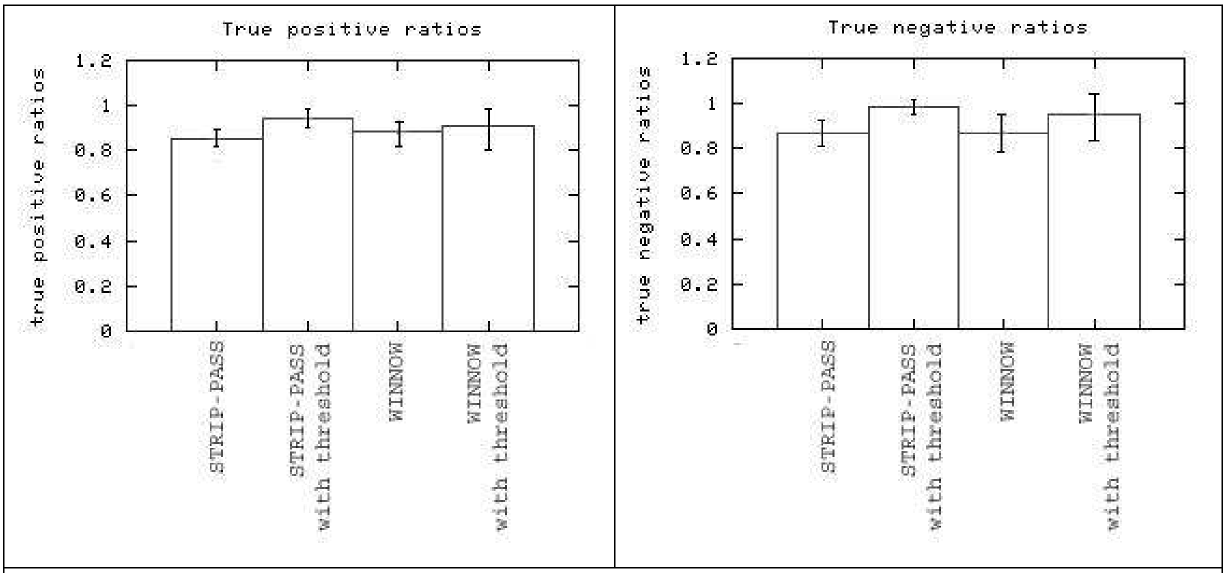

Figure 29 Comparison of relevant feature utilization methods with respect to the true positive and true negative ratios.

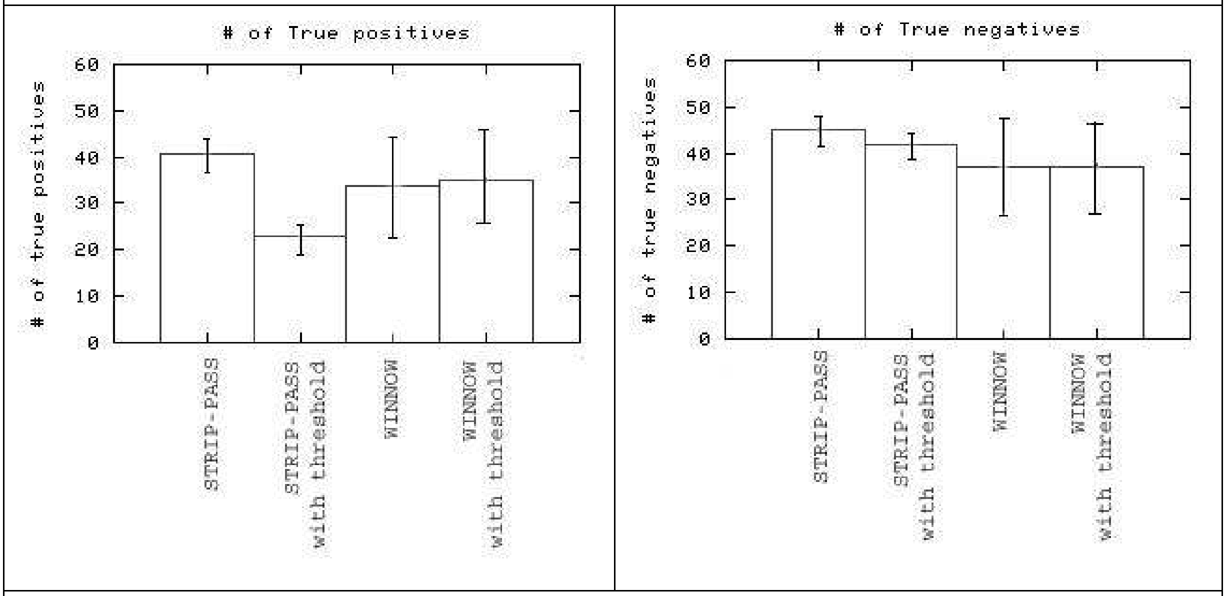

Figure 30 Comparison of relevant feature utilization methods with respect to the number of true positives and true negatives.

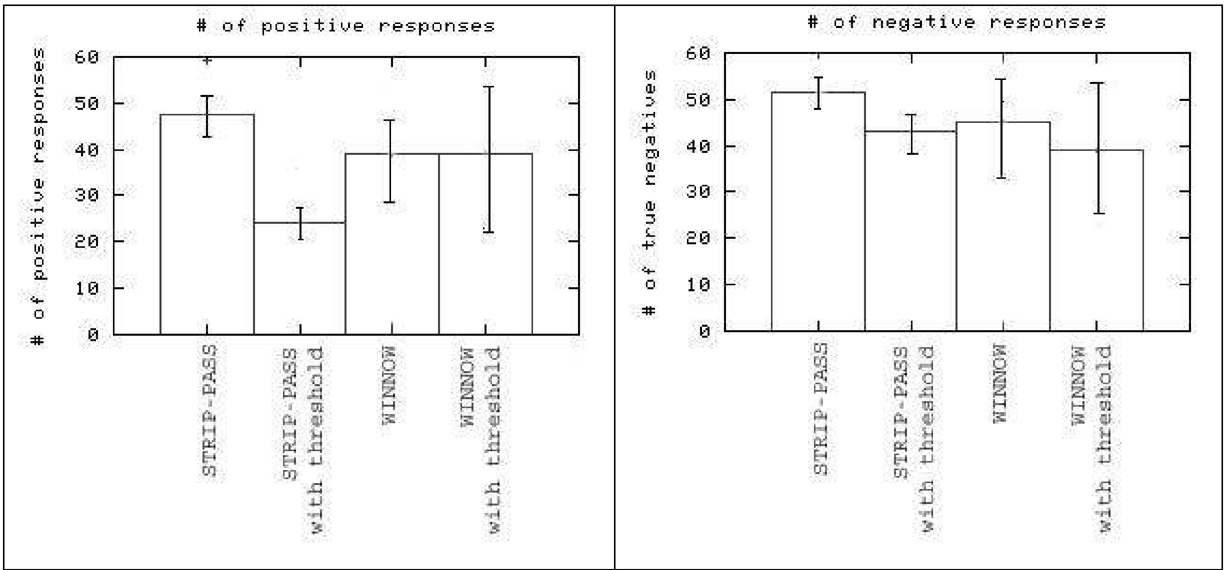

Figure 31 Comparison of relevant feature utilization methods with respect to the number of positive and negative responses.

CHAPTER 6

CONCLUSION

The affordance concept that has emerged in cognitive science has important implications for autonomous robots. In this thesis, we reviewed the affordance concept for autonomous robot control and proposed that invariant features of objects that support a specific affordance can be learned. We used a physics-based robot simulator to study the reachability affordance on the simulated KURT3D robot model. We proposed that, through training, the values of each feature can be split into strips, which can then be used to detect the relevant features and their characteristics.

Detection and use of relevant features and their characteristics have important implications. First, a robot looking for objects that would support a desired affordance can turn on the detectors of only the relevant features while leaving the rest turned off. Given that a robot would probably have hundreds of feature detectors; such ability allows a great perceptual speed-up. Second, our analysis shows that it is possible to achieve higher prediction accuracy on the affordance support of novel objects by using only the relevant features. Use of all the features available actually degrades the prediction accuracy. This is an important gain, since failures can have high costs in robotics and better prediction accuracy is desired.

The method we proposed enables a robot to perceive specific affordances in the environment. The next question is whether the affordances to be perceived should be pre-implemented in the robot, or the robot should discover novel affordances itself. The set of affordances that belongs to an agent-object system is infinite. If the robot is supposed to discover a subset of this set itself, goals of the robot can guide the discovery process. For example, let the goal of the robot be to reach from one place to another. If a chair blocks the path, the robot may learn that a chair with specific features affords blocking the path. It may not analyze the reachability of this chair. However, the robot needs a mechanism to determine that there is a change in its state caused by that object, in this case the robot can not continue navigation because it is blocked by the chair. For many cases, it may not be easy to decide cause of the change, and which change to consider. The chair can be seen responsible for the blocking but there can be another object in the ground that is out of view. Besides, the change in robot's state can be deceleration, lowering of magnet, breaking of a wheel, increase in the shaking, so the robot should decide which change to consider.

Finally, the work presented here only scratches the top of an interesting topic. Feature selection is already one of the active topics in computer vision and will probably be active in robotics, too. Detection of relevant features and their characteristics will allow robot to deal with novel objects while being highly responsive.

REFERENCES

[1] Mars Exploration Rover Mission. (2006, February). Retrieved January 8, 2006, from
http://marsrovers.jpl.nasa.gov/home/

[2] Trends in Japan. (2002, July). Retrieved January 6, 2006, from
http://web-japan.org/trends01/article/020705sci_r.html

[3] Grand Challenge.(2005, December). Retrieved January 3, 2006, from
http://www.darpa.mil/grandchallenge/

[4] Four-Legged League. (n.d.). Retrieved January 6, 2006, from
http://www.tzi.de/4legged/bin/view/Website/WebHome

[5] D. Whitehouse. (2005, July). Japanese develop 'female' android. BBC News. Retrieved from
http://news.bbc.co.uk/1/hi/sci/tech/4714135.stm

[6] iRobot Corporation: Scooba. (n.d.). Retrieved January 2, 2006, from
http://www.irobot.com/sp.cfm?pageid=128

[7] I, Robot. (n.d.). Retrieved January 4, 2006, from
http://www.irobotmovie.com/

[8] SRI Technology: Shakey the Robot. (n.d.). Retrieved December 10, 2005, from http://www.sri.com/about/timeline/shakey.html

[9] R. R. Murphy, Introduction to AI Robotics. MIT Press, 2000. ISBN:026213383.

[10] V. Braitenberg, Vehicles: Experiments in Synthetic Psychology. Cambridge, Mass.: MIT Press, 1984. ISBN:062117110523

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

[12] R. R. Murphy, R. C. Arkin, "An Architecture for Action-Oriented Sensor Fusion", in Proceedings of the International Conference on Intelligent Robotics and Systems (IROS), Raleigh, N.C., 1992.

[13] E. Gat, "Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots", in Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92). pp. 809-815, 1992.

[14] K. Konolige, K. Myers, E. Ruspini, and A. Saffiotti, "The Saphira architecture: a design for autonomy", Journal of Experimental and Theoretical Artificial Intelligence, vol. 9, pp. 215-235, 1997.

[15] P.J. Besl, R. C. Jain, "Three-dimensional object recognition", ACM Computing Surveys (CSUR), vol. 17 (1), Annals of discrete mathematics, 24, pp. 75 - 145, 1985.

[16] A. Morales, P. J. Sanz, and A. P. del Pobil, "How can I, robot, pick up that object with my hand ?", in European Conference on Artificial Intelligence (ECAI 2004), International Cognitive Robotics Workshop. pp. 78-83, Valencia, 2004.

[17] M. Broxvall, S. Coradeschi, L. Karlsson, and A. Saffiotti, "Recovery planning for ambiguous cases in perceptual anchoring", in Proceedings of the AAAI 2005 Conference. Pittsburg, PA, July 2005.

[18] G. De Giacomo, L. Iochhi, D. Nardi, and R. Rosati, "A theory and implementation of cognitive mobile robots", J. Logic Computation, 9(5):759-785,1999.

[19] D. Buchsbaum, and B. Blumberg, "Imitation as a first step to social learning in synthetic characters: a graph-based approach", in Proceedings of the 2005 ACM Siggraph/Eurographics Symposium on Computer Animation (SCA '05). ACM Press, New York, 2005.

[20] C. Breazeal, D. Buchsbaum, J. Gray, D. Gatenby, and B. Blumberg, "Learning from and about others: towards using imitation to bootstrap the social understanding of others by robots", Artificial Life, vol. 11, No. 1-2, pp. 31-62, 2005.

[21] I. Cos-Aguilera, L. Cañamero, G. M. Hayes, "Motivation-driven learning of action affordances", in Proceedings of Agents that Want and Like: Motivational and Emotional Roots of Cognition and Action, Workshop of the AISB'05 Convention, Hertfordshire, UK, 2005.

[22] J. J. Gibson, The Ecological Approach to Visual Perception. Lawrence Erlbaum Associates, Inc., 1990. ISBN: 0898599598

[23] A. Slater, Perceptual Development: Visual, Auditory and Speech Perception. Taylor & Francis, Inc., 2000. ISBN: 0863778518

[24] P. Fitzpatrick, G. Metta, L. Natale, L. S. Rao, and Sandini, "Learning about objects through action: initial steps towards artificial cognition", in IEEE International Conference on Robotics and Automation (ICRA 2003), Taipei , Taiwan, 2003.

[25] A. Stoytchev, "A Behavior-Grounded Representation of Tool Affordances", in Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2005.

[26] A. Stoytchev, "Toward Learning the Binding Affordances of Objects: A Behavior-Grounded Approach", in Proceedings of AAAI Symposium on Developmental Robotics, 2005.

[27] M. Wünstel and R. Moratz, "Automatic Object Recognition within an Office Environment", Canadian Conference on Computer and Robot Vision (CRV 2004), IEEE, 2004.

[28] M.A. Lewis, "Perception Driven Robot Locomotion", Journal Robot Society of Japan, vol. 20, 3, pp. 51-56, 2002.

[29] M.A. Lewis and L.S. Simó, "Certain Principles of Biomorphic Robots", Autonomous Robots, vol. 11, pp. 221-226, 2001.

[30] A.C. Slocum, D.C. Downey, and R.D. Beer, "Further experiments in the evolution of minimally cognitive behavior: From perceiving affordances to selective attention", in From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior (J. Meyer, A. Berthoz, D. Floreano, H. Roitblat and S. Wilson, eds), pp. 430-439, MIT Press, Paris, France, 2000.

[31] R. Smith. (2004). Open Dynamics Engine v0.5 User Guide. Retrieved December 5, 2005, from
http://www.ode.org/ode-latest-userguide.html

[32] R. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson, "Is visually guided reaching in early infancy a myth?", Child Development, vol. 64, pp. 1099-1110, 1993.

[33] R. Keen, R. Carrico, M. Syulvia, and N. Berthier, "How infants use perceptual information to guide action", Developmental Science, vol. 6, pp. 221-231, 2003.

[34] L. Claxton, R. Keen, and M. McCarty, "Evidence of motor planning in infant reaching behavior", Psychological Science, vol. 14, pp. 354-356, 2003.

[35] D. Heffernan and J. A. Thomson, "Perceiving what is reachable during rapid growth in adolescence", in Studies in Perception and Action IV. Hillsdale (M. A. Schmuckler and J. M. Kennedy, eds), pp. 149-152, Lawrence Erlbaum Associates, 1997.

[36] J. R. Pijpers, F. C. Bakker, F. Holsheimer, "Fatigue and reachability", in Studies in Perception and Action IV. Hillsdale (M. A. Schmuckler and J. M. Kennedy, eds), pp. 145-148, Lawrence Erlbaum Associates, 1997.

[37] P. Rochat, N. Goubet, and S. J. Senders, "To reach or not to reach? Perception of body effectivities by young infants", Infant & Child Development, vol. 8, pp. 129–148, 1999.

[38] A. Blum, P. Langley, "Selection of relevant features and examples in machine learning", Artificial Intelligence, vol. 97, pp. 245-271, 1997.

[39] A. Blum, "Empirical support for winnow and weighted-majority algorithms: results on a calendar scheduling domain", Machine Learning, vol. 26(1), pp. 5-23, 1997.