

FUNCTION AND APPEARANCE-BASED EMERGENCE OF OBJECT CONCEPTS
THROUGH AFFORDANCES

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

BY

İLKAY ATIL

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

NOVEMBER 2010

Approval of the thesis:

**FUNCTION AND APPEARANCE-BASED EMERGENCE OF OBJECT CONCEPTS
THROUGH AFFORDANCES**

submitted by **İLKAY ATIL** in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering Department, Middle East Technical University by,

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

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

Asst. Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering Department, METU**

Asst. Prof. Dr. Erol Şahin
Co-supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Fatoş Yarman Vural
Computer Engineering, METU

Asst. Prof. Dr. Sinan Kalkan
Computer Engineering, METU

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

Prof. Dr. Göktürk Üçoluk
Computer Engineering, METU

Asst. Prof. Dr. Didem Gökçay
Informatics Institute, METU

Date:

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

Name, Last Name: İLKAY ATIL

Signature :

ABSTRACT

FUNCTION AND APPEARANCE-BASED EMERGENCE OF OBJECT CONCEPTS THROUGH AFFORDANCES

Atıl, İlkey

M.Sc., Department of Computer Engineering

Supervisor : Asst. Prof. Dr. Sinan Kalkan

Co-Supervisor : Asst. Prof. Dr. Erol Şahin

November 2010, 58 pages

One view to cognition is that the symbol manipulating brain interprets the symbols of language based on the sensori-motor experiences of the agent. Such symbols, for example, what we refer to as nouns and verbs, are generalizations that the agent discovers through interactions with the environment. Given that an important subset of nouns correspond to objects (and object concepts), in this thesis, how function and appearance-based object concepts can be created through affordances has been studied. For this, a computational system, which is able to create object concepts through simple interactions with the objects in the environment, is proposed. Namely, the robot applies a set of built-in behaviors (such as pushing, lifting, grasping) on a set of objects to learn their affordances, through which objects affording similar functions are grouped into object concepts. Moreover, the thesis demonstrates that the discovered object concepts are beneficial for learning new tasks by analyzing the learning performance of learning a new task with and without object concepts.

Keywords: concepts, affordances, multi-task learning, language embodiment and grounding

ÖZ

İŞLEV VE GÖRÜNÜM TEMELLİ NESNE KAVRAMLARININ SAĞLARLIKLAR ARACILIĞIYLA OLUŞTURULMASI

Atıl, İlkey

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Sinan Kalkan

Ortak Tez Yöneticisi : Yrd. Doç. Dr. Erol Şahin

Kasım 2010, 58 sayfa

Bilişsellik hakkındaki bir görüşe göre semboller işleyen beyin, dile ait sembolleri kişinin duyu-motor deneyimleri üzerinden yorumlamaktadır. Bu tür semboller, örneğin isim ve fiil dediklerimiz, kişinin çevreyle olan etkileşimlerinden genellemeler yapmasıyla keşfedilmektedir. İsimlerin büyük çoğunluğunun nesnelere (ve nesne kavramlarına) karşılık geldiği düşünülürse, bu tezde, işleyiş ve görünüm temelli nesne kavramlarının sağlarlıklar aracılığıyla nasıl elde edilebileceği araştırılmıştır. Bunun için, ortamdaki nesnelerle basit etkileşimler üzerinden nesne kavramları oluşturabilen hesapsal (ing. computational) bir sistem sunulmuştur. Daha detaylı anlatmak gerekirse, bir robot sahip olduğu belirli davranışları (itme, kaldırma, tutma gibi) nesnelere uygulayarak sağlarlıkları öğrenmekte ve bu yolla benzer fonksiyonları sağlayan nesneleri gruplayarak nesne kavramlarını oluşturmaktadır. Dahası, tezdeki çalışmalar keşfedilen nesne kavramlarının yeni görevlerin öğrenilmesinde fayda sağladığını yeni görevlerin öğrenme performanslarının nesne kavramlarının kullanılıp kullanılmamasıyla nasıl değiştiğinin incelenmesi aracılığıyla göstermektedir.

Anahtar Kelimeler: kavramlar, sađlarlıklar, çoklu-öğrenme, dilin cisimleştirilmesi ve temelendirilmesi

To my beloved Mom and Dad

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my advisor, Sinan Kalkan for his endless support, guidance and patience through my studies. He guided me at every level of this work and made this thesis possible to finish, I couldn't even think of completing this thesis without him.

I'd like to thank to my co-supervisor, Erol Şahin for his guidance and uncanny ability of seeing the solution when we lose our hopes after many hours of all-nighters. I also thank you for giving us such a great chance to work at your robotics laboratory and on the iCub, it was priceless. I hope we didn't wasted such a great opportunity.

My special thanks goes to my friends and colleagues who made working in the lab such a great experience. I'd like to thank to Barış Akgün for his joyfulness and countless hours of chit-chat and nerd conversations. Fatih Gökçe for being such a honest and great friend, I still hope to make work with you in the future. Hande Çelikkanat for her merry laughs and helping me whenever I needed. Nilgün Dağ for our joint-work during the thesis and the joyful hours we spent. Doruk Tunaoglu for his friendship and our shared interest of rock music. Emre Uğur for helping me from the other side of the planet and Tahir Bilal for his knowledge and companionship. I'd also thank to the new members of our lab; Kadir Fırat Uyanık, Güven İşçan, Mustafa Parlaktuna, Onur Yürüten, Çiğdem Avcı, Asil Kaan Bozcuoğlu and Yiğit Çalışkan for making our lab even more joyful.

My gratitudes goes to my friends İren Berk Özalp, Ece Beyazıt, Caner Kavakoğlu, Oğuzcan Samsun, Onur Şirikçi and many more who I couldn't mentioned here.

Dear mother and father, I don't even know how to thank you enough for being the best parents in the world. When I want to thank you, words become so meaningless. I also thank to all of my relatives for supporting me with their good wishes and preys during my studies.

This work is partially funded by the EU project ROSSI (FP7-ICT-216125) and by TÜBİTAK (Turkish Scientific and Technical Council) BİDEB Graduate Study Grant (2210).

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ	v
DEDICATION	vii
ACKNOWLEDGMENTS	viii
TABLE OF CONTENTS	ix
LIST OF FIGURES	xi
CHAPTERS	
1 Introduction	1
1.1 Contributions	2
1.2 Organization of the Thesis	5
2 Background and Literature Survey	6
2.1 Perception-Action-Language	6
2.2 Affordances	9
2.2.1 Literature on Usage of Affordances	12
2.3 Object Concepts	13
2.3.1 Literature on Object Concepts	14
2.4 Multi-Task Learning	15
3 Methods and Experimental Setup	17
3.1 Range Camera	17
3.2 Behaviors	18
3.3 Data	19
3.4 Feature Extraction	20
3.5 Feature Selection	21
3.6 Unsupervised Clustering: Robust Growing Neural Gas	22

3.7	The System Proposed in This Thesis	27
3.7.1	Obtaining Effect Clusters	27
3.7.2	Feature Selection: Analysis of the Entity Space	29
3.7.3	Creation of Object Concepts: Unsupervised Clustering of the Entity Space	29
3.7.4	Benefits of the Object Concepts	31
3.8	Performing Multi-Task Learning	32
4	Acquired Concepts and Multi-Task Learning	34
4.1	Concepts	34
4.1.1	Push-left and Push-right	34
4.1.2	Grasp	36
4.1.3	Lift	39
4.1.4	Rotate	40
4.2	Multi-Task Learning	42
4.2.1	Object Concepts and Learning Time	42
4.2.1.1	First Experiment - Similar Behaviors	42
4.2.1.2	Second Experiment - Different Behaviors	43
4.2.2	Object Concepts and Training Set Size	46
4.2.2.1	Third Experiment: Fixed Number of Epochs	47
4.2.2.2	Fourth Experiment: Fixed Mean Squared Error	47
4.3	Discussion over Using Single Feature for Acquisition of Concepts	49
5	Discussion	53
5.1	Future Work	54
	REFERENCES	55

LIST OF FIGURES

FIGURES

Figure 1.1	Different approaches to the perception of affordances. The entity space is the space of the initial views of the objects whereas the effect space is the space of the effects that can be achieved through executing behaviors on the entities in the entity space. The data in the entity and the effect space is raw in the sense that the information is as simple and low-level as possible.	3
Figure 2.1	Relations of entities, behaviors and effects in our affordance formalization .	13
Figure 2.2	Learning four tasks as independent neural networks	16
Figure 2.3	Learning four tasks in a single neural network	16
Figure 3.1	SwissRange SR4000 Range Camera	18
Figure 3.2	Behaviors used for interacting with objects	19
Figure 3.3	Used objects in the experiments. Object set contains three different shapes; spheres, cubes and cylinder with three different sizes; small, medium and big. . .	20
Figure 3.4	Visual description of the shape indexes. This figure is taken from [13] by the courtesy of Dağ N.	21
Figure 3.5	Steps of the RGNG	25
Figure 3.6	Minimum Description Length values for different number of clusters . . .	26
Figure 3.7	Comparison of x-Means and RGNG clustering results	26
Figure 3.8	The three steps of Our Proposed System	28
Figure 3.9	Effects of behaviors on all object types	29
Figure 3.10	Multi-Task learning method	32
Figure 4.1	Object concepts for push-left and push-right	35

Figure 4.2	Weights of features for the push-left and push-right behaviors calculated by ReliefF	35
Figure 4.3	Detailed visualization of the acquired concepts for push-left and push-right	36
Figure 4.4	Object concepts for grasp	37
Figure 4.5	Weights of features for the grasp behavior calculated by ReliefF	38
Figure 4.6	Detailed visualization of the acquired concepts for grasp	38
Figure 4.7	Object concepts for lift	39
Figure 4.8	Detailed visualization of the acquired concepts for lift	39
Figure 4.9	Object concepts for rotate	40
Figure 4.10	Weights of features for the rotate behavior calculated by ReliefF	41
Figure 4.11	Detailed visualization of the acquired concepts for rotate	41
Figure 4.12	Structure of the MTL network and previously learned affordances	43
Figure 4.13	The structure of the STL network	44
Figure 4.14	Learning times of push-right affordance by MTL and STL	44
Figure 4.15	The structure of the Multi-Task Learning and previously learned affordances	45
Figure 4.16	Learning times of push-right affordance by MTL and STL	46
Figure 4.17	Comparison of performances of MTL and STL	48
Figure 4.18	Comparison of learning times of MTL and STL	49
Figure 4.19	Data distribution of our artificial features	50
Figure 4.20	Combined data distribution of our artificial features	51
Figure 4.21	Acquired object concepts by using different number of features	52

CHAPTER 1

Introduction

We use language to communicate with others. Such communication is possible because we share the same meanings for the words we use. The meaning of the word “chair” is given by the concept created by our sensori-motor experiences with different kinds of chairs, which involve the appearances as well as the functionalities of chairs. We form our concepts ourselves in an embodied fashion and use them to acquire a language. All of these may sound easy but the underlying mechanisms which enable us to do all of these without our conscious efforts are still waiting to be discovered.

Language acquisition is a topic at the intersection of many different disciplines; linguistics, psychology, neuroscience, cognitive science, et cetera. People from these disciplines try to understand the underlying mechanisms that enable us to learn and speak languages. In cognitive science, many researchers suggest that language is learned based on sensori-motor representations [6, 16, 24, 50, 54] of the environment. Babies play with toys, try to shake them, bite them, etc. What seems like a sheer child game actually serves as building sensori-motor representations as a grounding for future skills. This enables learning gradually more complex tasks in the future. Still, how those representations are formed, how they are stored and used are hardly known. In this thesis, we propose a system which learns certain affordances and create concepts about the objects as grounded on sensori-motor information at the same time.

Affordances, as first coined by J.J. Gibson [23], describe the action possibilities presented by the environment to an agent. Affordance relations can be analyzed to discover the properties of objects enable us to perform certain actions and what kind of objects share such properties. Using such informations derived from affordances, we can find the functional and appearance-based similarities between objects which enables us to create object concepts.

When Gibson described the term affordances, he stated that affordances are “directly perceivable”, meaning that we can perceive the action possibilities offered by the environment without recognition. For example, detecting a sittability affordance does not require the person to recognize that the object is a chair. Hence, the sittability affordance is not bounded with a chair but with direct perception of features (a small flat surface strong enough to carry a person).

The idea of direct perception was strongly disagreed by some researchers. Ullman [60] strongly counter-argued the idea of direct perception with a memo in 1980. Fodor and Pylyshyn discussed whether it is possible to have directly perceivable affordances and how direct this perception can be [18]. Sun et al. [56] show that using object categories for affordance learning (indirect perception) yields better results than using direct perception. Although Sun’s work demonstrates usefulness of object categories, this categorization is more like recognition of objects (independent categorization from robot’s abilities, solely depending on appearance) which in a way does not agree with the Gibsonian view of affordances.

Different from the works above, our stand about the directness of affordances is more similar to what Neisser states as his own understanding from direct perception. In his editorial book [41], Neisser says that:

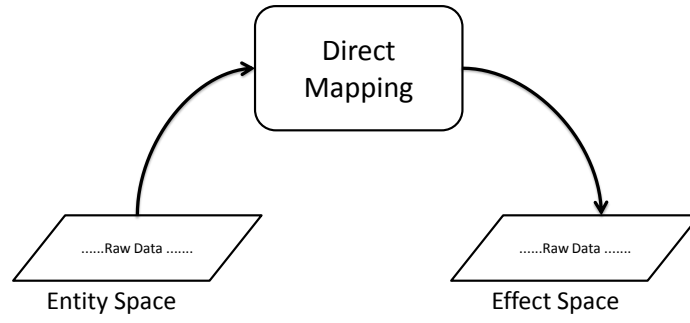
“At the basic level¹, then, objects are categorized by their looks and by their affordances. Because affordances can be perceived directly, [...], both of these criteria are essentially perceptual.”

We can show the difference of these ideas about perception of affordances graphically. Figure 1.1 shows the direct and indirect perception of affordances and our approach to the perception of affordances in this thesis.

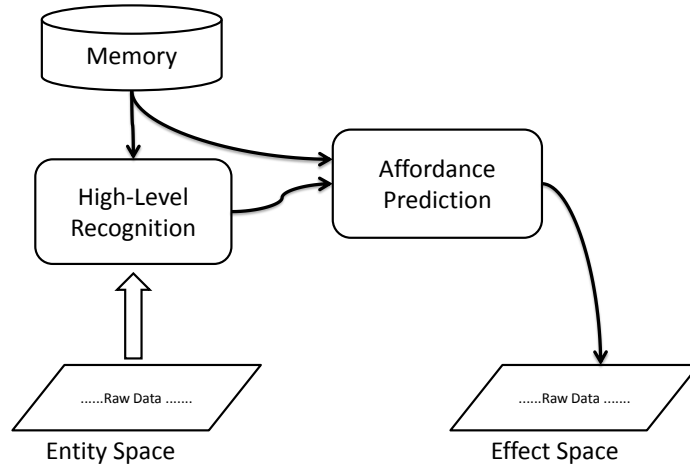
1.1 Contributions

This thesis makes the following contributions:

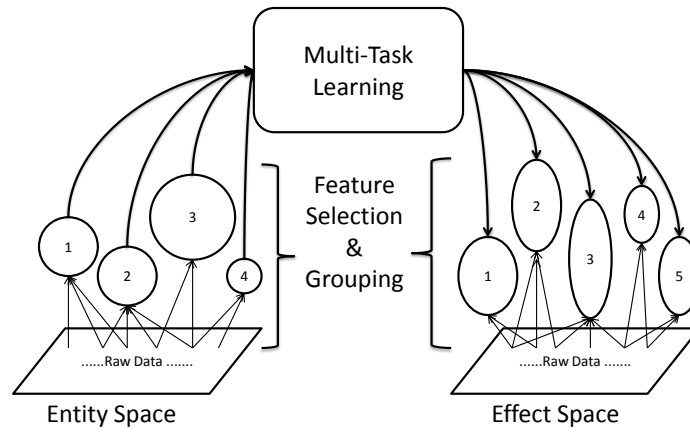
¹Concept of a basic level object category is described by Rosch and Mervis [48, 49, 38]. They state that objects are categorized hierarchically. In this hierarchy, the middle level corresponds to the basic level of categorization where categories like chair, bed, dog, cow reside.



(a) Direct Perception



(b) Indirect Perception



(c) Proposed System

Figure 1.1: Different approaches to the perception of affordances. The entity space is the space of the initial views of the objects whereas the effect space is the space of the effects that can be achieved through executing behaviors on the entities in the entity space. The data in the entity and the effect space is raw in the sense that the information is as simple and low-level as possible.

- A computational system which can form “object concepts” grounded on sensori-motor experiences of an agent has been proposed.
- Acquisition of object categories depend on both appearances and functionalities of objects.
- It has been demonstrated that affordance relations are suitable for acquiring object concepts.
- Object concepts were acquired incrementally through simple physical interactions with the objects in the environment.
- The proposed system can both use its own interactions with the environment or demonstrations of a teacher to acquire object concepts. This makes both embodied learning and learning-by-demonstration present in the same system.
- To show the benefits of object concepts, object concepts have been used as grounding for learning affordances. It has been shown that object concepts provide reduction on the necessary training set size and training times.
- The Multi-Task Learning approach was used to learn affordances.

These contributions have appeared in the following papers:

- N. Dağ, İ. Atıl, S. Kalkan, E. Şahin, *Emergence of Object and Verb Concepts through Affordances*, Cognitive Processing, International Quarterly of Cognitive Science, Special Issue on ”Cognitive Robotics - Perception-Action-Interaction: Systems and Architectures”, 2011, submitted.
- İ. Atıl, N. Dağ, S. Kalkan, E. Şahin, *Affordances and Emergence of Concepts*, 10th International Conference on Epigenetic Robotics (EPIROB), pages 11-18, 2010.
- N. Dağ, İ. Atıl, S. Kalkan, E. Şahin, *Learning Affordances for Categorizing Objects and Their Properties*, 20th International Conference of Pattern Recognition (ICPR), pages 3089-3092, 2010.
- B. Akgün, N. Dağ, İ. Atıl, T. Bilal, E. Şahin, *Unsupervised Learning of Affordance Relations on a Humanoid Robot*, 24th International Symposium on Computer and Information Sciences, pages 254-259, 2009.

1.2 Organization of the Thesis

The organization of this thesis is as follows: The following chapter is dedicated to background information and literature survey. In chapter 3, we describe our experimental setup, the methods used in our experiments and our system for creation of object concepts. Chapter 4 describes the acquired concepts and shows the benefits of using the developed object concepts in learning new affordances. Chapter 5 concludes the thesis with a discussion and a summary of the contributions and a list of extensions.

CHAPTER 2

Background and Literature Survey

In this chapter, we first present the literature about the relation between perception-action-language and how our proposed system differs from the literature. Then, we describe what affordances and their formalizations are and how they can be used to derive object concepts. Afterwards, we describe how we define object concepts and the literature about them. Finally, we present what Multi-Task Learning method is and why this method is used in this thesis.

2.1 Perception-Action-Language

The close relation between action and language has been pointed out by many experimental researchers which investigate perception, action and language processing mechanisms in humans and animals [2, 7, 24, 42, 47] . The common conclusion suggests that language is grounded in the organism's sensori-motor experiences. Animals with similar action capabilities (affordances) end up with a similar grounding for their sensori-motor experiences which as a result enables them to communicate based on the same meanings.

Association of meanings to symbols is a long debated topic in many disciplines, especially in Artificial Intelligence that thrives to define intelligence. Alan Turing [1] claimed that if a human engaging a non-verbal communication with a computer behind a curtain cannot decide whether the one behind the curtain is a computer or not, then we can claim that we have built a computer (program) which has intelligence. Now, this test is known as the Turing Test. It set a goal for much of the artificial intelligence research and caused philosophical debates on what intelligence is. J. Searle used a thought experiment, the Chinese room experiment [52], to argue that the ability of merely deceiving a human from behind a curtain does not make

the program intelligent, or cognitive. Searle claimed that, if the program is unaware of the meanings of the symbols it manipulates, then it is not reasonable to talk about intelligence.

Harnad [26] stated the problem of linking meanings with symbols as the *symbol grounding problem*. He argued that the gap between the meanings and symbols cannot be bridged by an external programmer and trying to close this gap by external programming is like “learning Chinese from the Chinese dictionary”. Instead, he proposed that symbols should be grounded in the *sensory projections* of objects and events in the environment. Harnad discusses three kinds of symbolic representations and their groundings as:

“(1) *iconic representations*, which are analogs of the proximal sensory projections of distal objects and events, and (2) *categorical representations*, which are learned and innate feature-detectors that pick out the invariant features of object and event categories from their sensory projections. [...] Higher-order (3) *symbolic representations* grounded in these [...] symbols, consist of symbol strings describing category membership relations (e.g., “An X is a Y that is Z”).”

The symbol grounding problem becomes more obvious for robots. Robots are able to interact with the environment physically and they might have similar sensori-motor experiences with humans depending on their physical bodies. However, their sensori-motor experiences will always be different from ours and the issue of how they can develop a shared set of symbols to represent the basic concepts of a language (nouns, verbs etc.) remains as an open question.

Rizzolatti et. al. [47] discovered a neuron system in monkeys that underlies such a shared grounding. Specifically, they found mirror neurons in the area F5 of the monkey brain which are activated during the execution of certain actions and also during the passive observation (perceptual observation without any movements) of the same actions. For example, when a monkey eats a nut, certain neurons in area F5 fires. When the same monkey sees another monkey (or even a human) eating a nut, the same neurons in area F5 fires again. These findings show that mirror neurons take role both at action generation and understanding. This property of mirror neurons make them candidates as the underlying mechanism for understanding others’ actions and intentions. After the discovery of such mirror neurons, many researchers [2, 12, 21, 29, 32] believe that the mirror neurons serve for sharing the same grounding for perception action representations, enabling members of a species to communicate.

Nishitani et. al. [42] reports the presence of a close perception-action-language relation by pointing to the research on Broca’s region in the human brain. They state the difference of

their current understanding about the Broca's region from the early works as follows. In the early works, Broca's region was considered to be an exclusive speech-production area. However, recent findings show that the Broca's region contains representations of hand actions and mouth-face gestures. In other words, Broca's region takes role in both language and action generation. This shows us the close connections between actions and language.

Apart from these neuroscientific findings, computational and robotic studies try to create mechanisms for the solution of the symbol grounding problem. In their study, Marocco et. al. [35] indicate the grounding of language on actions and propose a system for the embodiment of action words. They demonstrate how the iCub humanoid robot platform can learn the meanings of action words by physically interacting with its environment. The system performs actions using trained Back-Propagation-Through-Time neural networks to interact and manipulate the objects in the environment. Then, the system links the effects of its own actions with the observed action on the objects before and after the action. This study stresses the formation of links between words and actions by a process based on the interaction of the agent with its environment (on the agent's sensori-motor experiences).

Steels and Kaplan [54] demonstrates the role of social learning on language acquisition. They explore the hypothesis that language communication is bootstrapped by social learning. The robot platform SONY AIBO plays simple social games with a human to correlate words with objects. During these social games, the human presents an object and speaks a word for AIBO to correlate. The results of the work shows that the category formation is bootstrapped (guided) by the social learning. In this thesis, we will facilitate such social learning in a slightly different form. In our setup, the learner will perform some behaviors on the objects in the environment and a teacher will provide words (labels) for the effects.

Roy et. al. [50] proposes a computational model of word acquisition which can learn from multi-modal sensory input. The CELL (Cross-channel Early Lexical Learning) system uses utterances of an infant with their corresponding video images of simple objects to discover words and categorize objects. The system can successfully acquire words from the raw sensor data without the need for a human transcription or labeling. However, this work does not allow the observer to interact with the environment, the learning is passive. The proposed system in this thesis uses the learner as an active participant in the formation process of object concepts.

Cangelosi et. al. [6] proposes an embodied model for grounding of language on actions.

The work investigates the generation of grounding and grounding transfer between multiple robots. The word grounding achieved as follows; each robot learns to execute eight basic actions by observing another robot and imitating its movements. Meanwhile, the corresponding words to each action is presented to the input of the learner’s neural system. Since the words are learned together with the actions, they become grounded on actions. The grounding transfer is achieved by learning high-level actions using a human’s description of the high-level action in terms of the previously learned eight basic actions. The ability to learn new high-level actions based on ground level actions is an important ability for any embodied system. Similarly in this thesis, we will show that how our acquired object concepts enable us to learn affordances better.

In this thesis, we propose a computational perception-action system which forms appearance and function-based object concepts from an agent’s sensori-motor experiences. The proposed system differs from the literature with its property of using both appearance and functionality of objects to acquire object concepts. Different from the passive observant systems in the literature, the proposed system actively interacts with the objects in the environment to gather knowledge about the objects in order to use them for concept acquisition. We further demonstrate how these object concepts can be used to learn the affordances of an agent in the environment. Up to our knowledge, the demonstration of how Multi-Task Learning can be used for learning affordances is the first in the literature.

2.2 Affordances

The term “affordance” is first coined by J. J. Gibson [23], an American psychologist, to refer to “*action possibilities presented to an actor by its environment*”. Affordances depend on both the actor and the environment, they are neither a property of the actor nor the property of the environment. Affordances encapsulate both the actor and the environment and represent their relation in terms of actions. Depending on the actor, objects in the environment may provide different affordances. A pebble, for example, is throwable for a human but is not throwable for an insect. On the other hand, the same pebble affords hiding to an insect whereas the hiding affordance by the pebble is not valid for a human.

One of the important properties of the affordances is that they are detected by online sensori-

motor information (e.g. visual, tactile) and does not require recognition. For example, in order to detect the sittability affordance a chair provide us, we do not have to first recognize the chair and then understand it provides sittability. We perceive the flat surface of a chair and understand that it provides sittability to us. This property of affordances makes them a good tool for investigation of perception-action-language relations by helping us to understand the formation of sensori-motor groundings for language.

Representation of affordances is an important issue. There are different formalizations for representing affordances in the literature (e.g. [11, 53, 55, 58]). Turvey [58] defined an affordance as a *potential* of a thing. Such potentials are activated or become realized when they are combined with their complements (e.g. an actor). This formalization explicitly attaches affordances to the environment instead of defining them as a relation between an agent and the environment.

Stoffregen [55] argues the formalization of Turvey and claims that affordances are *properties of the agent-environment system*. In this view, instead of being properties of an actor or the environment, affordances are emergent properties of the agent-environment system.

Chemero [11] proposed a definition for affordances which is similar to the definition of Stoffregen. While Stoffregen defined the affordances as the properties of the actor-environment system, Chemero defined them as the *relations between the abilities of actors and features of the environment*. For example, if an agent can lift weights smaller than 10 kg, then an object which is 5 kg provides liftability to this agent.

Steedman [53] has a view of affordances different than the three views presented above. In this view, an agent creates *object schemas* which define the set of affordances an object provides to the agent. Object schemas are defined with events and actions. For example, Steedman suggests that a door is linked with actions of *pushing* and *going-through* and the pre and post-conditions of applying these actions to the door. In this example, the object schema of the door has an affordance-set consisting pushability and traversability. Such set of affordances for any object schema can be extended via learning. This formalization is suitable for planning where Steedman argues that reactive/forward-chaining planning is the best candidate.

In Şahin et. al. [51], an affordance is represented by a relation between an entity (*e*), a behavior (*b*) and an effect (*f*) which can be shown with a relation:

$$a = (e, b, f). \quad (2.1)$$

An entity is the sensori-motor perception of an object in the environment by an actor. This perception can be visual or multi-modal (e.g. visual, tactile, auditory). A behavior is an action the actor can perform, for example pushing or pulling. An effect is the observable outcome of applying one of actor's behaviors on an entity. For example, when we squeeze an egg our sensori-motor perception of egg will change, this change in the perception makes our effect. As we can see, an affordance relation includes both the actor (behavior) and the environment (entity) via their relation (effect).

When we analyze the possible affordance relations for actor-environment couples, we can see that there is a many-to-many relation between entities, behaviors and effects. Figure 2.1 shows these relations. An entity can have more than one behavior-effect couple simply because different behaviors yield different effects. Similarly, a behavior can be applied on many entities and may yield many effects. To clarify this many-to-many relations, let us analyze the affordance relations for an apple, egg and a glass (cup) with a human actor. Assume our actor can bite, push and squeeze. The full list of affordance relations we have is as follows:

- (*apple, bite, bitten*)
- (*apple, push, rolled*)
- (*apple, squeeze, no change*)
- (*egg, bite, bitten*)
- (*egg, push, rolled*)
- (*egg, squeeze, crushed*)
- (*glass, bite, no change*)
- (*glass, push, slided*)
- (*glass, squeeze, no change*)

Affordance relations are important for us because they enable us to make generalizations which leads to creation of object concepts. Assume we have a single affordance relation (*green-apple, bite, bitten*) which does not tells us much. Then we see a red-apple and apply our bite behavior to get a second affordance relation (*red-apple, bite, bitten*) which does not

tells us much either. However, we can combine this two affordance relations by finding the invariant properties of the entities and discarding the variant properties to get the generalized affordance relation ($\langle *-\text{apple} \rangle$, *bite*, *bitten*). $\langle *-\text{apple} \rangle$ in the affordance relation denotes the invariant properties of entities which have the (bite,bitten) behavior-effect couple in their affordance relations¹. This affordance relation tells us much more than the two separate relations we had before. Now we know that regardless the color we can bite an apple and get the effect bitten. This generalization enables us to make predictions. For example, when we see a yellow-apple we can predict it can be bitten by using the affordance relation ($\langle *-\text{apple} \rangle$, *bite*, *bitten*).

2.2.1 Literature on Usage of Affordances

Affordances are used in the literature for different purposes. Montesano et. al. [40] uses affordances for imitation. A robot first learns affordances by interacting with the environment. The benefits of learning such affordances are demonstrated via simple imitation games between a human and a robot. First, the human applies a behavior on an object and the robot observes the created effect. Then, the human presents different objects for the robot to choose one and interact with. The robot selects the correct object and the behavior to perform to get the same effect as the human created.

Doğar et. al. [15] uses affordances to create goal-directed actions from primitive behaviors (turn left, turn right etc.). In the concept of affordances, a robot first learns what kind of effects it can create and links these effects with the perceptual properties of the environment. Then, the robot is asked to perform one of the goal-directed actions (avoid, approach or traverse) in an environment with different kinds of objects and obstacles. The robot performs successfully by using learned affordances to predict the effects of its actions and match them with the desired goal-directed action.

In this thesis, we will use our affordance relations to acquire different object concepts. The proposed system will interact with the objects in the environment to learn affordance relations and then find the invariant properties of entities which will result as object concepts. The acquired object concepts will further be used for learning new affordances.

¹ In the general form, ($\langle e \rangle$, b, f) denotes the invariant properties of all *e* entities (an equivalence class) which has the b,f behavior-effect couple.

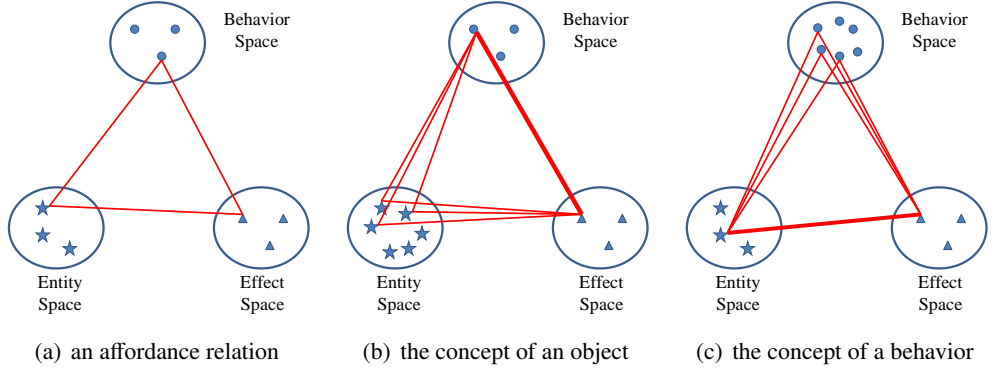


Figure 2.1: Relations of entities, behaviors and effects in our affordance formalization

2.3 Object Concepts

E. J. Gibson [22] claims that learning affordances leads to discovering features and invariant properties of objects. We argue that such invariant object properties correspond to “object concepts”. Object concepts can refer to perceptually and/or functionally different sets of features which are also known as appearance-based categorization and function-based categorization respectively. Most of the object categorization works in the computer vision area falls into the appearance-based categorization. Function-based categorization categorizes objects according to their functional capabilities (object categorization using affordances by Dağ et al. [14]). There are studies using appearance-based or function-based categorization but as Borghi et al. [4] states “adult humans perform both appearance and function-based categorization of objects”. In this thesis, our understanding of “object concepts” depend on both appearance and functionality of objects. Hence, the proposed system in this thesis uses both appearance and function of objects to create object concepts.

When we use a certain word in our communication, we make the assumption that this word has the same meaning for the listener. If a word does not correspond to the same meaning for both the speaker and the listener, then a healthy communication is not possible. This is where the significance of *concepts* are understood. Concepts (e.g. concepts of verbs, nouns etc.) provide grounding for the meanings of words in any language. Psychologists define the term “concept” as the knowledge we associate with the referent of the concept and what we know about the object [5]. For example, the concept of *heavy* includes our knowledge about it. When someone says: “The ball is *heavy*.”, we can understand what *heavy* means by using

our *heavy* concept. We interacted with many objects during our infancy and discovered that we cannot lift some objects. We later learn that things we cannot lift are called heavy so we link our past experiences which we conceptualized with the word “heavy”. These discoveries yield to the emergence of object concepts, in this case the *heavy* concept. Hence, the word *heavy* becomes grounded on our object concepts.

Object concepts can be formalized as $(\langle e \rangle, b, f)$, where $\langle e \rangle$ denotes the invariant properties of entities which give the same effect f when we apply the same behavior b on them, with our affordance formalization as shown in equation 2.1. Figure 2.1b shows the graphical representation of an object concept. It is the set of all entities which share the same behavior-effect pair in their affordance relation. For example, the object concept of *heavy* can be described as $(\langle * \text{-heavy} \rangle, \text{lift}, \text{not-affected})$ which can be derived from three affordance relations; (green heavy ball, lift, not-affected), (blue heavy sphere, lift, not-affected) and (orange heavy box, lift, not-affected). This formalization enables us to easily create object concepts out of affordance relations.

2.3.1 Literature on Object Concepts

The creation of object concepts through interactions with the environment gained interest in the literature. Nolfi and Marocco [43] uses tactile-sensing in order to categorize objects by using a robot arm with tactile sensing ability. The robot arm used is a three segment robot arm with crude touch sensors at each segment. They present objects to the robot arm to interact with. After the evolutionary learning process, the robot arm finds the correct movements to interact with the presented object and understand its category by using the touch sensors’ responses.

In Dağ et. al. [14], a robot first discovers the affordances of different objects and use these affordances to categorize objects. They first create effect prototypes by finding the significant changes on the features of objects (e.g. increase in the x position due to push left behavior). Then objects are categorized by their set of effect prototypes. They also demonstrate the generalization ability of their system by presenting novel objects to the system which system never interacted before and calculate the object category. Effect prototypes in this work can be considered as verb concepts since each effect prototype consists generalized information about affordances of objects.

In Sun et al. [56] work, a room cleaning robot categorizes objects by affordances to successfully clean a floor. Their proposed system is called Category-Affordance (CA) where robot first discovers the categories of objects then it learns the affordances of objects through their categories. They argue that the *direct perception* of affordances limits systems' learning and generalization abilities and by using object categories to perceive affordance they follow an *indirect perception* approach. Similarly, Uğur et al. [59] shows the usage of affordances to detect the traversability affordance in the environment by a mobile robot.

2.4 Multi-Task Learning

Multi-Task Learning is an approach which uses similarities between multiple tasks in order to improve the performance of a learning system. Many empirical studies show that learning related tasks *together* gives better results than learning them separately [3, 8, 9, 10, 17, 27, 57]. This is done by transferring knowledge between tasks while learning multiple tasks in parallel. Learning similar tasks together enables the system to use learning signals of one task to be used for another task which shares a similarity. These properties of Multi-Task Learning approach makes it better than Single-Task Learning approaches where every task is learned independently. For the sake of clarity, we will refer to the Multi-Task Learning and Single-Task Learning approaches as MTL and STL from now on.

MTL has proven to be a promising learning approach. Caruana is one of the first to point to the benefits of the MTL [8, 9, 10]. In his works, he demonstrates the benefits of the MTL over the STL by comparing their learning performances on simulated and real-world problems using backpropagation neural networks. He lists five mechanisms to achieve MTL which depend on different principles. Here we will explain one of them in detail, *representation bias*.

To clarify the working principle of the MTL, let us use an example. One of the simplest ways to achieve MTL is using neural networks with backpropagation. Assume we have four similar tasks which give a single output given an input vector. Figure 2.2 shows the four neural networks which we use to learn four tasks independently. Since all of the networks are independent from one another, there is no possibility for sharing information. Even if all four tasks are exactly the same, we have no choice but to learn all of them from scratch.

Figure 2.3 shows the MTL approach. Since all tasks work on the same input, we can combine

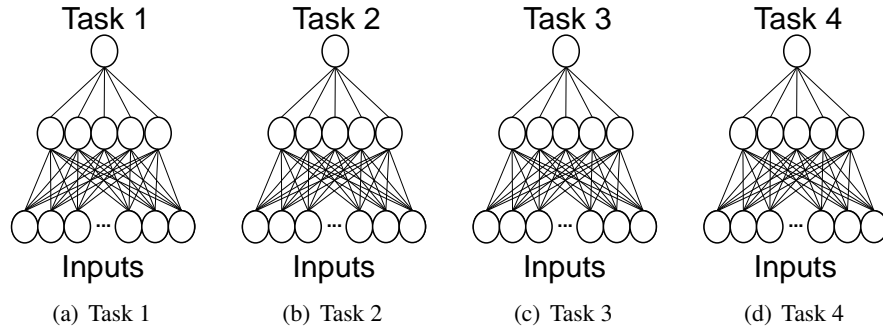


Figure 2.2: Learning four tasks as independent neural networks

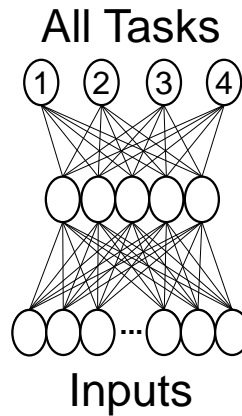


Figure 2.3: Learning four tasks in a single neural network

them as the output layer of our new neural network. This way we have the same input layer and all our tasks in the output layer. Notice that all of the tasks are connected to the same hidden layer. Since the hidden layer is serving to all of our tasks the learning process in the hidden layer will be guided by all of the tasks. If our tasks are similar, then their combined learning signals (calculated by the learning function of the neural network) will boost the learning process. Furthermore, the weights in the hidden layer will form a shared grounding for all tasks. If we add another similar task to the output layer, the learning process will not start from scratch. On the contrary, the already learned weights in the hidden layer will serve as a grounding for the new task and this will provide a benefit in terms of learning performance.

CHAPTER 3

Methods and Experimental Setup

In this chapter, we first describe our experimental framework then, we introduce our methods for creation of object concepts and their further usage in Multi-Task Learning.

3.1 Range Camera

During recent years, usage of range information of the scene gained popularity among researchers. We know that human brain and many other animal brains extract depth information from 2D visual input from eyes by using depth cues (including stereo vision). There are large number of studies trying to develop artificial vision systems [20, 28, 36, 45]; however, such artificial vision systems are far from extracting reliable depth information in all environments.

Range cameras, on the other hand, do not use stereo vision and directly perceive the environment and create a range image which contains the depth information of the scene. This property of range cameras makes them desirable for researchers whose main interest is not the process of depth information but its further usage. Although, range sensors were not so precise and affordable in the past decades. They are more available and affordable due to the many developments on range sensors (time-of flight sensors).

There are multiple types of range sensors which differ according to their measuring principle and used signal type. For example, laser range finders use a laser beam to measure the distance of a single point by measuring the required time of flight in order to laser to go to the measurement point and come back to the camera. Multiplying the time-of flight by the speed of the laser beam gives the distance of that point. Measuring multiple points around the scene (e.g a window of scene) gives a range image. Other range finders use infrared light or sonar

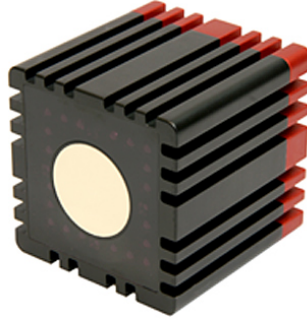


Figure 3.1: SwissRange SR4000 Range Camera

sound to measure the distance and get a range image of the environment. There are works which use range sensors as the main sensor of a robot in order to navigate, map or segment the environment around the robot [30, 34, 62].

The range camera we used in our experiments is SwissRanger SR4000 infrared range finder¹ which can be seen in figure 3.1. The camera can capture range images with a resolution of 176×144 at 30 frames per second. The camera provides three kinds of images; range, amplitude and confidence images. Range image contains the depth information of the scene. Amplitude image contains the returning signal strength which tells us how much reflective each point in the image (like a greyscale image). Confidence image describes the amount of certainty about the measurement of a point. High confidence corresponds to static parts of the scene whereas low confidence might correspond to dynamic or problematic (multiple path, glass, overexposure etc.) parts of the environment.

3.2 Behaviors

We apply five simple behaviors to the objects in our experiments. Our behaviors are, *push-left*, *push-right*, *grasp*, *lift* and *rotate-45 degrees*. Push-left and push-right behaviors are relevant in terms of their effects on objects. If an object is rollable, it will be rolled by both push-left and push-right behaviors, the only difference is the direction of the movement. Similarly, grasp and lift behaviors are relevant in terms of their effects. If an object is graspable, then it is also liftable (because grasping precedes lifting and none of the objects are too heavy). Rotate behavior does not have any similarity with other behaviors and is added to the behavior

¹Website of the product: "<http://www.mesa-imaging.ch/prodview4k.php>"

repertoire for this reason.

Figure 3.2 illustrates how our behaviors are executed by a human.

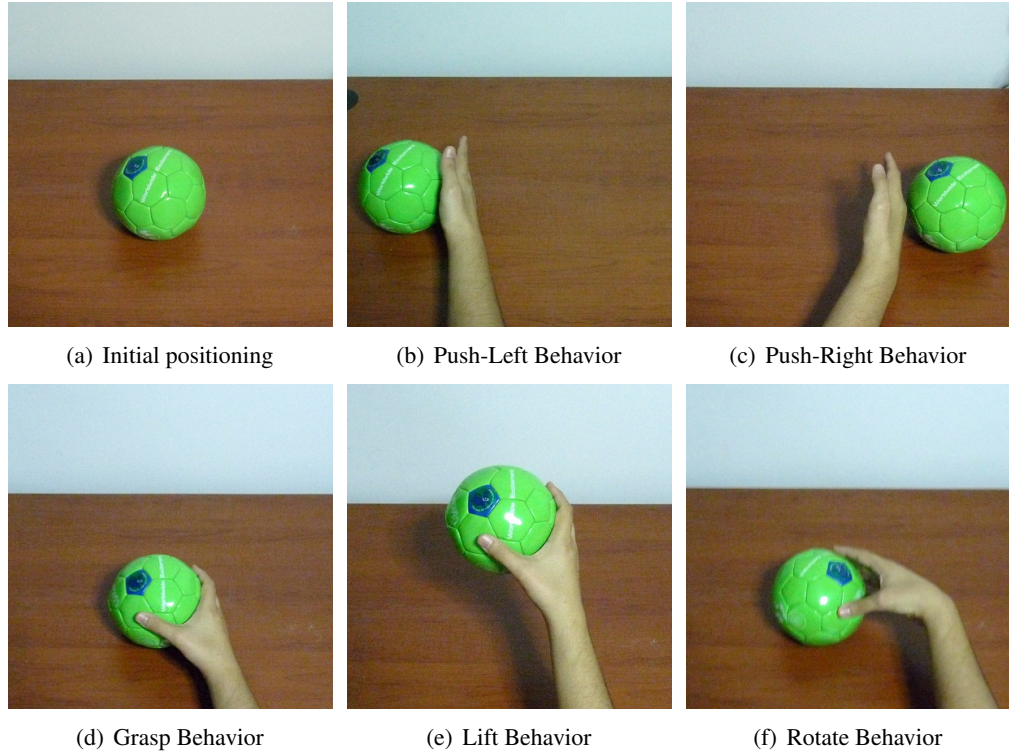


Figure 3.2: Behaviors used for interacting with objects

3.3 Data

We used 9 different objects composed of three different kinds of shapes and three different sizes which are shown in figure 3.3. We have boxes, spheres and cylinders where all have roughly three different sizes; small, medium and big.

We capture the data at two points of the execution of a behavior, one before the execution and one after the execution. We name the captured features before the execution as *initial features*. Initial features also corresponds to the initial state of the *entities*. Features captured after the execution are called *final features* since they represent the final situation. In order to get our *effects*, we take the difference between final and initial features. Hence, our effects will represent the change on the features of objects.

We apply five simple behaviors which are described in section 3.2 on each object five times

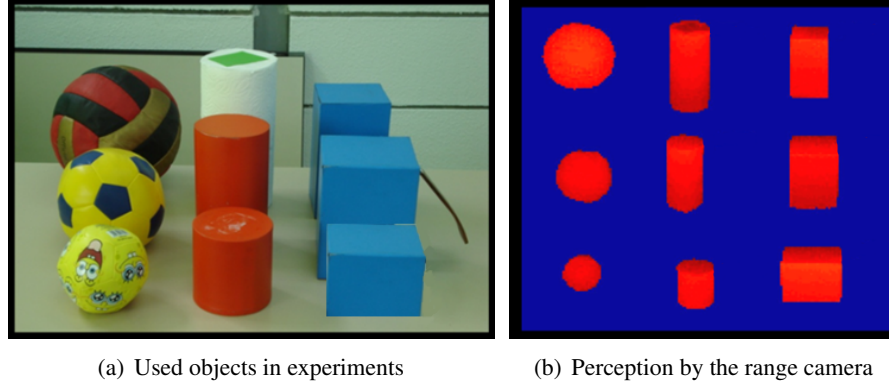


Figure 3.3: Used objects in the experiments. Object set contains three different shapes; spheres, cubes and cylinder with three different sizes; small, medium and big.

with slightly different initial positioning of objects. In total, we have $5 \text{ behaviors} \times 9 \text{ objects} \times 5 \text{ repetitions} = 225 \text{ samples}$.

3.4 Feature Extraction

Feature extraction from captured raw experimental data is an important step for the system. Extracted features refine the raw data and depending on the selected features, it elaborates some properties of the environment. The proposed system assumes that any affordance to be learned is perceivable through the available features. Thus, our features should be suitable to perceive affordances of the object in the environment. Also, depending on the goal, some features can be more useful than others. For example, Haar-like features are widely used for object and face detection due to their good-performance on encoding a scene, especially faces and objects [39, 61].

In our system, the features are extracted only from the object in front of the camera. The data from the camera consists of the complete perception of the environment besides the object. Thus, we first segment the scene into two; the object and the background.

For background segmentation, we apply thresholding on the amplitude image obtained from the range camera. We constructed our background as a low reflective scene to get a difference in amplitude between the object and the background. Then we apply amplitude thresholding on the average amplitude value difference to get the object only.

As summarized in [13], there are a variety of features that can be extracted from range data.

In this thesis, we utilize the following features whose extraction methods were developed by Dağ N. in [13]:

- 3D position of the object as X, Y and Z coordinates relative to the range camera.
- 10 features for the shape of the object. These 10 features are formed as a 10-bin histogram of the shape indexes as shown in the figure 3.4. Details of the shape indexes are described in the work by Koenderink and van Doorn [31].
- 3D orientation of the object in discretized form. The direction along which the object is longest is considered to be the orientation of the object. A Support Vector Machine classifier is trained to categorize the orientation of the object into one of 8 directions.
- 3 features for the size of the object in three axes. Object size at each orientation is measured from one extrema to another (e.g. in x axis, distance from the leftmost part to the rightmost part of object).

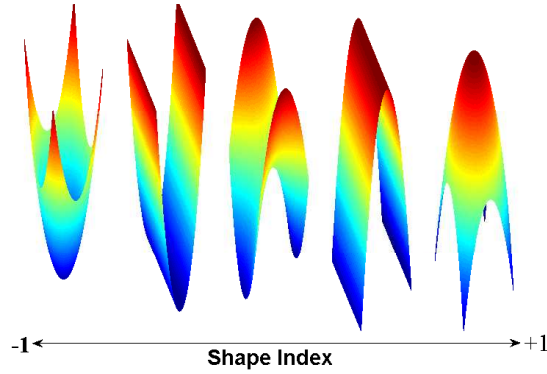


Figure 3.4: Visual description of the shape indexes. This figure is taken from [13] by the courtesy of Dağ N.

There are in total 17 features, which have been used in the rest of the thesis for the experiments.

3.5 Feature Selection

We use ReliefF [33] for feature selection. The algorithm calculates a weight for each of the given features according to a class label as described in Algorithm 1. The features which are

relevant with the class label take high scores whereas the features which are not relevant with the class label take low scores.

Algorithm 1 Pseudo code for ReliefF Algorithm

Each training instance is described by a vector of attributes and a class label

A: number of features

m: number of instances

k: number of nearest neighbours

set all weights[A] = 0;

for $i = 1$ to m **do**

 Randomly select an instance R;

 find k nearest hits H with the same class;

 find k nearest miss M(C) from each of other classes where $C \neq \text{class}(R)$;

for $a = 1$ to A **do**

$$\begin{aligned} \text{weights}[a] = & \text{weights}[a] - \sum_{j=1}^k \text{distance}(R, H_j, a) / (m \times k) \\ & + \sum_{C \neq \text{class}(R)} [\sum_{j=1}^k \text{distance}(R, M_j(C), a)] \end{aligned}$$

end for

end for

3.6 Unsupervised Clustering: Robust Growing Neural Gas

In order to select an unsupervised clustering method, we first determined the properties our method should have. We list our expectations from an unsupervised clustering method as follows:

- Ability to find natural clusters in the data
- Ability to find the correct number of clusters automatically (no under or over partitioning of data)
- Not affected by ordering of the data
- Not affected by noise and outliers in the data

After the literature survey, we found two candidates for unsupervised clustering; x-Means [44] and Robust Growing Neural Gas (RGNG) [46]. x-Means is able to determine the opti-

mal number of clusters automatically. Similarly, RGNG uses Minimum Description Length (MDL) criteria to find the optimal number of clusters for the given data. RGNG also has the ability to form clusters incrementally, real-time and able to track changing distributions over time. Furthermore, RGNG is able to capture the topology of the clusters with its dynamic neighbourhood relations. RGNG is not affected by the ordering of the data and robust against noise and outliers. x-Means does not have any of these properties; so, we decided to use RGNG as our unsupervised clustering method. The authors of the RGNG compare their work with x-Means and G-Means algorithms:

“The x-Means algorithm [44] and G-Means algorithm [25] are two famous representatives. However, the performance of most of these growing approaches may deteriorate significantly when data sets are contaminated by several outliers. Further, even if the actual number of clusters is detected, the obtained positions of corresponding cluster centers will be deviated significantly from the actual positions due to outliers.”

The Robust Growing Neural Gas algorithm is an incremental self-organizing network with a dynamic topological structure. The self-organizing network consists nodes where each node has a vector which holds the position of the node in the training data space. The RGNG performs a competitive learning between nodes to find the closest node each training data point. At the end of training, the nodes represent the center position of the clusters in the data and topological connections between these clusters.

The RGNG algorithm is the improved version of the Growing Neural Gas (GNG) [19] algorithm. The Growing Neural Gas algorithm itself is an improved version of the the Neural Gas algorithm by Martinetz et al [37]. While the neural gas and the growing neural gas algorithms are suitable to learn multi-dimensional data distributions successfully, the RGNG can be used to cluster multi-dimensional data due to its ability to determine the optimal number of clusters by using MDL criteria.

Algorithm 2 shows the pseudo-code for RGNG. The RGNG algorithm typically starts with two nodes. After a predetermined number of iterations over the training set a new node has been inserted between the node with the highest error rate and its neighbour with the highest error rate among all of the neighbours. Each node has a vector which determines its position

in the n -dimensional input space (where n is the dimensionality of the training data). For every instance of the training data, the algorithm finds the closest node (e.g. winner node) and updates its vector. The algorithm also updates the vector of the closest neighbour of the winner node (e.g. second winner).

Algorithm 2 Pseudo code for Robust Growing Neural Gas Algorithm

```

max c: max number of nodes

max e: number of epochs

m: number of instances

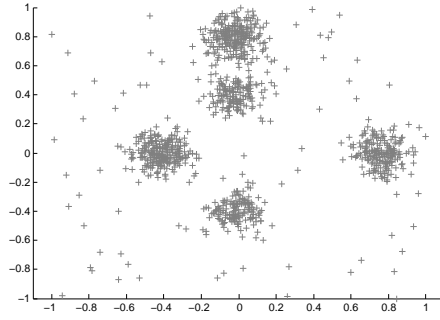
for node count = 2 to max c do
    for epoch = 1 to max e do
        for instance  $i=1$  to  $m$  do
            Find the winner node for the  $i^{th}$  instance
            Update the position of the winner node and its neighbour
            Update the connections between the winner and its neighbour
        end for
    end for
end for

Add a new node between the node with the highest error and its neighbour

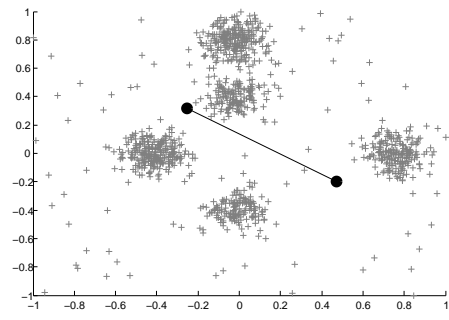
```

To demonstrate the working RGNG, we will use a 2 dimensional synthetic data where there are five natural clusters with noise and some outliers. Figure 3.5(a) shows the synthetic data distribution. We can see the positions of the nodes of RGNG as the number of nodes increase in the figure 3.5(b)-(h). We can see that after positioning five nodes, new nodes cannot find stable positions for themselves. Figure 3.6 shows the calculated MDL values for each number of nodes. After finding the minimum value of MDL, the RGNG determines there are five clusters in the data. We also run x-Means on the same data for comparison. x-Means algorithm finds four clusters in the data. We present the final cluster positions of the x-Means and RGNG algorithms in the figure 3.7. It is clear that the clustering of the RGNG is better than the xMeans since we know there are five clusters in the synthetic data and RGNG gives the correct result.

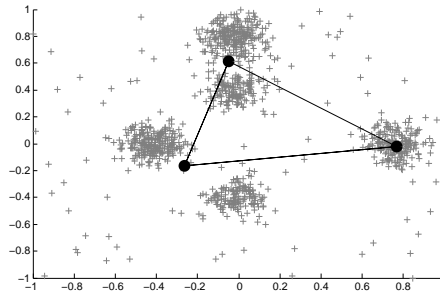
Topology management of the RGNG is done by adding or removing connections between nodes: some connections get removed if they are not reinforced for a long time (e.g. aged enough) and other connections are created according to the update rules. See [46] for details.



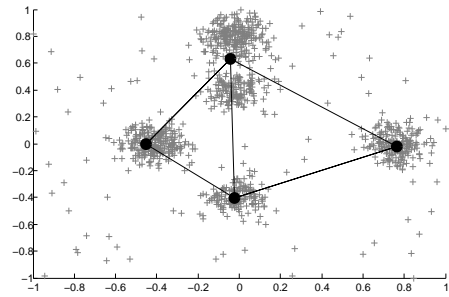
(a) Synthetic Data Distribution



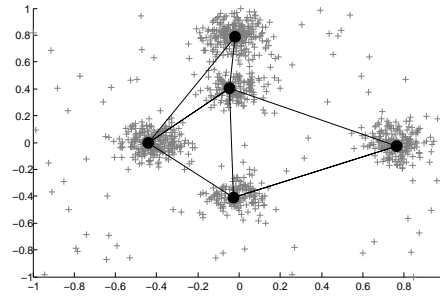
(b) RGNG with 2 clusters



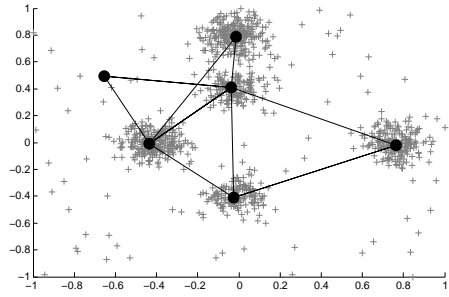
(c) RGNG with 3 clusters



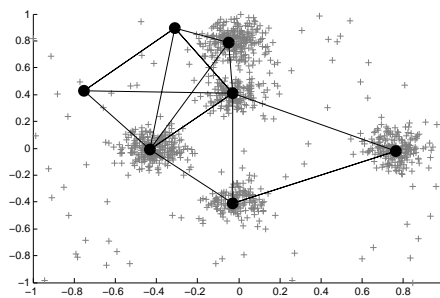
(d) RGNG with 4 clusters



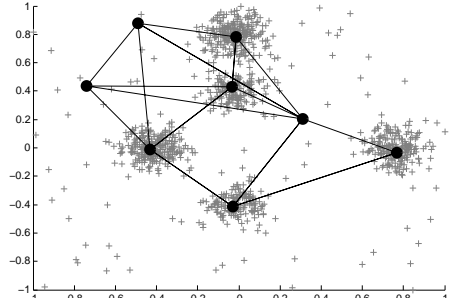
(e) RGNG with 5 clusters



(f) RGNG with 6 clusters



(g) RGNG with 7 clusters



(h) RGNG with 8 clusters

Figure 3.5: Steps of the RGNG

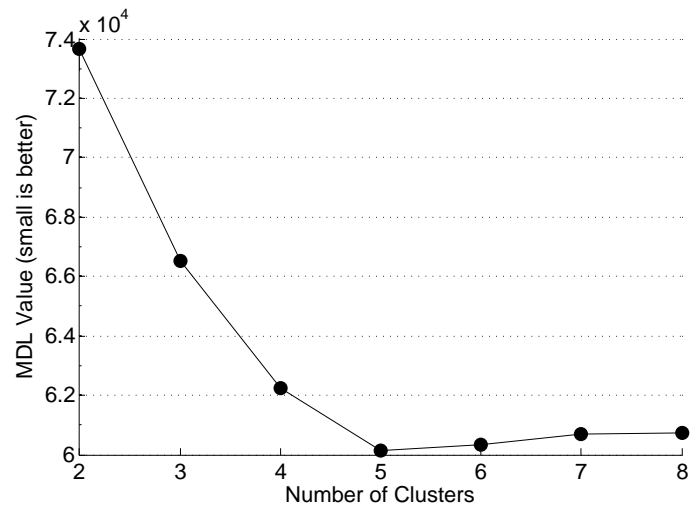


Figure 3.6: Minimum Description Length values for different number of clusters

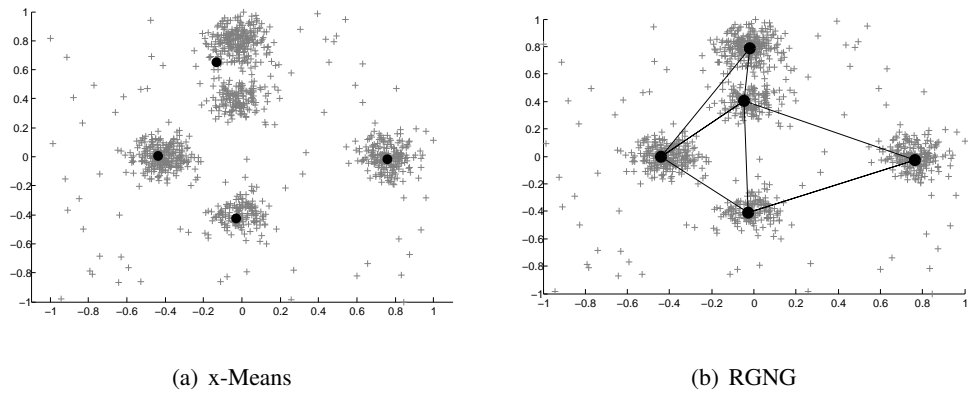


Figure 3.7: Comparison of x-Means and RGNG clustering results

3.7 The System Proposed in This Thesis

The proposed system has three main steps. The graphical representations of these steps are shown in Figure 3.8:

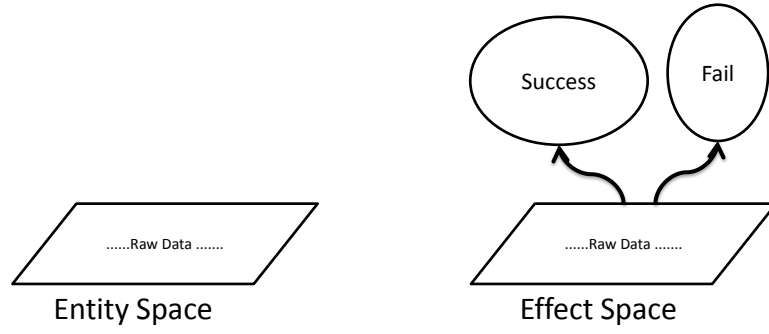
- Obtaining Effect Clusters: *Learning the Separation*
- Feature Selection: *Analysis of the Entity Space*
- Creation of Object Concepts: *Unsupervised Clustering of the Entity Space*

These three steps enable us to use both the appearances of objects and their affordances to create our object concepts. Object concepts are created by clustering the entity space as can be seen in Figure 3.8(c). The entity space contains the perception of objects before any behavior applied to them; hence, this space mainly represents the appearance of an object. If we cluster the entity space without feature selection, then the object concepts will only depend on the appearance. This does not satisfy our goal of using both appearances and affordances for creation of object concepts. The entity space also contains invariant features of affordances; so, we must find a way to select them in order to make our concepts depend on both appearance and affordances. In order to do this, we must first find what our affordances depend on.

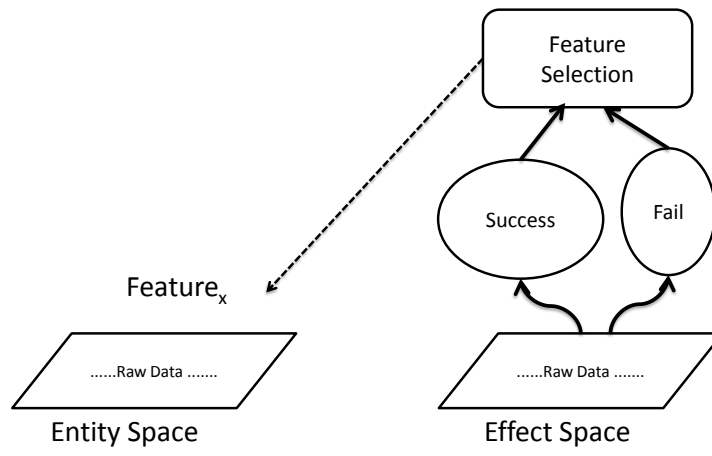
3.7.1 Obtaining Effect Clusters

Whether an object provides a certain affordance or not can be determined by looking at the effect space. The effect space consists the results of applying behaviors to presented objects. If an object has liftability affordance and we applied lift behavior on it, then the result must be a success (*lifted*). Similarly, the effect on an object which does not have liftability affordance will be a failure (*no-change*). Figure 3.9 shows the effects of all behaviors on all object types.

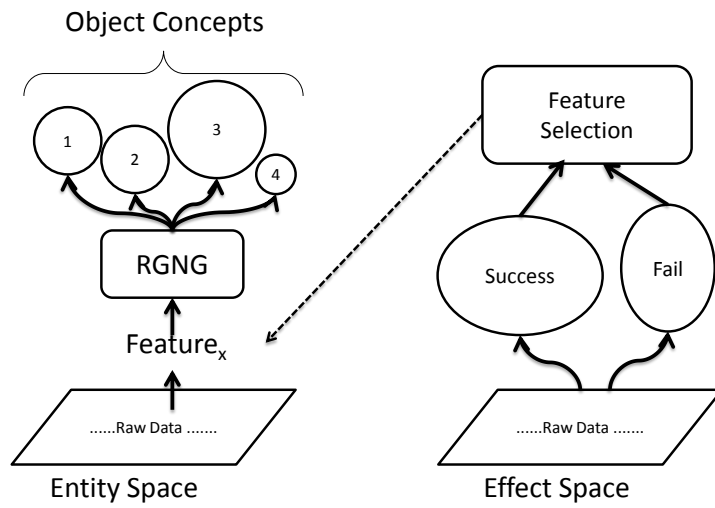
Separation of effect space into two as success and fail is performed by a human. We can think of this process as a teacher telling us what is considered as a success and what is considered as a failure. In this way, the teacher transfers its understanding of success and fail (grounding for their meanings) to us which enable us to create a similar grounding with the teacher. In the long run, it enables us to communicate on the same meaning ground.



(a) Obtaining Effect Clusters



(b) Feature Selection



(c) Creation of Object Concepts

Figure 3.8: The three steps of Our Proposed System










Effects of behaviors on different objects									
									
Push	Roll	Roll	Roll	Drag	Drag	Drag	Drag	Drag	Drag
Lift	Lift	Lift	Stay	Lift	Lift	Stay	Lift	Lift	Stay
Grasp	Able	Able	Not	Able	Able	Not	Able	Able	Not
Rotate	Same	Same	Same	Differ	Differ	Differ	Same	Same	Same

Figure 3.9: Effects of behaviors on all object types

At the end of this step, our system looks like Figure 3.8a.

3.7.2 Feature Selection: Analysis of the Entity Space

We have the success and fail clusters for each behavior. The next step is to find what causes the effect of a behavior to result as a success or a failure. As we mentioned before, entity space contains information about an object before a behavior is applied. If we analyze the entity space, we can find an invariant feature which is shared among objects which fall to success and objects which fall to fail effect clusters.

As can be seen in the Figure 3.9, object types which result as success indeed share some invariant property. In the case of push behaviors (both left and right), round objects are labeled as success (rolled) where cornered objects are labeled as failure (dragged). Thus, we can perform a feature selection to find which feature best describe this separation for each behavior in the entity space.

At the end of this step, our system looks like Figure 3.8b.

3.7.3 Creation of Object Concepts: Unsupervised Clustering of the Entity Space

Now, we have a single feature which best separates the success-fail groups. Before continuing to learning part, let us further analyze the previous steps. We first formed groups in the effect space in a supervised way. Then, we analyzed the entity space to find the best feature

that captures the success-fail separation. By doing so, we transfer information from the effect space to the entity space which satisfies our goal of learning affordances based on both appearances and affordances (appearance information is already present in the entity space, affordance information is transferred via feature selection according to the supervised effect groups).

Our next step is unsupervised clustering of the entity space over the best feature which resulted from the feature selection. We apply Robust Growing Neural Gas (RGNG) algorithm to create clusters. The number of clusters is determined by RGNG automatically. This step completes the creation of high-level object concepts which are based on both functions and appearances of objects. As we stated before, we could have performed unsupervised clustering on the entity space without performing the previous steps but that would not satisfy our goal ¹. The previous steps created a clustering in the entity space that is grounded on both the affordances and appearances. At the end of this step, our system looks like Figure 3.8c.

In order to justify the necessity of the steps of our system, we would like to take a reverse approach to the ordering of the steps. Assume we have the same set of objects. When we try to cluster our objects, we face a decision problem; “what property of objects our clustering will depend on?”. It is known that in unsupervised clustering, the set of features play a crucial role on the resulting clusters. Since clustering gives equal weight to all of the features given, a feature which can give us the desired clustering result can be easily suppressed by irrelevant features. Our system solves this by looking at the effect space. Let us say the affordance we want to learn is grasping. The condition of grasping an object is its size (assuming no handles are available). If the object is bigger than the hand of the actor (single hand grasp by a human or robot) then it is not graspable. The ideal clustering of the entity space should be grounded on the size property of the objects. Our system analyzes the success-fail grouping on the effect space and finds that the size feature is the best discriminative feature between the success-fail groups. Then the system performs the unsupervised clustering over the size feature. This, in the end, gives us object concepts which are based on both appearances and functions of our objects.

¹ We actually performed unsupervised clustering on the entity space without selecting any features and using the whole feature set out of curiosity. The system resulted with four clusters where the first cluster included small and medium sized balls, the second cluster included the big ball and the small cube, the third cluster included medium and big sized boxes and the final cluster included all cylinders. The reason for such clustering is that shape is a big part of our features.

3.7.4 Benefits of the Object Concepts

Creation of object concepts creates a grounding based on the functions and appearances of objects. This grounding can be used for many purposes, for example creation of higher-level concepts, language grounding, affordance learning, etc. In order to show the benefits of the object concepts, we use them in affordance learning. In this thesis, we consider learning an affordance as being able to predict the effect of a certain behavior given the entity. For example, when the system can successfully predict the effects of the lift behavior, we say that the system has learned the liftability affordance.

One of the ways of affordance learning is to learn a mapping from the entity space to the effect space over raw data or basic features which we can call direct perception approach as we have shown in Figure 1.1a. Although this approach may yield good results, using raw data makes the system sensitive to noises, low success rate against novel situations, incapability to deal with high number of affordances and most of all the system cannot use the knowledge of previously learned affordances for learning new ones. As Sun et. al. [56] stated:

“For each new affordance, the robot would need to acquire substantial additional training datasets and construct additional independent classifiers. This is the defining property of the DP (Direct Perception) approach, and a major barrier to scalability.”

Our object concepts eliminate the problems related with the direct perception approach. In order to show the benefits of the acquired object concepts, we perform Multi-Task Learning over our object concepts to learn affordances. As described in section 2.4, Multi-Task learning (MTL) facilitates inter-task similarities and transfers knowledge from one learned task to another in order to perform a better learning. Learning a new task for us corresponds to learning a new affordance. Knowledge transfer can greatly reduce the cost of learning a new affordance, for example if the system has already learned the push-left affordance, learning the push-right affordance becomes very easy due to high similarity between push-left and push-right affordances. Details of the MTL learning process has been described in the following section.

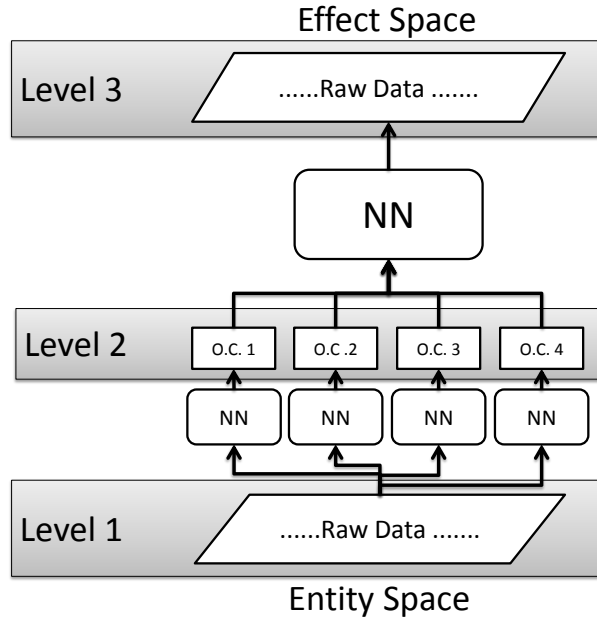


Figure 3.10: Multi-Task learning method

3.8 Performing Multi-Task Learning

From the concepts derived by the system, we construct the MTL network shown in Figure 3.10. The entity and the effect spaces consist features extracted from the raw perceptual data, which is captured by the range camera. We use feedforward neural networks with backpropagation for learning different steps of the MTL network. After our system creates object concepts, we prepare the training data for the MTL network. Training data of the MTL network consists three levels:

- Level 1: Entity Space Features
- Level 2: Object Concepts
- Level 3: Effect Space Features

First, the MTL network learns the mapping from entity space features to object concepts (from level 1 to level 2). This step is performed whenever new object concepts have been created (e.g. learning a new affordance) by the system. So, we have more than one learned

mapping from all entity features to object concepts. Having more than one mapping here is necessary because each newly created object concept depend on a different separation of the entity space.

Second, the MTL network learns the mapping from object concepts to the effect space (from level 2 to level 3). Contrary to the first part of the MTL network, this part is shared among all affordances in order to obtain the Multi-Task Learning property. The input of this part of the MTL network is all created object concepts and the output is all effect features of all learned affordances. This way, knowledge transfer between similar affordances is possible.

CHAPTER 4

Acquired Concepts and Multi-Task Learning

In this chapter, we present the acquired object concepts acquired by our system as described in the previous chapter. We further analyze the acquired object concepts by analyzing their contents and similarities between each other. We also show the benefit of our object concepts by using Multi-Task Learning (MTL) and comparing it with Single-Task Learning (STL).

4.1 Concepts

We acquire our concepts by applying the steps described in section 3.7. Object concepts are acquired for all affordances separately. We will now analyze the results of the system for all affordances.

4.1.1 Push-left and Push-right

Figure 4.1 illustrates the entity and effect spaces and created clusters for push-left and push-right affordances. On the left hand side, we can see the four clusters which correspond to our object concepts. On the right hand side, we can see the desired separation of the effect space. When we inspect the contents of the acquired concepts, we can see that one concept includes small and middle sized balls. Another concept includes only large balls. Another concept includes small sized cubes and small and middle sized cylinders and the last concept includes medium and large sized cubes and large cylinders.

The acquired concepts for the push-left and push-right affordances depend on the shape of the objects. The feature which encodes the roundness of the object surface, 10th bin of the

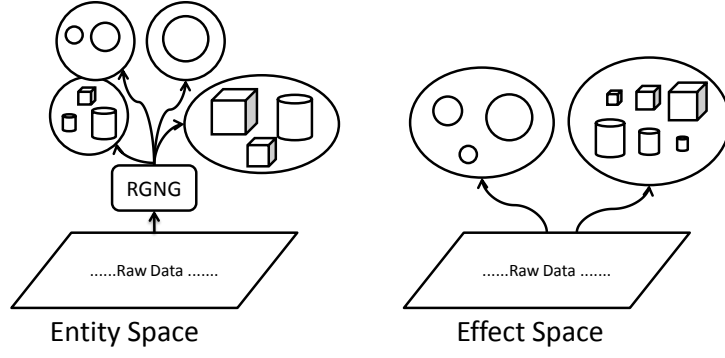


Figure 4.1: Object concepts for push-left and push-right

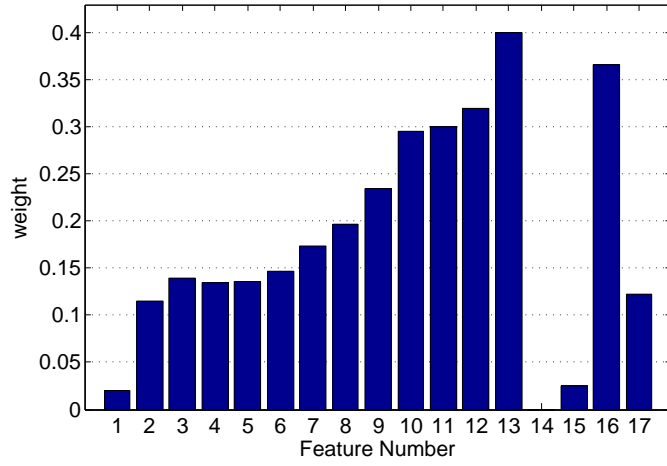


Figure 4.2: Weights of features for the push-left and push-right behaviors calculated by ReliefF

shape histogram, was discovered as the most relevant feature. Figure 4.2 shows the calculated weights of features by ReliefF algorithm. Unsupervised clustering of the entity space over the selected feature creates a separation which is based on object roundness. We can see that the desired separation on the effect space has been successfully captured by the system since none of the created clusters violate the desired separation. In other words, we can represent the separation on the effect space by combining acquired object concepts on the entity space.

We further analyze the acquired concepts by analyzing the distribution of our objects through the most relevant feature dimension. Figure 4.3 shows our objects' distribution through the

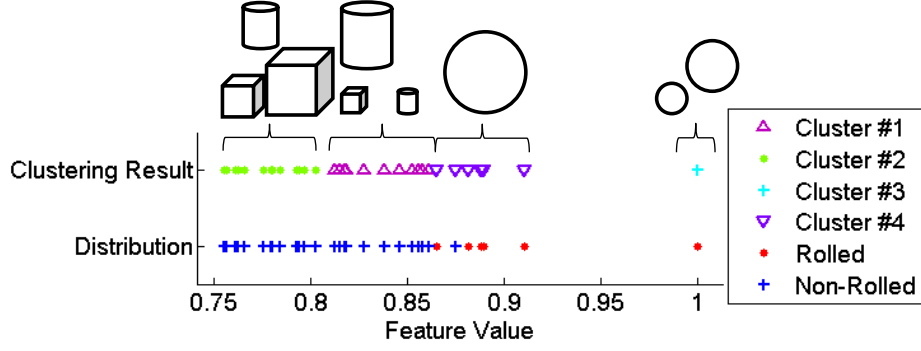


Figure 4.3: Detailed visualization of the acquired concepts for push-left and push-right

most relevant feature dimension. Round objects have higher values than the angular objects. The small and medium sized balls have the highest values and create a distant concept from others. The large ball is closer to the angular objects than to the small and medium sized balls according to the 9th bin of the shape index. Due to its large size, curvature on the surface of the large ball is smaller compared to the small and middle sized balls, which causes surface patches to look like flat. Since other angular objects has flat surfaces, the large ball is close to them. Still, our system does not mix the large ball with other angular objects and creates a cluster which contains only the large ball.

4.1.2 Grasp

Figure 4.4 illustrates the entity and effect spaces and created clusters. On the left hand side, we can see the four clusters which correspond to our object concepts. On the right hand side, we can see the desired separation on the effect space. When we inspect the contents of the acquired concepts we can see that one concept includes only small sized balls. Another concept includes small sized cubes and cylinders and middle sized balls. Another concept includes only large cylinders and the last one includes small and medium sized cubes, medium sized cylinders and large balls.

The acquired concepts for the grasp affordance depends on the size of the objects as we expected. The feature which encodes the height of the objects has been discovered as the most relevant feature. Figure 4.5 shows the calculated weights of features by ReliefF algorithm. Unsupervised clustering of the entity space over the selected feature creates a separa-

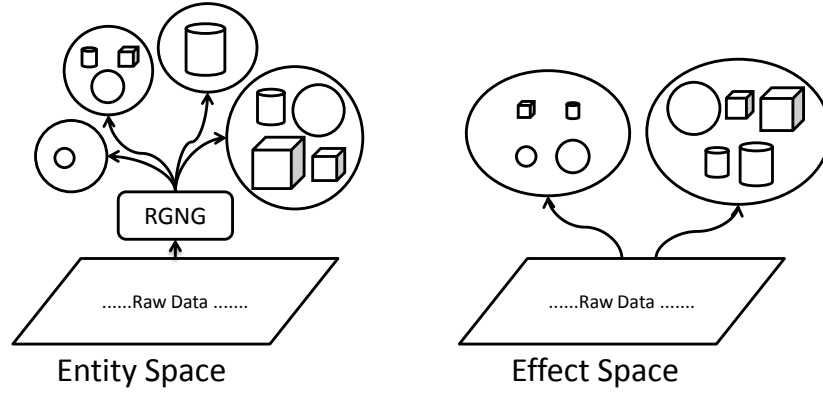


Figure 4.4: Object concepts for grasp

tion which is based on object size. Acquired object concepts capture the desired separation on the effect space. When we look at the contents of our concepts, we can see that graspable objects are represented by two concepts, which include small balls, cubes and cylinders and also medium sized balls, and non-graspable objects are represented by the other two concepts, which include medium sized cubes and cylinders and large sized balls, cubes and cylinders.

We further analyze the acquired concepts by analyzing the distribution of our objects through the most relevant feature dimension. Figure 4.6 shows our objects' distribution through the most relevant feature dimension. As we go from left to right along the feature dimension, the height of objects increases. The small ball, which is the smallest object among our object set, is placed at the leftmost of the distribution and creates a concept on its own. The small sized cube and cylinder with the medium sized ball create another concept since their heights are very close to each other. The large cylinder is positioned at the rightmost of the distribution since it is the tallest object and creates a concept on its own. Other objects which are medium sized cube and cylinder and large ball and cube create the final concept. One might expect to find three concepts with all small, middle and large sized objects, respectively. However, the actual sizes of our objects creates a distribution based on height as in figure 4.6, where we see the formation of four clusters.

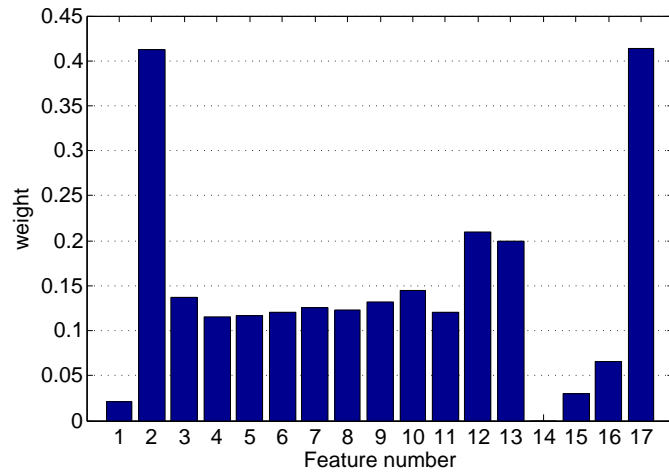


Figure 4.5: Weights of features for the grasp behavior calculated by ReliefF

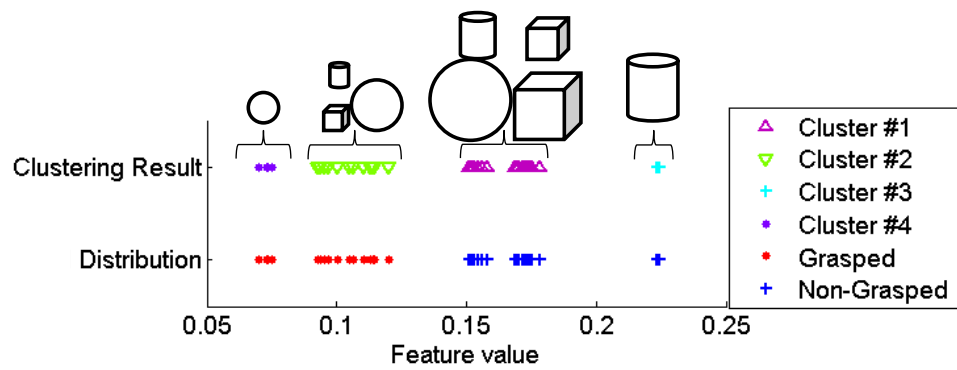


Figure 4.6: Detailed visualization of the acquired concepts for grasp

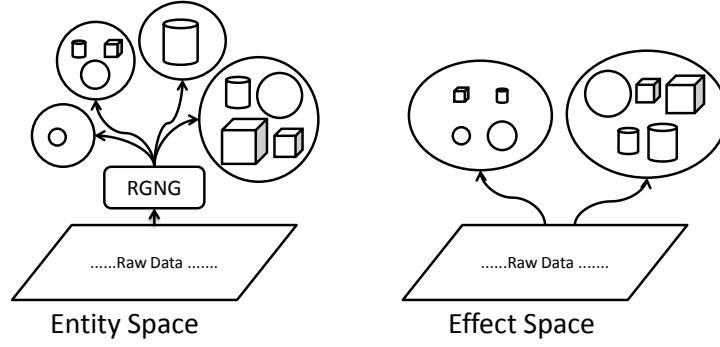


Figure 4.7: Object concepts for lift

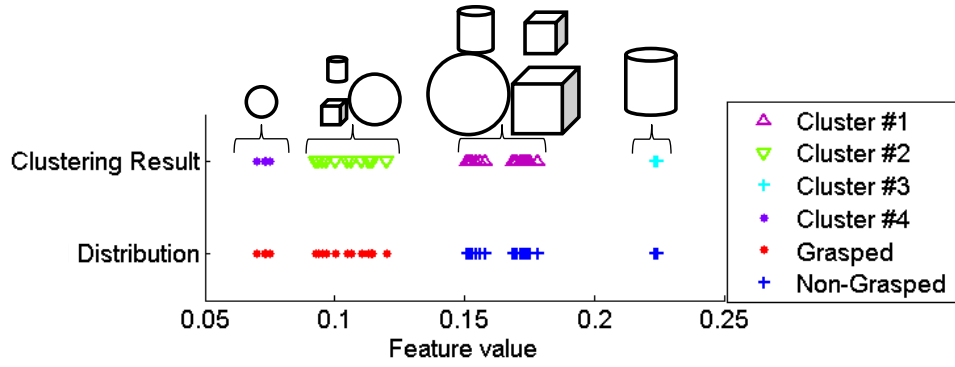


Figure 4.8: Detailed visualization of the acquired concepts for lift

4.1.3 Lift

Figure 4.7 illustrates the created concepts for the lift affordance. On the left hand side, we can see the four clusters which correspond to our object concepts. On the right hand side, we can see the desired separation on the effect space.

The acquired concepts for the lift affordance are the same with the concepts of grasp affordance because liftability depends on graspability (in our experimental setup, we cannot lift an object without first grasping it). Lifting depends on the size and weight of the objects but since the weight of the objects are best perceivable over the size of an object with our feature set, the most relevant feature for the lift behavior is size, as it is for grasp. The ReliefF algorithm gives the same weights as it gives for the grasp affordance shown in figure 4.5.

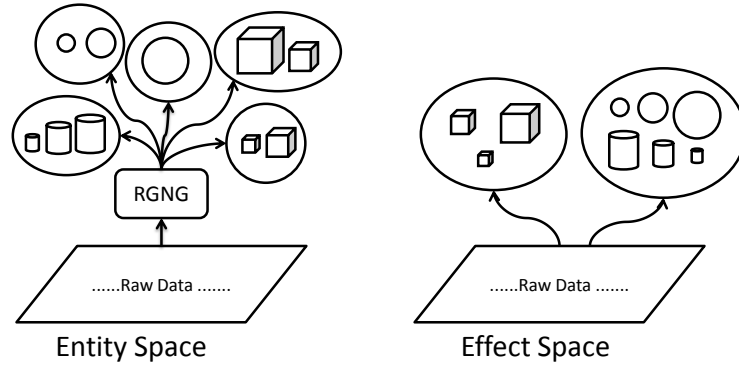


Figure 4.9: Object concepts for rotate

4.1.4 Rotate

Figure 4.9 illustrates the entity and the effect spaces and the created clusters. On the left hand side, we can see the five clusters which correspond to our object concepts. On the right hand side, we can see the desired separation on the effect space. When we inspect the contents of the acquired concepts, we can see that one concept includes small and medium sized balls, where big balls create a concept on their own. Cylinders with all sizes create a third concept. Another concept includes small and medium sized cubes and the last concept includes medium and large sized cubes. The acquired concepts for the rotate affordance depend on the shape of the objects. The 4th bin of the shape histogram has been discovered to be the most relevant feature. Figure 4.10 shows the calculated weights of features by ReliefF algorithm. Clustering over the selected feature creates a separation based on the symmetry of the objects. Symmetrical objects in our object set are balls and cylinders and non-symmetrical objects are cubes. If the shape of an object is symmetrical, then no visible change happens when we rotate the object 45 degrees. If the object is non-symmetrical, then the perceptual shape of the object changes (of course, the shape of the object remains same in reality).

We further analyze the acquired concepts by analyzing the distribution of our objects through the most relevant feature dimension. Figure 4.11 shows our objects' distribution through the most relevant feature dimension. We can say that the objects with similar shapes are close to each other. One exception to this is large balls. As mentioned in the section 4.1.1, large balls look similar to cubes due to the fact that their surface curvature is small and they seem to have flat surface patches.

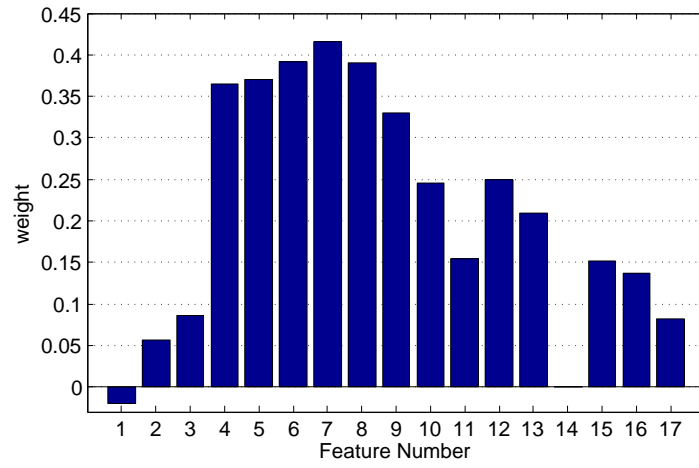


Figure 4.10: Weights of features for the rotate behavior calculated by ReliefF

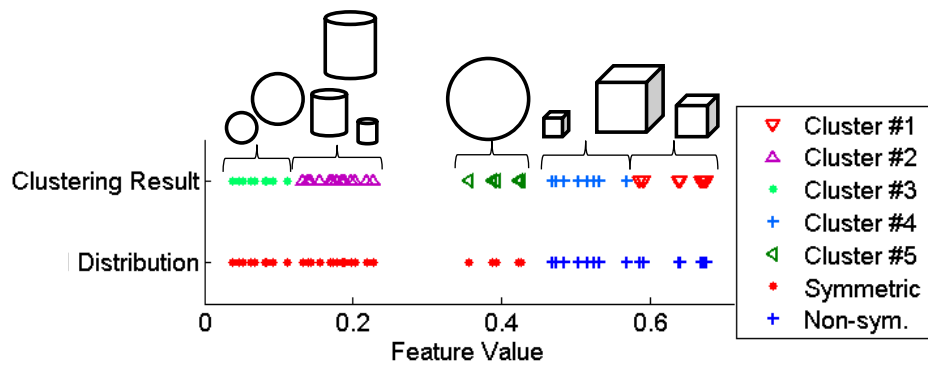


Figure 4.11: Detailed visualization of the acquired concepts for rotate

4.2 Multi-Task Learning

In this thesis, we show that our object concepts are beneficial for learning affordances. We will use multi-task affordance learning (MTL) to learn our affordances as we described in section 3.8 and our object concepts will serve as a grounding for all of our affordances. In the following four experiments, we will compare the MTL method which will learn with our object concepts with the STL method which will learn in a traditional way.

4.2.1 Object Concepts and Learning Time

The main benefit of MTL is the faster learning of related tasks, affordances in our case, due to the knowledge transfer between tasks and the ability to add or remove tasks as we please. In the first two experiments we will show how our object concepts can provide a faster learning. First, we will train the MTL network with two affordances. Then, we will measure the learning times of a third affordance by MTL and STL (Single-Task Learning) methods to find which method is more beneficial. The main difference between MTL and STL is that MTL will be using our object concepts as a grounding whereas STL will learn the affordance without using any object concepts. Moreover, in the first experiment, the new affordance will be relevant with one of the affordances the MTL network has been trained with. In the second experiment, there will be no relevancy between the new affordance and the already known affordances by the MTL network. In both experiments, the STL network will learn the new affordance as an independent task.

4.2.1.1 First Experiment - Similar Behaviors

The first experiment aims to show that the acquired object concepts create a grounding and accelerate the learning of a new affordance, which is relevant to one of the already-learned affordances by the MTL network. We start by training the MTL network with the push-left and the grasp affordances. Structure of the MTL network and the learned affordances can be seen in Figure 4.12. Then, we choose the new affordance to be the push-right affordance, which is relevant with the push-left affordance. Feature selection step of the system chooses the same feature, 9th bin of the shape histogram, for the push-left and push-right affordances. Since the system already created object concepts for push-left over the selected feature and

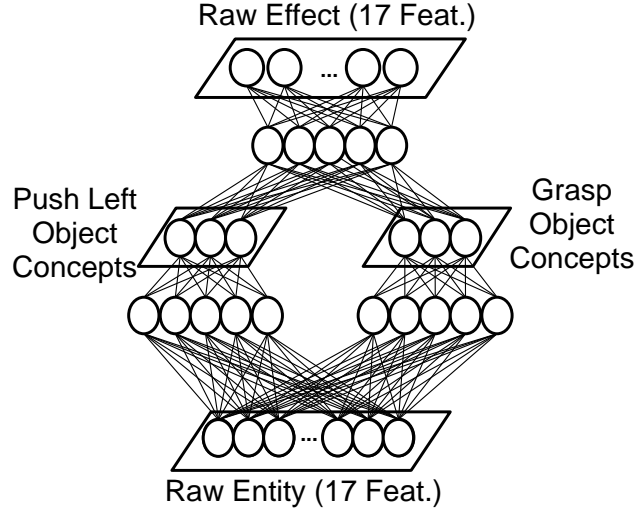


Figure 4.12: Structure of the MTL network and previously learned affordances

learned the mapping in the MTL network, the system does not perform another clustering over the selected feature. This decision saves up time at the first part of the MTL network, where the mapping from the entity space to object concepts has been learned. However, the major reduction of learning time is achieved at the second part of the MTL network where mapping from object concepts to the effect space has been learned. Since no new object concepts has been created, the only change on the training data of the second part of the MTL network is the prediction of push-right effects. In order to learn the push-right affordance, the STL network learns the mapping from the entity space to the effect space. Figure 4.13 shows the structure of the STL network.

Figure 4.14 shows the learning times of push-right affordance with MTL and STL. We repeated the learning process 200 times in order to get an averaged learning time. Each learning process continued until the mean-squared error reduces to 0.02 for both methods. It can be clearly seen that usage of object concepts greatly reduces the learning time of a new affordance.

4.2.1.2 Second Experiment - Different Behaviors

In the first experiment, the new affordance was relevant with one of the affordances the MTL network learned. In the second experiment, there will be no relevance between the new af-

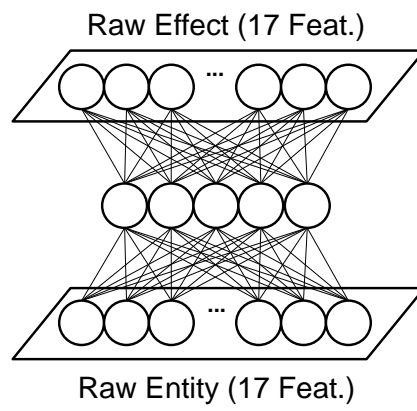


Figure 4.13: The structure of the STL network

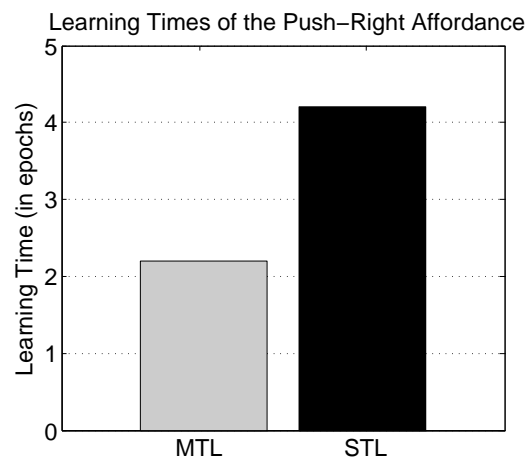


Figure 4.14: Learning times of push-right affordance by MTL and STL

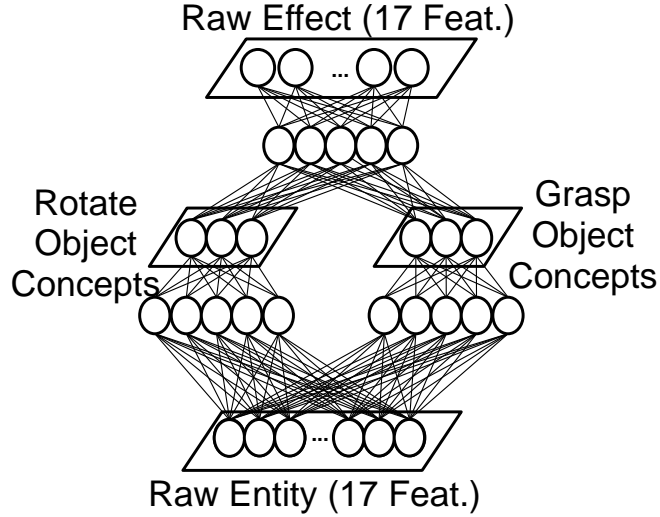


Figure 4.15: The structure of the Multi-Task Learning and previously learned affordances

fordance and the affordances the MTL network has already learned. This experiment shows that if the new affordance to be learned is not relevant with the already learned affordances, then object concepts do not provide a faster learning time. That is because the grounding, which is created by the object concepts, cannot provide help for a non-relevant affordance. However, the learning speed is not the only benefit our object concepts provide. The new object concepts will improve the already present grounding and provide faster learning for future relevant affordances. We first train the MTL network with rotate and grasp affordances. The structure of the MTL network and learned affordances can be seen in Figure 4.15.

The new affordance to be learned is the push-right affordance as it was in the first experiment. The feature selection step chooses the 9th bin of the shape histogram, as expected. The system looks whether the selected feature has been used to acquire object concepts or not in the past to prevent redundant object concept acquisition. Since both rotate and grasp object concepts have been created over different features in the entity space, the system acquires object concepts for the push-right affordance and perform learning at the first part of the MTL network. This causes an increase in the learning time. Still, like in the first experiment, the major increase in the learning time is caused by the second part of the MTL network. Since there is no similarity between the learned affordances and the new affordance, it takes longer for the MTL network to learn the new affordance than the STL network. Figure 4.16 shows the learning times of the MTL and STL networks for push-right affordance. We repeated the

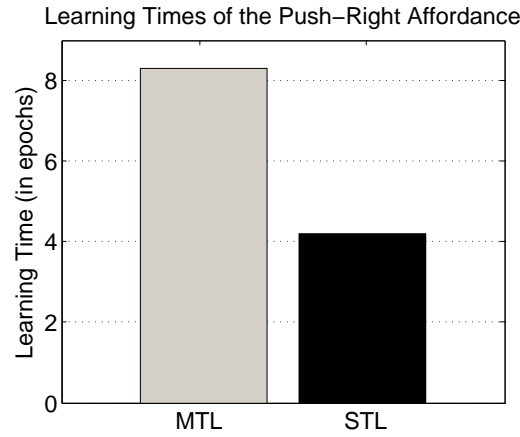


Figure 4.16: Learning times of push-right affordance by MTL and STL

learning process 200 times in order to get an averaged learning time. Each learning process continued until the mean-squared error reduces to 0.02 for both methods.

4.2.2 Object Concepts and Training Set Size

Having a grounding for some topic means we have gathered and processed knowledge on that topic. In the first two experiments we show that our object concepts, grounding for affordances, shorten the learning time of new affordances. Although the learning time is reduced, we still need to gather data before we start learning. Gathering data may not be easy depending on the setup. For example, gathering 1000 samples with a robot arm requires significant amount of time and generally machine learning tasks require larger data sets to be able to learn the given task appropriately. Thus, reducing the size of the learning set is more important than reducing the learning time because reducing the size of the learning set gives us two advantages; reduced data gathering time and reduced learning time, since there is less data to process.

Another benefit is the reduction in the number of interactions to learn a new affordance. Object concepts hold generalized knowledge about the known affordances which we can use for other related affordances. In other words, if a new affordance we try to learn is related with one of the affordances we already know then we need less number of learning samples to learn it

properly.

In our third and fourth experiments, we will mainly analyze the effect of different training set sizes on the performance. We have 45 training samples for each of our affordances. We will change the size of the training set between 15 and 45 samples with steps of 5 samples (e.g. 15,20,25...40,45). While we change the training set size, there are two ways we can compare the MTL and STL methods. We can fix the number of epochs the learning systems can use and then compare the performances or we can fix a performance goal and compare how many epochs the methods need to reach that performance goal, as we did in the first and second experiments. In the following subsections we will compare the MTL and STL methods by both ways. In all of our experiments we will use our object concepts as a grounding in the MTL method and we will not use any object concepts in the STL method.

4.2.2.1 Third Experiment: Fixed Number of Epochs

In the third experiment, we will fix the number of epochs we will allow both the MTL and STL networks to learn the new affordance. Then, we will compare the performances of MTL and STL methods to find which one can learn better with less number of training samples. Figure 4.17 shows the acquired performances with MTL and STL when we allow them to learn a single epoch (a single pass over the training set). We see that MTL achieves better performance than STL. Moreover, regardless of the training set size, STL does not perform better than the MTL. Even when STL is given 45 training samples, it cannot achieve better performance than MTL, which is trained with only 15 training samples.

We further increased the number of epochs we allow the systems to learn and found out that the dominance of the MTL method over STL method does not change with the number of epochs. This result shows that due to the usage of object concepts and the grounding they create we can learn new affordances better and faster with less number of training samples than traditional STL approaches.

4.2.2.2 Fourth Experiment: Fixed Mean Squared Error

In the fourth experiment, we will fix a performance goal (mean squared error) for MTL and STL and compare which method requires less number of training epochs with less number

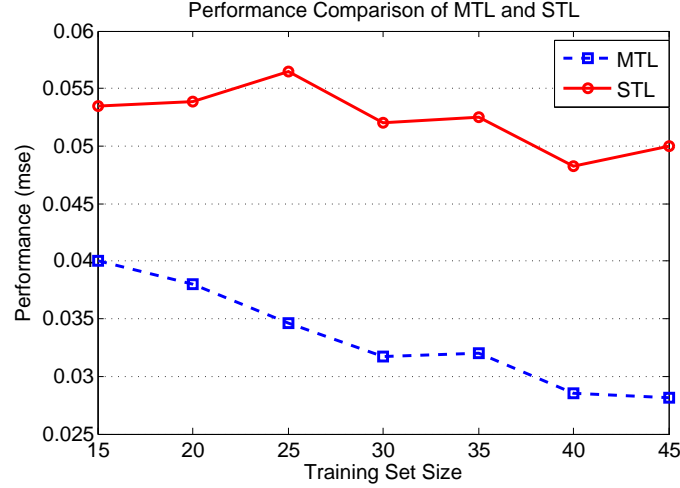


Figure 4.17: Comparison of performances of MTL and STL

of training samples. The main difference of this experiment from the first and second experiments is that we are focused on the effect of the training set size over the learning time.

Figure 4.18 shows the results of our fourth experiment. Both the MTL and STL networks try to lower the mean squared error to 0.02 as a performance goal and we repeat the learning process for each training set size 200 times to get an averaged value. The y-axis of the figure shows the required number of training epochs by the MTL and STL networks for different training set sizes. Similarly with the third experiment, MTL is better than STL for all sizes of training data. The longest training time is required for training set size of 25 samples. After this size, the required training time decreases as we increase the size of the training set.

All of the experiments we conducted show the benefits of using object concepts for the affordance learning. The first and the second experiments show the benefits of object concepts for learning time while they show the importance of task relevance for the system. When the new task, affordance, is relevant with the known tasks, learning takes shorter time. The third and the fourth experiments show the benefits of object concepts by reducing the necessary training set size and shortening the learning time. As we stated before, reduced training set size saves us time at both data gathering processes and the learning processes and is an important benefit of our system. In this thesis, we have shown the benefits of object concepts for learning affordances as a proof of concept demonstration. Object concepts create a grounding for high-level processes and can be successfully used for many different purposes.

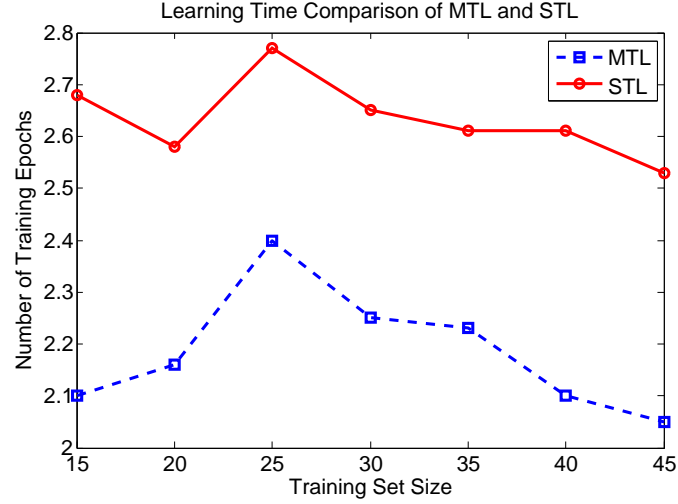


Figure 4.18: Comparison of learning times of MTL and STL

4.3 Discussion over Using Single Feature for Acquisition of Concepts

The second step of our system is the feature selection step, as we explained in section 3.7.2. The system performs a feature selection on the entity space and selects a single feature which best represents the desired separation. In this section, we will try to clarify the reasons of using a single feature for acquiring object concepts.

Since we have a set of features which encode different properties of the perceptual world (size, shape, position, orientation) we need to separate the relevant features from the irrelevant features. The system decides on the relevancy of each feature by grading its ability to represent the desired separation, as we explained in detail in section 3.7.2. A desired separation is given by a demonstrator/teacher/actor and it represents how the outcomes of a certain action should be divided. For example, the desired separation for the push behavior separate the effect of the push behavior into two as rolled and non-rolled. This clarifies the reason of performing feature selection but does not clarify why we use a single feature instead of three or five or more features. There are two reasons why we use a single feature in our system.

The first reason can be best explained by a generic example. Assume we want to acquire object concepts for a certain behavior and instead of using a single feature, we decided to use the two best features selected by our feature selection algorithm. Figure 4.19 shows the distribution of the data in our two selected artificial features. As we can see in both figures, there

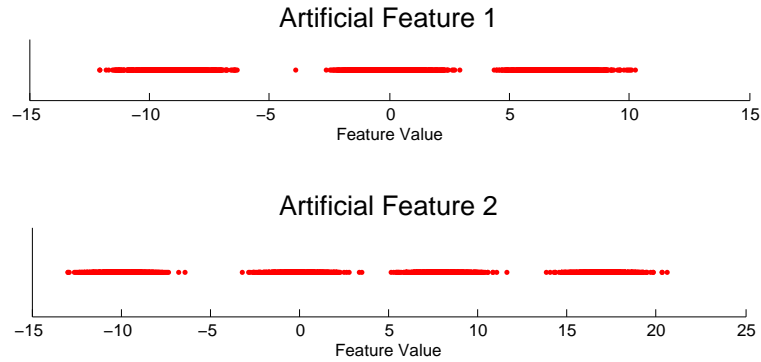


Figure 4.19: Data distribution of our artificial features

are clusters of data which will be detected as natural clusters by our unsupervised clustering algorithm and eventually become our object concepts. According to our artificial feature 1 there should be three concepts (since the unsupervised clustering algorithm will create three clusters) and according to artificial feature 2 there should be four concepts. If we were using a single feature, there would be three or four concepts according to which one of the features is the most relevant one. But, in this example, we will use both of the two best features to acquire our object concepts.

When we use both of the artificial features, we obtain a distribution as in the figure 4.20¹. It becomes clear that the number of clusters increases because the combination of the two features creates a multiplication effect. Our decision to use two features caused to create twelve object concepts. This is not good for us because we want to have as few concepts as possible in order to keep them as “*concepts*”. As number of concepts increase for a certain behavior, their ability of being a *concept* decreases since their representation becomes specialized instead of generalized.

The second reason is an ability which we want our system to have, preventing redundant work. Intuitively, we can predict that object concepts for push-left and push-right behaviors will be the same since the only changing factor is the direction of the push. However, detecting that both push-left and push-right behaviors will yield the same object concepts is not an easy task for a computer. How can we make our system detect such similarities and prevent any

¹ We are assuming that our artificial features are not correlated. Figure 4.20 demonstrates the worst case combination. In the best case (in minimum), there can be four clusters.

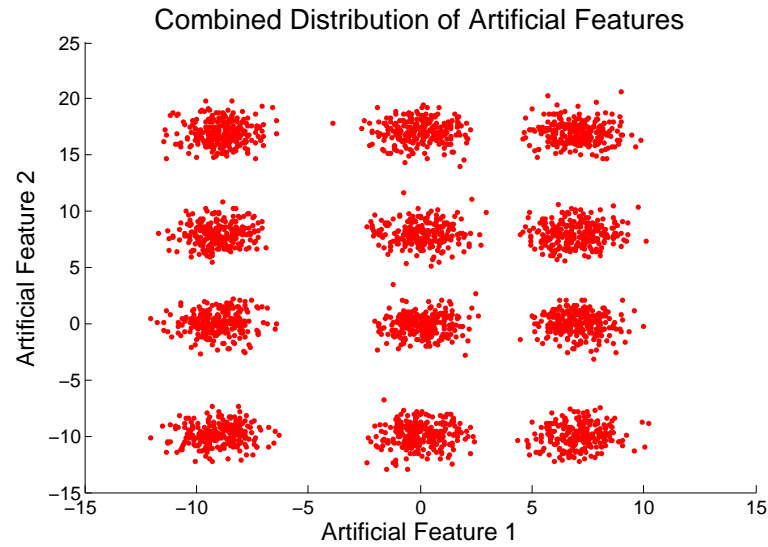


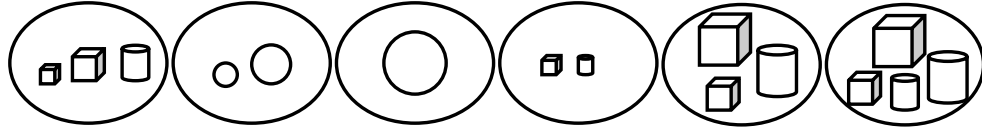
Figure 4.20: Combined data distribution of our artificial features

redundant work and duplicate concepts? We certainly do not want any duplicate concepts since they will not have any positive effects on the system.

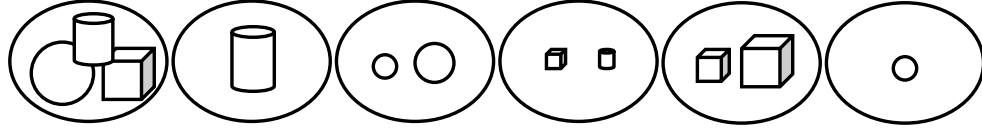
One way to solve this problem is to compare concepts with each other to find duplicate ones according to a similarity metric which will be defined by the user. This may solve the duplicate concept problem but does not solve the redundant work problem. We still have to acquire concepts before detecting that they are duplicate. The previous step of acquiring concepts is the feature selection. Push-left and push-right behaviors both depend on the same features because they are very similar. Thus, if we compare the selected features before applying unsupervised clustering we may get rid of our redundant work problem. However, there is no guarantee that two similar behaviors will select the same set of features. If we compare a large set of features our chances of having the same set of features will decrease. In order to maximize our chances we should compare only the best feature selected by the two behaviors to decide whether to perform an unsupervised clustering or not.

These two reasons we described above supports why we use only the best feature in our system. In order to provide more experimental data, we also acquired our concepts by increasing the number of features we use to two and three features for all behaviors ². Figure 4.21 shows

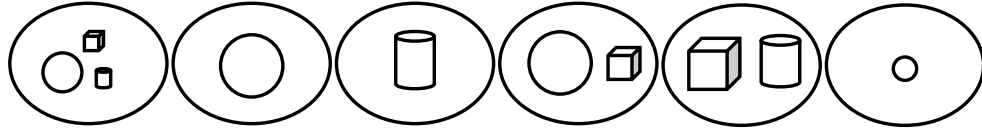
² When we further increase the number of features we use (e.g. to 10 or more) we get the shape based



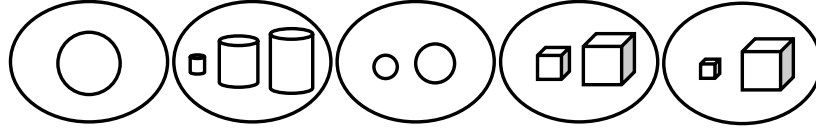
(a) Acquired concepts for push-left and push-right using two and three features



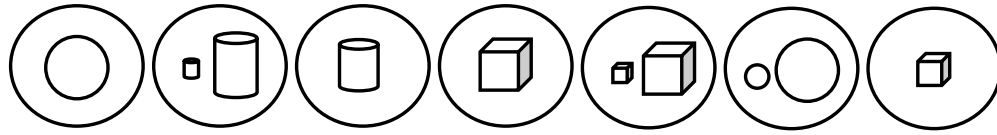
(b) Acquired concepts for grasp and lift using two features



(c) Acquired concepts for grasp and lift using three features



(d) Acquired concepts for rotate using two features



(e) Acquired concepts for rotate using three features

Figure 4.21: Acquired object concepts by using different number of features

the acquired concepts for all of our behaviors using two or three best features. Using two or three features gives more concepts than using a single feature. When we analyze the contents of the acquired concepts we can see that as we increase the number of features, some of the concepts are separated. For example, in figure 4.21(a), the fifth and the sixth concepts are due to such separation.

clustering which we mentioned in the section 3.7.3.

CHAPTER 5

Discussion

In this thesis, we presented a system which creates object concepts based on both objects' appearances and functionalities (i.e., what the objects afford). Appearance and function of objects have been used in many works separately, our work presents a novel approach by combining both of them. The object concepts acquired have been shown to be useful for learning new tasks in a simple Multi-Task Learning scenario. We compared the learning times and the training set sizes of Multi-Task Learning and Single-Task Learning to show the benefits of the object concepts.

We pointed out to the connection between object concepts and the language. Our system learned the lift, grasp, push and rotate affordances and created the object concepts via each of them. We can say that the system has created the grounding for the meanings of *small*, *big*, *round*, *angular*, *light*, *heavy* and *circular* words. Emergence of such object concepts may explain the creation process of a shared knowledge grounding of different agents and how a common grounding emerges.

Our experiments with MTL and STL also demonstrated the usefulness of the MTL approach for the machine learning. We think the robotics, embodied cognition and similar fields can gain much from MTL approaches since those fields are dominated by the STL approach currently.

We stated that the object concepts create a grounding for further uses. We pointed to the flow of knowledge during the explanation of our system which we think is an important property of our system. As system continues to interact with the environment to discover new affordances, the knowledge in the object concept grounding is used to learn the new affordance. This ability of the system facilitates the learning of new affordances.

5.1 Future Work

There are several directions which this work can be further improved. The current system uses supervised effect clusters in order to start the object concept creation process. Although this dependence of supervision for creation of effect clusters has its own benefits, performing this step as an unsupervised process may present better advantages. The system can create its own clusters in the effect space in an unsupervised way and later link them to a clustering shown by a teacher. This improvement enables the system to develop concepts without a teacher (e.g. self development). When another person (e.g. teacher) appears, the system can link this other person's meanings with its own grounding to create a common ground for the meanings. We believe this improvement can make our proposed system more open-ended and approvable. One effect of such an improvement will be the discovery of actor's affordances on its own without a need for someone to tell what to learn. This direction of future work is also relevant with the research of how similar but independent agents form a common language and grammar to communicate via simple interactions.

Another improvement can be on the feature sets. As we mentioned in section 3.4, our decisions on a feature set makes some properties of the environment more distinct or more subtle. Such a bias by a designer can and will constrain the scalability of the system (e.g. some tasks cannot be learned). A feature creation system, which creates features according to the needs of the system (current task), can greatly improve the scalability of the system. The system, as it is, assumes that any affordance to be learned is observable through the feature set and there exists at least one feature which is able to separate the success-fail groups in the effect space without much intersection. In the case that the feature set does not include a separating feature for the given task, our system is helpless and cannot correctly create the object concepts. However, with a feature creation mechanism the system can continue to learn new affordances regardless of their feature dependence. A literature survey reveals that the field of new feature creation is not intensely researched and this direction shows promising results.

REFERENCES

- [1] M. Alan. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
- [2] M.A. Arbib. *Action to language via the mirror neuron system*. Cambridge Univ Pr, 2006.
- [3] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.
- [4] A. Borghi, A. Di Ferdinando, and D. Parisi. The Role of Perception and Action in Object Categorisation. In *Connectionist Models of Cognition and Perception: Proceedings of the Seventh Neural Computation and Psychology Workshop, Brighton, England, 17-19 September 2001*, page 40. World Scientific Pub Co Inc, 2002.
- [5] A.M. Borghi. Object concepts and embodiment: Why sensorimotor and cognitive processes cannot be separated. *La nuova critica.*, 49-50:90–107, 2007.
- [6] A. Cangelosi and T. Riga. An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots. *Cognitive science*, 30(4):673–689, 2006.
- [7] S.F. Cappa and D. Perani. The neural correlates of noun and verb processing. *Journal of Neurolinguistics*, 16(2-3):183–189, 2003.
- [8] R. Caruana. Multitask connectionist learning. In *Proceedings of the 1993 Connectionist Models Summer School*, pages 372–379, 1993.
- [9] R. Caruana. Learning many related tasks at the same time with backpropagation. *Advances in Neural Information Processing Systems*, pages 657–664, 1995.
- [10] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
- [11] A. Chemero. An outline of a theory of affordances. *Ecological Psychology*, 15(2):181–195, 2003.
- [12] M. Dapretto, M.S. Davies, J.H. Pfeifer, A.A. Scott, M. Sigman, S.Y. Bookheimer, and M. Iacoboni. Understanding emotions in others: mirror neuron dysfunction in children with autism spectrum disorders. *Nature neuroscience*, 9(1):28–30, 2005.
- [13] N. Dağ. Emergence of verb and object concepts through learning affordances. Master’s thesis, Dept. of Computer Engineering, Middle East Technical University, 2010.
- [14] N. Dağ, İ. Atıl, S. Kalkan, and E. Şahin. Learning affordances for categorizing objects and their properties. *International Conference on Pattern Recognition*, 2010.

- [15] M.R. Dogar, M. Cakmak, E. Ugur, and E. Sahin. From primitive behaviors to goal-directed behavior using affordances. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 729–734. IEEE, 2007.
- [16] P.F. Dominey and J.D. Boucher. Developmental stages of perception and language acquisition in a perceptually grounded robot. *Cognitive Systems Research*, 6(3):243–259, 2005.
- [17] T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615–637, 2006.
- [18] J.A. Fodor and Z.W. Pylyshyn. How direct is visual perception?: Some reflections on Gibson’s Ecological Approach. *Vision and Mind: Selected Readings in the Philosophy of Perception*, pages 167–227, 2002.
- [19] B. Fritzke. A growing neural gas network learns topologies. *Advances in neural information processing systems*, pages 625–632, 1995.
- [20] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, 1993.
- [21] V. Gallese and A. Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in cognitive sciences*, 2(12):493–501, 1998.
- [22] E.J. Gibson. Perceptual learning in development: Some basic concepts. *Ecological Psychology*, 12(4):295–302, 2000.
- [23] J.J. Gibson. The theory of affordances. *Perceiving, acting, and knowing: Toward an ecological psychology*, pages 67–82, 1977.
- [24] A.M. Glenberg and M.P. Kaschak. Grounding language in action. *Psychonomic Bulletin & Review*, 9(3):558–565, 2002.
- [25] G. Hamerly and C. Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems*, volume 17, pages 281–288, 2003.
- [26] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.
- [27] T. Heskes. Empirical Bayes for Learning to Learn. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 367–374. Morgan Kaufmann Publishers Inc., 2000.
- [28] W.A. Hoff and N. Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):121–136, 1989.
- [29] M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, J.C. Mazziotta, and G. Rizzolatti. Grasping the intentions of others with one’s own mirror neuron system. *PLoS Biol*, 3(3):e79, 2005.
- [30] I. Kamon, E. Rivlin, and E. Rimon. A new range-sensor based globally convergent navigation algorithm for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 429–435, 1996.

- [31] J.J. Koenderink and A.J. van Doorn. Surface shape and curvature scales. *Image and vision computing*, 10(8):557–564, 1992.
- [32] E. Kohler, C. Keysers, M.A. Umiltà, L. Fogassi, V. Gallese, and G. Rizzolatti. Hearing sounds, understanding actions: action representation in mirror neurons. *Science*, 297(5582):846–848, 2002.
- [33] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *Machine Learning: ECML-94*, pages 171–182. Springer, 1994.
- [34] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.
- [35] D. Marocco, A. Cangelosi, K. Fischer, and T. Belpaeme. Grounding Action Words in the Sensorimotor Interaction with the World: Experiments with a Simulated iCub Humanoid Robot. *Frontiers in Neurobotics*, 4, 2010.
- [36] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 204(1156):301–328, 1979.
- [37] T.M. Martinetz, S.G. Berkovich, and K.J. Schulten. ” Neural-Gas” Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE transactions on Neural Networks*, 4(4):558–569, 1993.
- [38] C.B. Mervis and E. Rosch. Categorization of natural objects. *Annual review of psychology*, 32(1):89–115, 1981.
- [39] T. Mita, T. Kaneko, and O. Hori. Joint Haar-like features for face detection. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1619–1626. IEEE, 2005.
- [40] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor. Learning Object Affordances: From Sensory–Motor Coordination to Imitation. *Robotics, IEEE Transactions on*, 24(1):15–26, 2008.
- [41] U. Neisser. From direct perception to conceptual structure. *Concepts and conceptual development: Ecological and intellectual factors in categorization*, pages 11–24, 1987.
- [42] N. Nishitani, M. Schurmann, K. Amunts, and R. Hari. Broca’s region: from action to language. *Physiology*, 20(1):60, 2005.
- [43] S. Nolfi and D. Marocco. Active perception: A sensorimotor account of object categorization. In *From Animals to Animats 7: Proceeding on the Sixth International Conference on Simulation of Adaptive Behavior*, 2002.
- [44] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann.
- [45] N. Qian. Computing stereo disparity and motion with known binocular cell properties. *Neural Computation*, 6(3):390–404, 1994.
- [46] AK Qin and PN Suganthan. Robust growing neural gas algorithm with application in cluster analysis. *Neural Networks*, 17(8-9):1135–1148, 2004.

- [47] G. Rizzolatti and M.A. Arbib. Language within our grasp. *Trends in neurosciences*, 21(5):188–194, 1998.
- [48] E. Rosch and C.B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.
- [49] E. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976.
- [50] D.K. Roy and A.P. Pentland. Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146, 2002.
- [51] E. Sahin, M. Cakmak, M.R. Dogar, E. Ugur, and G. Ucoluk. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior*, 15(4):447–472, 2007.
- [52] J.R. Searle. Minds, brains, and programs. *Behavioral and brain sciences*, 3(03):417–424, 1980.
- [53] M. Steedman. Formalizing affordance. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, pages 834–839, 2002.
- [54] L. Steels and F. Kaplan. AIBO’s first words: The social learning of language and meaning. *Evolution of Communication*, 4(1):3–32, 2002.
- [55] T.A. Stoffregen. Affordances as properties of the animal-environment system. *Ecological Psychology*, 15(2):115–134, 2003.
- [56] J. Sun, J.L. Moore, A. Bobick, and J.M. Rehg. Learning Visual Object Categories for Robot Affordance Prediction. *The International Journal of Robotics Research*, 29(2-3):174–197, 2010.
- [57] S. Thrun and L. Pratt. *Learning to learn*. Kluwer Academic Pub, 1998.
- [58] MT Turvey. Affordances and prospective control: An outline of the ontology. *Ecological Psychology*, 4(3):173–187, 1992.
- [59] E. Ugur and E. Sahin. Traversability: A case study for learning and perceiving affordances in robots. *Adaptive Behavior (in press)*, 2010.
- [60] S. Ullman. Against direct perception. *Behavioral and Brain Sciences*, 3(03):373–381, 1980.
- [61] P.I. Wilson and J. Fernandez. Facial feature detection using Haar classifiers. *Journal of Computing Sciences in Colleges*, 21(4):127–133, 2006.
- [62] H. Zhao and R. Shibasaki. Reconstructing a textured CAD model of an urban environment using vehicle-borne laser range scanners and line cameras. *Machine Vision and Applications*, 14(1):35–41, 2003.