Studies in Semantic Modeling of Real-World Objects using Perceptual Anchoring
To my former supervisor Silvia Coradeschi, 
may you rest in peace.
Studies in Semantic Modeling of Real-World Objects using Perceptual Anchoring
Cover photo: Jens Hellman

© Andreas Persson, 2019

Publisher: Örebro University 2019
www.oru.se/publikationer-avhandlingar

Print: Örebro University, Repro 04/2019

ISSN 1650-8580
ISBN 978-91-7529-283-0
Abstract


Autonomous agents, situated in real-world scenarios, need to maintain consonance between the perceived world (through sensory capabilities) and their internal representation of the world in the form of symbolic knowledge. An approach for modeling such representations of objects is through the concept of perceptual anchoring, which, by definition, handles the problem of creating and maintaining, in time and space, the correspondence between symbols and sensor data that refer to the same physical object in the external world.

The work presented in this thesis leverages notations found within perceptual anchoring to address the problem of real-world semantic world modeling, emphasizing, in particular, sensor-driven bottom-up acquisition of perceptual data. The proposed method for handling the attribute values that constitute the perceptual signature of an object is to first integrate and explore available resources of information, such as a Convolutional Neural Network (CNN) to classify objects on the perceptual level. In addition, a novel anchoring matching function is proposed. This function introduces both the theoretical procedure for comparing attribute values, as well as establishes the use of a learned model that approximates the anchoring matching problem. To verify the proposed method, an evaluation using human judgment to collect annotated ground truth data of real-world objects is further presented. The collected data is subsequently used to train and validate different classification algorithms, in order to learn how to correctly anchor objects, and thereby learn to invoke correct anchoring functionality.

There are, however, situations that are difficult to handle purely from the perspective of perceptual anchoring, e.g., situations where an object is moved during occlusion. In the absence of perceptual observations, it is necessary to couple the anchoring procedure with probabilistic object tracking to speculate about occluded objects, and hence, maintain a consistent world model. Motivated by the limitation in the original anchoring definition, which prohibited the modeling of the history of an object, an extension to the anchoring definition is also presented. This extension permits the historical trace of an anchored object to be maintained and used for the purpose of learning additional properties of an object, e.g., learning of the action applied to an object.

Keywords: Perceptual Anchoring, Semantic World Modeling, Sensor-Driven Acquisition of Data, Object Recognition, Object Classification, Symbol Grounding, Probabilistic Object Tracking.
Acknowledgements

First of all, I would like to express my greatest gratitude to my supervisors, Amy Loutfi and Alessandro Saffiotti, for supporting and guiding me throughout my research, and, most importantly, for allowing me the opportunity to do what I’m doing (and doing so in my own pace). Thank you!

I would also like to show my gratitude to a number of people that have, in one way or another, been "anchored" to my life during my studies by referring to the following figure:

Figure X: Many thanks goes to: 1) my family, 2) my close friends and fellow beer drinkers, 3) the community of the ReGROUND project, 4) the colleagues at Örebro University, 5) the fellow researchers of the Machine Perception and Interaction (MPI) lab., and 6) my "outdoor coffee break" buddies.

Another big thank you goes to Göran Falkman for his feedback and his time spent on reviewing this thesis.
Last but not least, I would like to express my thankfulness for the support during the final hours of compiling this thesis by extending a special thank to: Ida Andersson-Norrie for coordinating my life beyond this thesis, Kalle Räisänen for spelling the checking, Jens Hellman for taking photographic pictures, and Erik Norgren for realizing the physical copy of this thesis.

Thank you all!
## Contents

1 Introduction ............................................. 1  
1.1 Background and Motivation .......................... 1  
1.2 Problem Statement and Objectives ................. 3  
1.3 Research Questions .................................. 5  
  1.3.1 What are Available and Open Resources? ....... 6  
  1.3.2 When are Scenarios Beyond "Toy Examples"? .... 7  
  1.3.3 Why Matching of Anchors at Perceptual Level? 7  
1.4 Methodology ........................................... 8  
1.5 Contributions ........................................ 9  
1.6 Thesis Outline ....................................... 11  

2 Related Work ............................................. 13  
2.1 Semantic Visual Perception ......................... 13  
  2.1.1 Perception of Objects .......................... 14  
  2.1.2 Instance and Category Level of Perception .... 15  
  2.1.3 Perception of Actions Applied to Objects .... 18  
2.2 Semantic World Modeling ............................ 19  
  2.2.1 Tracking and Data Association ................. 19  
  2.2.2 Characteristics of World Modeling ............ 20  
2.3 Available On-line Resources of Information ......... 21  
  2.3.1 Large Image Datasets .......................... 21  
  2.3.2 Robotics with Cloud Resources ................. 21  
2.4 Perceptual Anchoring ................................ 23  
  2.4.1 The Evolution of Perceptual Anchoring ......... 23  
  2.4.2 Probabilistic Perceptual Anchoring ............. 24  

3 Rudimentaries of Perceptual Anchoring ................. 27  
3.1 Components of Anchoring ............................ 27  
3.2 Anchoring Functionalities ............................ 29
4 An Architecture for Perceptual Anchoring
   4.1 Architecture Overview .................................. 34
      4.1.1 System Description Outline ...................... 35
      4.1.2 Notes on Anchoring Notation .................... 36
   4.2 Perceptual System ..................................... 36
      4.2.1 Input Sensor Data .................................. 37
      4.2.2 Object Segmentation ............................. 37
      4.2.3 Feature Extraction ............................... 38
      4.2.4 Extract Binary Descriptors ...................... 39
      4.2.5 Used Attributes in Relation to Publications ....... 39
   4.3 Symbolic System ..................................... 41
      4.3.1 Predicate Grounding .............................. 41
      4.3.2 Object Classification ............................ 42
   4.4 Anchor Management System ........................... 43
      4.4.1 Matching Function ............................... 44
      4.4.2 Creating and Maintaining Anchors ............... 47
      4.4.3 Learning the Anchoring Functionalities .......... 49
   4.5 Discussion .......................................... 53

5 Tracking of Anchors Over Time ................................ 55
   5.1 Extension of the Anchoring Definition ................... 56
      5.1.1 Maintaining the Historical Trace of an Anchor ...... 56
   5.2 Variations of Object Tracking .......................... 57
      5.2.1 Object Tracking on Perceptual Level .............. 57
      5.2.2 High-Level Object Tracking ...................... 59
   5.3 Anchoring Track Functionality ........................ 59
   5.4 Learning Movement Actions from Maintained Anchors ...... 60
      5.4.1 Evaluation of Sequential Learning Algorithms ...... 61
      5.4.2 Using Action Learning to Improved Anchoring ..... 61
   5.5 Reasoning About Relations Between Anchored Objects .... 63
      5.5.1 Proof-of-Concept ............................... 63
   5.6 Discussion .......................................... 65

6 Anchoring to Talk About Objects ................................... 67
   6.1 Top-Down Anchoring .................................. 67
   6.2 Integration of a Multi-Modal Dialogue System .......... 68
      6.2.1 Semantic System ............................... 68
      6.2.2 Dialogue System ............................... 70
      6.2.3 Establish Joint Communication Routines .......... 70
   6.3 Fluent Spoken Dialogue about Anchored Objects ........ 71
      6.3.1 Semantic Ambiguities .......................... 72
   6.4 Discussion .......................................... 73
7 Conclusions 75
  7.1 Challenges and Contributions .................................. 75
  7.2 Critical Assessment ........................................... 76
  7.3 Future Work .................................................. 79

References 81
List of Figures

X Acknowledgements ......................... iii
1.1 An illustration of conceptual spaces .................. 7
3.1 Graphical illustration of the anchoring components .............. 29
3.2 Illustration of the anchoring functionalities .................. 30
4.1 An architecture overview of the anchoring framework .............. 35
4.2 Example of segmented objects .......................... 38
4.3 An illustration of distinct key-point features .................. 40
4.4 The procedure for grounding color predicates .................. 42
4.5 Grounding of size attributes .......................... 42
4.6 Example of symbolically classified objects .................. 43
4.7 Example of anchored objects .......................... 48
4.8 A depiction of proposed human-annotation interface .............. 50
4.9 Average classification accuracy of learning the anchoring functionalities through supervised classification .............. 51
4.10 Average classification accuracy of learning the anchoring functionalities through semi-supervised classification .............. 52
5.1 Anchoring in combined with different object tracking approaches .......................... 57
5.2 A depiction of adaptive particle filter-based object tracking .................. 58
5.3 Exemplified feedback loop for improving anchoring from action learning .................. 62
5.4 Benefits of combining object anchoring with probabilistic object tracking .................. 64
6.1 Integrated multi-modal dialogue and anchoring framework .............. 69
6.2 Ambiguous candidates with respect to a growing anchor-space .............. 72
List of Tables

2.1 Time-line over the evolution of perceptual anchoring. . . . . . . 25
4.1 A summary of attributes used for various presented publications. 40
5.1 Confusion matrices for the learning of object movement actions. 61
6.1 Acceptable queries used for dialogues about anchored objects. . 71
List of Publications

The work behind this thesis has partly been published in a number of papers. The papers are referred throughout this thesis using their Roman numerals, as they are presented below.

Publications Included in this Thesis


Author Contributions

Regarding the publications presented in this thesis, the particular contributions by the author of this thesis (A. Persson) can be summarized as follows:

- Paper I: A. Persson phrased and motivated the use of a database-centric approach in the context of perceptual anchoring. In particular, A. Persson exemplified how the use of database approaches can support both
the symbol grounding, as well as the anchoring procedure per se. Moreover, A. Persson integrated and evaluated different database approaches in relation to efficient storage and maintenance of anchored objects. Furthermore, A. Persson prepared the manuscript together with A. Loutfi.

Paper II: Firstly, A. Persson was individually responsible for: 1) the design and development of the presented bottom-up anchoring framework, including the introduction of the use of publicly available resources for the purpose of support the anchoring procedure, 2) the formulation of the definition of the proposed anchoring matching function, and 3) the collection of data and the evaluation of the proposed anchoring procedure. Secondly, A. Persson integrated probabilistic reasoning into the anchoring and presented the proof-of-concept of reasoning about anchored objects together with P. Zuidberg Dos Martires. Finally, A. Persson prepared the manuscript together with the other authors.

Paper III: A. Persson contributed through: 1) the development of the framework suggested for anchoring and tracking object trajectories, both in space and time, 2) the collection of data used as training data for evaluated learning algorithms, and 3) the conceptual improvement of anchoring through the use of the coupling between predicted actions and object categories. Moreover, A. Persson specified an extension of the anchoring notation, which was required for this work in order to maintain the trajectories of anchors over time. Finally, A. Persson prepared and revised the manuscript together with M. Längkvist and A. Loutfi.

Paper IV: A. Persson conducted a survey regarding the use of binary feature descriptors in the context of object classification tasks in dynamic environments. Following the initial survey, A. Persson defined, implemented and evaluated the proposed algorithm. Furthermore, A. Persson prepared and revised the manuscript together with A. Loutfi.

Paper V: A. Persson was, firstly, responsible for the development and the implementation of the proposed anchoring framework. Secondly, A. Persson integrated the anchoring framework with a multi-modal dialogue system and conducted the experiments together with S. Al Moubayed. Thirdly, A. Persson established the use of a lexical databases for the purpose of both extending natural language descriptions, as well as resolving semantic ambiguities. Finally, A. Persson prepared and revised the manuscript together with S. Al Moubayed and A. Loutfi.

Other Publications of Relevance


Chapter 1
Introduction

In this chapter, we introduce the background and the motivation for the use of perceptual anchoring in relation to semantic world modeling situations. We specify the problem statement and describe how previous work on anchoring has been done in small-scale with "toy examples". Motivated by the intention of shifting anchoring towards real-world scenarios, we further portray the need for different techniques and different sources of information in order to address this challenge.

1.1 Background and Motivation

Consider the classical street performer's trick where a ball is hidden under one of three identical cups. The performer rapidly moves the cups, and the task of the observer is to follow the movement of the cups and to identify under which cup the ball is located. For an observer to successfully identify the right cup, he/she must successfully handle a number of subtasks. First, despite the fact that each of the cups is visually similar, the observer must create an individual notion of each cup as a unique object so that it can be identified (e.g. "the cup in the middle"). Likewise, the observer must recognize the ball as a unique object. Secondly, even though the ball is hidden under one of the cups, the observer makes the assumption that although the ball is not perceived, it should still be present under the cup. Third, as the performer rapidly moves the cups, the observer should track the cup under which the ball is hidden. Finally, the observer also needs to realize that cups can contain balls, and therefore as the cup moves, so does the ball. Depending on the level of skill of the performer (and perhaps some additional tricks) the street performer's cup trick can be a difficult one to solve.
Imagine that the observer in this scenario is an autonomous agent that perceives the world through perceptual sensory data. For the agent to handle this type of real-world situated scenario in a similar manner as a human would, the agent must maintain a consonance between the perceived world (through sensory capabilities) and its internal representation of the world. To achieve consonance, the agent must, first of all, interpret and process the sensory data that stem from each of the objects involved in the scenario and internally create a representation that encompasses the numerical properties that both distinguish an object from other objects (e.g., the position of an object), and that identify an individual object. However, this internal representation must further contain symbolic properties (e.g. the symbol ‘cup’), in order to enable the agent to refer to the objects by meaningful semantic symbols that are understandable by the human counterpart. The agent must therefore, in addition, handle the problem of symbolically grounding numerical properties to analogous meaningful semantic symbols. Moreover, for an agent to manage the scenario above, it is not sufficient to merely create each individual representations for every object. As the street performer rapidly moves the cups, the agent must further track and maintain the properties of each representation as corresponding objects are moved in space and over time. However, this requirement is particularly challenging for circumstances of occlusions and absence of perceptual updates about objects (i.e., while the ball is hidden underneath one of the cups). This entails that the agent can not entirely rely upon percepts from objects in order to update and maintain the properties of the internal representation of an object. Instead, the agent must probabilistically reason about certain properties (e.g the position of an object), and the relationships among objects (e.g. that cups can contain balls and as the cup moves, so does the ball), in order to track and maintain an updated representation of occluded objects.

A notion that aims to address the challenge of creating internal representations of objects (as discussed above), is the theory behind semantic world modeling. As discussed by [30], a semantic object model is a way of representing objects not only by their numeric properties, such as a position vector, but also by semantic properties which share a meaning with humans. A practical method for advancing the task of modeling such semantic representations of objects, perceived in the physical world, is through the concept of perceptual anchoring, initially presented by Coradeschi and Saffiotti in [18]. Perceptual anchoring, by definition, handles the problem of creating and maintaining, in time and space, the correspondence between symbols and perceptual data that refer to the same physical object. Those percept-symbol correspondences are, practically, encapsulated and managed in an individual internal representation, or anchor. The problem of creating and maintaining anchors was, subsequently, defined as the anchoring problem, first defined in [19]. In this thesis, we have explored the use of bottom-up anchoring [84], whereby anchors can be created by perceptual observations derived from interactions with the environment. For a practicable bottom-up anchoring framework, it is, however, essential to have
a robust anchoring matching function that accurately matches perceptual observations against perceptual data of previously maintained anchors. For this purpose, we have in this thesis introduced a novel method that approximates the anchoring matching function through a learned model.

Nonetheless, perceptual anchoring is a deterministic approach that requires observations of objects in order to maintain anchors. In the example above, situations will arise where the stream of perceptual sensor data for an individual object is compromised and the corresponding anchor is deterministically not updated, e.g., when the ball is hidden under one of the cups. In the absence of measurements, it is then necessary to speculate about the actual state of the real world given the compromised stream of perceptual data. The approach for handling such situations, presented in this thesis, is to couple the anchoring procedure with a probabilistic reasoning approach that can reason about objects and their relations, and thereby support the tracking of objects in case of occlusion (e.g., due to limitations in perception or due to interactions with other objects). The probabilistic nature of such a reasoning approach based on particle filters for high-level data association and object tracking [94], enables the anchoring framework to further handle both occlusions of objects through their relations with other objects, as well as updating the believed state of an anchored object, even though the object is not directly observed.

1.2 Problem Statement and Objectives

Based on the identification of limitations in previously presented anchoring approaches, we will in this section outline the problem statement and state the research objectives of this dissertation. The overall intention of this thesis is to propose a perceptual anchoring architecture that aims to address the problem of semantic modeling of real-world objects. However, the specific objectives, based on identified restraints in previously anchoring approaches, can be expressed as follows:

O1: Introducing a practical semantic world modeling approach relevant for any type of autonomous agent.

Perceptual anchoring was, initially, motivated by an emerging need for robotic planning systems to plan and execute actions involving objects [113]. The problem of anchoring objects has, therefore, mainly been investigated from the perspective of an autonomous mobile robot, e.g., in robot navigation scenarios. In this thesis, we demonstrate that anchoring is a concept that applies to any autonomous agent (static or mobile), which is required to create and maintain a semantically rich world model of perceived objects.

O2: Presenting a bottom-up architecture for sensor-driven acquisition of perceptual observations.
Perceptual anchoring has traditionally been considered a special case of symbol grounding [20], which concerns perceptual sensor data and symbols that denote physical objects. Previously presented works on anchoring have, therefore, mainly been presented in a top-down fashion with the emphasis on the symbolic level, while the continuous information found on the perceptual level has often been neglected once mapped to symbolic values (e.g., through the use of conceptual spaces [16]). For the work presented in this thesis, we have, instead, explored the use of sensor-driven bottom-up anchoring [84]. We demonstrate that a bottom-up anchoring approach moderately resembles the problem of data association [4]. In similarity with data association, a bottom-up anchoring approach must address the challenge of associating, modeling and maintaining a semantically meaningful world model based on measurements from percepts that are produced by the same object across different views and over time.

O3: Extending the architecture with probabilistic object tracking to maintain the state of objects in the absence of perceptual observations.

Probabilistic data association in relation to perceptual anchoring has been explored in previous work on probabilistic anchoring [10, 30]. However, previously presented works on the subject of probabilistic anchoring have suggested a tight coupling between object anchoring, probabilistic data association, and object tracking. For example, Elfring et al. suggested the use Multiple Hypothesis Tracking based data association (MHT)[103], in order to maintain changes in anchored objects, and thus, maintain an adaptable world model. However, a MHT procedure will inevitably suffer from the curse of dimensionality [8], and a highly integrated probabilistic anchoring approach will, therefore, further propagate the curse of dimensionality into the concept of anchoring. In this thesis, we will, instead, suggest that a loose coupling should be maintained in order sustain the benefits of both the ability to maintaining individual instances of objects in a larger scale, as in the case of perceptual anchoring, as well as efficiently and logically tracking object instances over time, as in the case of probabilistic object tracking.

O4: Integrating available resources of information to support the bottom-up acquisition of objects.

Previously presented work on anchoring has exclusively considered small-scale "toy examples", i.e., scenarios with a controlled environment and with a fixed number of possible objects that an agent might encounter. In this thesis, we move away from such toy examples and, instead, emphasize uncontrolled real-world situations of a larger scale, and with a changeable number of possible objects that an anchoring agent
might encounter. A changeable number of objects further implies that it is not possible to train a model to account for the vast variety of objects that the agent might encounter, and the agent must, therefore, rely upon other types of resources in order to, e.g., recognize and identify objects.

In a broader perspective, an ultimate objective of the proposed anchoring architecture is to establish a practical platform for both future academical studies (e.g., studies regarding robot manipulation of novel objects), as well as industrial applications (e.g., automated processes that require an object-centered representation of the environment). Whether this ultimate objective will be fulfilled, however, only the future can tell.

1.3 Research Questions

Shifting the focus from controlled "toy example" scenarios to uncontrolled real-world scenarios, together with the use of the full spectra of measurements that emerge from the use of both spatial and visual sensory data, will simultaneously introduce new challenges in the context of the anchoring problem – challenges that in small-scale scenarios are solved ad-hoc or by simplified means, and which can be outlined as follows:

Q1: How to efficiently and accurately, and without prior knowledge, recognize perceived objects as previously observed objects?

A controlled environment, with a fixed number of pre-trained objects, further implies a discrete number of objects to consider in an object matching procedure. Consequently, a matching candidate object is (simply) selected in a "winner-takes-all" manner. The same implication is not true for real-world scenarios with unlimited possibilities of objects. An anchoring agent without prior knowledge of possible objects that the agent might encounter, must exclusively rely upon a matching function that compares the measurements (or attribute values) of a perceived candidate object against the measurements of all previously anchored objects in order to determine if the candidate objects have previously been perceived (or not). This matching function must further account for a changeable number of perceived objects and efficiently handle the matching whilst an increasing number of objects is perceived over time.

Addresses objective O2, which, consequently, directs objective O1.

Q2: How can available and open resources of information support the processing and grounding of perceptual data?

Approaching real-world scenarios without prior knowledge of possible objects that the agent might encounter means that, to support the processing and grounding of perceptual data, other sources of information
must be considered in pursuit of detecting and identifying an individual object among the vast variety of possible objects. For this type of resource to rightly bolster the detection, recognition, and grounding of objects, the source of information must not only be of such a scale that it can account for the all the possibilities of objects within the domain of a scenario, but the data must also be conveniently available for use.

Directs objective O4, and supports objective O2.

Q3: *How to facilitate scalable maintenance and tracking of objects that are not directly perceived through perceptual sensor data?*

Proper data association and model-based object tracking are essential for object anchoring, and, consequently, for semantic modeling of real-world objects (as discussed by [30]). The probabilistic nature of tracking-based approaches will, however, introduce an additional dimension of complexity since perceived objects are maintained in a state of multiple hypotheses. Besides, there are some domain characteristics that differentiate semantic world modeling from target tracking (as outlined by [136]), e.g., a world modeling approach should take into consideration that most object states do not change over short periods of time. It is, therefore, important to separate semantic world modeling, through anchoring, from object tracking, in order to sustain the benefits of both the ability to maintaining individual instances of objects in a larger scale, as in the case of anchoring, as well as to efficiently track object instances over time, as in the case of probabilistic object tracking.

Addresses objective O3, and, likewise, supports objective O2.

### 1.3.1 What are Available and Open Resources?

By *available and open resources*, we refer to two particular types of sources of information in this case:

- **Online Resources**
  
  Resources publicly available on Internet. In this case, we consider two different type of resources: 1) *on-line resources*, i.e., information used directly in combination with the anchoring functionalities, e.g., the ConceptNet semantic network [118], and 2) *off-line resources*, i.e., information collected and used for off-line training and learning, e.g., the ImageNet database [29].

- **Statistical Information**
  
  The anchors created and maintained in an *anchor-space* as a resource per se. In this case, we consider both statistical information extracted from previously stored anchors, as well as the historical trace of anchors in order to learn additional properties of objects.
1.3. RESEARCH QUESTIONS

1.3.2 When are Scenarios Considered to be Beyond "Toy Examples"?

As stated above, we will in this thesis move away from the so-called "toy examples", and shift the domain of anchoring towards real-world situated scenarios. In this context, it is, however, not clearly defined when a scenario is considered to be beyond a toy example. To clarify what we mean when we refer to a scenario as real-world situated, we make the following assumptions:

- In order to account for scalability, an autonomous agent that is operating in real-world situated scenarios can assume neither that the particular types of objects nor the number of individual objects which the agent might encounter are known in advance.

- It is not possible to train a model to account for all the possible object instances that an autonomous agent might encounter in real-world situated scenarios. The agent must, therefore, rely upon other types of resources to recognize and identify objects.

1.3.3 Why Matching of Anchors at Perceptual Level?

In previously reported work on perceptual anchoring, the problem of matching anchors has mostly been addressed through a simplified approach based on the use of symbolic values (or left out entirely), where the predicate grounding relation mapping between symbolic predicate values and measured attribute values commonly is facilitated by the use of conceptual spaces [16], as exemplified in Figure 1.1.

![Figure 1.1](image)

**Figure 1.1:** An illustration of how conceptual spaces are used for mapping the predicate grounding relation between symbolic predicate values and measured attribute values.
A matching procedure based on symbolic values will inevitably also introduce two areas of concern, which motivate the use of an anchoring matching procedure at the perceptual level, as presented in this thesis:

- **Ambiguous results**: a symbolic system consists of a discrete number of symbols, and a matching procedure based on symbolic values might, therefore, introduce ambiguous results. The procedure of creating and maintaining anchors, based on the symbolic values, must consequently either be handled by a probabilistic system, as in the case of [30] or through the use of additional knowledge and the use of a reasoning system, as in the case of [25, 24].

- **Loss of information**: mapping measurable attribute values to predicate symbols, as exemplified in Figure 1.1, can be thought of as a discretization of the continuous perceptual attribute space. Matching anchors through a finite number of symbolic values does, therefore, not harness the full spectra of available information.

Moving the anchoring matching procedure to the perceptual level will, nevertheless, unconditionally introduce another level of complexity, since anchors must be compared based on continuous attribute values. The system must subsequently both recognize previously observed objects and detect (and anchor) new, previously unknown objects based on the result of an initial matching function. This is undoubtedly a challenging issue in scenarios beyond toy examples without a fixed number of possible objects that the system might encounter. It is largely this challenge that is addressed in this thesis.

### 1.4 Methodology

The anchoring architecture that is presented in this dissertation has evolved over the years of studies and together with the publications that are included in this thesis. There are, however, a few milestones that changed the course of action regarding chooses of methods and architecture decisions. In this section, we summarize those methods and decisions.

At the beginning of our research, we naively assumed that the problem of efficiently and accurately recognizing perceived objects as previously observed objects (Q1), in a bottom-up fashion, could solely be handled by distinct visual key-point features. This assumption further motivated studies beyond the primary objectives of this thesis, e.g., studies of application domains for an anchoring framework. Recognizing and maintaining objects based on visual key-point features is, however, a computationally demanding process. A couple of approaches have been presented that aims to reduce the computational complexity in the use of visual key-point features ([42, 90]). As a result of our early studies, we have introduced a similar approach that aims to address the
complexity of recognizing objects based on visual key-point features. The approach was systematically evaluated, and the results were showing a promising accuracy for evaluated datasets. The proposed approach required, however, the full view of an object in order to accurately recognize an object as a novel object or a previously observed object, which is not always the case for objects in dynamically changing environments.

Later in the studies, we changed focus and were, instead, investigating the use of publicly available resources of information for the purpose of supporting the anchoring process (Q2). In particular, we surveyed, at the time, recent trends in deep learning and publicly available large image datasets (e.g., the ImageNet database [29]). As a result of this survey, a Convolutional Neural Network (CNN) based object classification procedure ([120]), was integrated and used for the purpose of symbolically categorizing perceived objects at the perceptual level. A first connection to object tracking approaches (Q3), was further introduced alongside the integration of the object classification procedure. Due to an identified limitation in the original anchoring definition ([18, 19]), an extension of the definition was, in addition, proposed such that historical trace of an object, tracked in space and time, could be maintained. This extension paved the way for research in how additional object properties could be learned based on the maintained historical trace of an object, e.g., learning of the movement action that is applied to an object. Nonetheless, the problem of recognizing and anchoring perceived objects per se, was not further studied during this period of research, and the problem was, instead, handled ad-hoc by a suboptimal rule-based approach.

In the most recent research, we once again turned our attention towards the problem of accurately recognizing and correctly anchoring perceived objects (Q1). In addition, we have studied how the anchoring process can be facilitated and supported by the extension of high-level probabilistic object tracking (Q3), in order to speculate about the state of objects that are not directly perceived by the input sensory data, e.g., in case of object occlusions. Based on identified deficiencies in the anchoring approaches used in previous studies, we have for this research both formulated the theoretical procedure for recognizing perceived objects as previously observed objects, as well as hypothesized that the problem of anchoring objects, in a bottom-up fashion, is a problem that can be learned from examples. The formulated hypothesis was, subsequently, confirmed through quantitative validations, while the benefit of integrating high-level probabilistic objects tracking has, likewise, confirmed and portrayed through the proof-of-concept.

1.5 Contributions

The main contribution presented in this thesis is the introduction of an anchoring framework that is able both to handle and process the stream of computationally demanding spatial and visual perceptual data (which is provided by an
$RGB-D$ sensor), as well as managing the creation and maintenance of anchors without prior knowledge about perceived objects. This framework is the result of several specific contributions, which can be summarized as follows:

**C1:** Addressed the problem of *bottom-up anchor matching* within real-world situated scenarios, i.e., a dynamic scenarios with an arbitrary number of objects. Two improvements, in particular, have been introduced for the sake of practically anchoring objects in a bottom up manner: 1) a theoretical procedure for comparing attribute values measured from percepts, and 2) a novel method that approximates the anchoring matching problem through a learned model. (Chapter 4)(Paper I & Paper II)

| Question(s): | ✓ | □ | □ |

**C2:** Extended the original anchoring definition [18], such that the historical trace of an anchor is maintained while an anchored object is moved and tracked in space and time. (Chapter 5)(Paper III)

| Question(s): | □ | □ | ✓ |

**C3:** Tackled the challenge of enhancing the anchoring procedure with a probabilistic tracking functionality, which supports the anchoring procedure in case of limitations in perception, e.g., in case of occlusions. For this purpose, two different approaches have been examined (further addressed in Chapter 5):

- High-level object tracking through the integration of an inference system that can reason about objects and their relations, and thereby support the anchoring procedure. (Paper II)
- Object tracking based on 3-D point cloud data directly at the lowest perceptual level. (Paper III)

| Question(s): | □ | □ | ✓ |

**C4:** Integrated available and open on-line information resources into anchoring. In the interest of establishing the percept-symbol correspondence, two techniques, in particular, have been employed (further presented in Chapter 4):
1.6 Thesis Outline

The structure of this thesis is organized as follows:

**Chapter 2:** In this chapter, we summarize the field of research that is related to the studies in this thesis. In particular, we emphasize on bottom-up approaches for *semantic visual perception*, as well as *semantic world modeling*. We further declare related areas on the use of *available and open resources of information*, before concluding the chapter by summarizing the field of research related to *perceptual anchoring*.

**Chapter 3:** Based on the traditional anchoring definition [18], we will in this chapter present the background of perceptual anchoring, including
both the core components and definitions, as well as the main anchoring functionalities.

**Chapter 4:** In the fourth chapter, we present the essential method of this thesis – an anchoring architecture for maintaining a consistent notation of objects based on object attribute values.

**Chapter 5:** In this chapter, we present an extension of the anchoring framework that permits *probabilistic tracking of objects*, which support the deterministic anchoring procedure in cases where no observations of an object are given (e.g., in case of object occlusions).

**Chapter 6:** In the sixth chapter, we focus on the application domain of anchoring by presenting the integration between an anchoring framework and a multi-modal dialogue system. We further demonstrate how the integrated framework is utilized for maintaining a fluently spoken dialogue about anchored objects in human-robot interaction scenarios.

**Chapter 7:** In this last chapter, we conclude this thesis by a summary of the challenges and contributions. We further raise a couple of points of attention during a critical assessment of the presented work together with an outline of possible directions for future work.
Chapter 2
Related Work

In this chapter, we survey related approaches found in literature and outline the areas of related work. For the work presented in this thesis, we have focused on the problem of anchoring visual perceptual data. In the initial Section 2.1 of this chapter, we, therefore, survey the fields related to visual perception. The concept of perceptual anchoring has in recent years been explored in synergy with the problem of semantic world modeling. In Section 2.2, we accordingly summarize related work on the topic of semantic world modeling. Furthermore, as our intentions for the work behind this thesis are to explore the use of available and open third-party sources of information for the purpose of assisting the perceptual anchoring procedure, we continue by presenting the related work in the domain of available resources of information, presented in Section 2.3. In this section, we further survey similar architectures with cloud capabilities that are found in the literature, i.e., architectures that utilize online knowledge-base systems that provide information through web and cloud services. Finally, we conclude this chapter by presenting the related work and outlining the evolution of perceptual anchoring per se, presented in Section 2.4.

2.1 Semantic Visual Perception in Robotics

The area of visual perception spans across several sub-topics commonly associated with computer vision [121], e.g., object detection, object recognition, etc. Throughout the works presented in this thesis, we explore such computer vision-based methods and techniques for the purpose of extracting and processing visual perceptual data of individual objects. In particular, we emphasize the problem of semantic visual perception. The related works presented in this section are, therefore, restricted to the literature of presented approaches in which the semantic meaning of the perceived surrounding environment is considered, and in notably, related works which focus on the problem of grounding semantic symbols to visual perceptual sensor data.
2.1.1 Perception of Objects

Traditional object detection, in conditions of 2-D visual data, is commonly based on a Cascade Classifier [133, 79], which utilizes a sliding window approach to search for Regions of Interest (ROI) that match a pre-trained pattern. The classifier is typically trained based on feature patterns, such as Haar-like features [79] or Local Binary Patterns (LBP) [78], which are extracted from positive samples of the object of interest (e.g., a ‘car’ or a human ‘face’). In the interest of grounding semantic symbols through the use of this type of object detection approach, the object category is, consequently, given directly based on the category of known samples that was used during training of the detector. However, separate object detectors must be trained for each object category of interest, which makes this type of approach impractical for the detection of objects among several object categories.

3-D Perception

The emergence of affordable RGB-D sensors, such as the Microsoft Kinect sensor [138], together with the advancement in computation of 3-D depth data [112], have also resulted in a progression regarding methods for segmentation and detection of arbitrary 3-D objects. In the case a 3-D model of the object of interest is known in advance, e.g., plane, sphere, cylinder, etc., such objects can efficiently be detected and segmented with the use of an iterative model based segmentation algorithm, e.g., a Random Sample Consensus (RANSAC) algorithm [33]. For example, by utilizing such model based segmentation planar surfaces can be extracted in an early stage of the segmentation procedure. Arbitrary types of objects are, subsequently, segmented from the remaining 3-D points (i.e. points that are not part of prior estimated planar surfaces), based on estimated surface normals and through the use of 3-D clustering and segmentation techniques [111].

Searching 3-D point cloud data for nearest neighbors during clustering and segmentation of objects can, however, be a computationally costly process. A common practice is, therefore, to organize the point cloud data, e.g., by the use of a k-d search tree [9]. Contrarily, in case of already organized visual point cloud data (i.e., the organization of point cloud data is identical to the rows and columns of the imagery data from which the point cloud originates), the segmentation process can notably benefit from the organized structure by efficiently estimating 3-D surface normals based on integral images [51], and through the use of a connected component segmentation [131].

Deep Convolutional Perception

Recent technical advancements in computer graphics [93], which enable computational demanding processes to be executed in parallel on the graphics processing unit (GPU), have also enabled a trend within the field of machine learn-
2.1. SEMANTIC VISUAL PERCEPTION

ing that emphasizes the use of deep neural networks. This trend has further brought practical application domains for machine learning within a number of sub-topics of artificial intelligence, e.g., computer vision, speech recognition, etc. The benefit of deep learning applies, in particular, to the field of visual object recognition and classification, where deep neural networks permit artificial systems to learn intricate abstract patterns of objects.

Accompanied by the initial work presented by Krizhevsky et al., which demonstrated an outstanding performance with the use of a deep Convolutional Neural Network (CNN) on the 1,000 image classification challenge [110], there have followed a number of prominent succeeding deep learning approaches for object recognition ([40, 41, 104]), semantic segmentation ([61, 81]), and object classification ([49, 120]). A particularly prominent approach in the context of object classification is the GoogLeNet architecture [120]. In this work, Szegedy et al. introducing a 22 layers deep CNN for object classification and detection tasks. Following the work presented in [120], He et al. took the deep aspect one step further and introduced a substantially deeper architecture based on residual networks (ResNet) [49]. Another prominent approach for object localization and detection is the region based approach, called Regions with CNN features (R-CNN) [41]. Girshick et al. further suggested a domain-specific fine-tuning paradigm for the training of datasets with sparse samples, which can significantly boost the performance of domain-dependent semantic classification tasks.

2.1.2 Instance and Category Level of Perception

An autonomous agent that is concerned with real-world objects can, however, not exclusively rely upon aforementioned advancements in deep learning for detecting and recognizing objects, such as the methods introduced in Section 2.1.1. This issue arises from the same intricate abstract patterns of objects that are learned from several similar instances of an object category, which makes individual object instances of the same category indistinguishable. For example, it is impossible to make a distinction between two perceived instances of ‘mug’ objects if the system exclusively relies upon the classification labels given as results from a deep learning approach (trained for recognizing mugs among various other categories).

For an autonomous agent that handles real-world objects, it is, therefore, important to differentiate between the category- and instance level of objects. This distinction is a subject that has been acknowledged by the authors of the work behind the Open-Vocabulary system [44]. Guadarrama et al. introduced a combination of category- and instance level object recognition that is utilized for the purposes of both enriching the semantics of objects, as well as distinguishing between object instances. However, the Open-Vocabulary system is a top-down approach that emphasizes object recognition based on natural language queries, i.e., retrieving the best candidate object for a natural language
request. Throughout the work presented in this thesis, we, instead, approach the problem of category- and instance level object recognition bottom-up.

**Local Visual Key-point Features**

A technique for obtaining instance level object recognition is to rely upon firmly established local visual key-point features. Vector-based local visual features, such as SIFT [85] and SURF [6, 5], have successfully been used for over a decade for the purpose of detecting and identifying objects. However, vector-based visual features are computationally costly and can therefore become a bottleneck when used for vision applications (especially when used for real-time applications). As an alternative to vector-based features, a number of computationally efficient binary-valued visual features, such as BRIEF [15], ORB [108], BRISK [76] and FREAK [3], have instead been proposed.

Common for all local visual features (both vector-based and binary) is that feature point descriptors are computed for images patches around distinct image key-points. Feature descriptors are, therefore, often coupled with a key-point detector. Calonder et al. suggested the use of the CenSurE [1] or FAST [107] detector for detecting the key-points over which intensity difference tests over randomly selected pixel pairs are computed to represent images patches for the BRIEF descriptor [15]. A shortcoming of the BRIEF descriptor is, however, that it is sensitive to in-plane rotations and scaling. As an improvement, Rublee et al. suggested ORB (Oriented FAST and Rotated BRIEF) [108], which extended FAST by intensity centroids [106] for orientation, together with introducing a greedy search algorithm for selecting the most uncorrelated BRIEF binary tests (those of high variance) for improving rotation invariance. Inspired by the AGAST extension to FAST [86], Leutenegger et al. suggested a scale-space detector for identifying key-points across scale dimensions using a saliency criterion for their Binary Robust Invariant Scalable Key-points (BRISK) [76]. A binary string for each key-point is, thereupon, calculated over a rotation invariant sample pattern consisting of uniformly appropriately scaled concentric patches around each key-point.

Even though feature descriptors are often coupled with suggested key-point detectors, there is not a strict bound between descriptors and detectors. In reality, feature descriptors and key-point detectors can be combined arbitrarily. Similar to BRISK, but with focus on feature detectors, [3] presented their human retinal inspired Fast Retina Key-point (FREAK) [3], in which a cascade of binary strings is computed as the intensities over retinal sampling patterns of a key-point patch. Another recent publication focuses on the extraction of feature descriptors is the work on learning compact binary descriptors through the use of a CNN [80], which is trained with augmented key-point patches at different rotations.

In addition to feature descriptors extracted from 2-D image patches, there has lately been an increasing interest in extracting corresponding compact bi-
2.1. SEMANTIC VISUAL PERCEPTION

Binary feature descriptors from 3-D point cloud data. A first extension of the Signature of Histograms of Orientations (SHOT) descriptor [127, 128] was initially presented by Prakhya et al. In [100], the authors suggested a binary quantization that converts vector-based SHOT descriptors into an analogue binary vector (B-SHOT), which is reported to require 32 times less memory usage. Another prominent binary quantization method is the Binary Histogram of Distances (B-HoD) [59], which is a lightweight representation of the vector-based histogram of distances (HoD) descriptor [58]. Most recently, another novel approach was presented through the introduction of Binary Rotational Projection Histogram (BRoPH) descriptors [139], which generates binary descriptors directly from the point cloud data by transforming the description of the 3-D point cloud into series of binarized 2-D image patches.

Matching of Feature Descriptors

The main benefit of binary feature descriptors, compared to vector-based descriptors, is the support of fast brute-force matching (or linear search) by calculating the Hamming distance between features [47], which for binary strings is the count of the number of bits set to one in the result of XOR between two strings. This distance can be computed extremely efficiently with the use of the POPCNT instruction in x86_64 architectures of today.

Nonetheless, brute-force matching is only practical for smaller datasets. In the literature, the use of hashing techniques has been considered the best practice for improving matching of binary feature descriptors. The use of binary hash codes for large-scale image search has been studied in depth by Grauman and Fergus, who recently published an extended review on the topic [43]. Together with introduction of the ORB algorithm [108], the authors also suggested a Locality Sensitive Hashing (LSH) [39] approach as a nearest neighbor search strategy, where features were stored in different buckets over several hash tables. An alternative approach that proposes mapping features to much smaller binary codes, has been introduced by Salakhutdinov and Hinton through the concept of Semantic Hashing [115]. Semantic Hashing is, however, mainly practical for searching of nearest neighbors that differ by only a couple of bits. Another prominent work on the use of hashing techniques is Multi-Index Hashing, presented in [95], which presented results on datasets with binary strings of lengths up to 128 bits. In general, however, it is also the length of the binary strings and the amount of memory required to store all the buckets for a hashing approach that also limits such hashing approach.

In summary, there are currently two prominent approaches for dynamic and efficient matching of visual features that are directly (or with minimal alteration) applicable to binary feature descriptors:

- Clustered Search Trees: A common approach to speed-up matching when working with vector-based visual feature descriptors (SIFT [85] and SURF...
is through the use of clustered search trees. However, clustered search trees, suitable for vector-based features, are not mutually applicable for binary features. To address this problem, Muja and Lowe have presented their work on a similar clustered search tree algorithm for binary visual features [90], which is largely inspired by the GNAT algorithm [13].

Bag-of-Features Improved by Vectors of Locally Aggregated Descriptors (VLAD) and Vocabulary Adaption: Another prominent approach to speed-up matching of visual features is through the use Bag-of-Features (or Bag-of-Keypoints) [23]. An adaption of such approach for binary feature descriptors, and in particular ORB descriptors [108], has been reported by Grana et al. in [42]. Nonetheless, the main idea of clustering feature descriptors through $k$-majority clustering with Hamming distances (instead of traditional $k$-means clustering with Euclidean distances), as suggested by [42], is equally applicable for any kind of binary visual descriptors. A drawback of the use a Bag-of-Features approach is, however, that the vocabulary (as a result of training) needs further supervised training for categorization, e.g., through the use of Support Vector Machine (SVM) or a Naïve Bayes classifier, as suggested in [23].

A suggested approach to overcome this drawback of further training of the vocabulary is through Vectors of Aggregated Local image Descriptors (VLAD), first introduced in [54], and later revisited in [55]. VLAD can be considered a simplification of the Fisher kernel representations [99], where a compact code is aggregated with the use of the cluster centers as result of training the Bag-of-Features vocabulary.

2.1.3 Perception of Actions Applied to Objects

In the literature, the problem of recognizing object actions, i.e., the actions that are applied to objects, is typically addressed through the use of hand- and object tracking. A prominent work which uses such approach of combined hand- and object tracking to recognize kitchen activities is the work presented in [72], where the object recognition is utilized by an SVM classifier and where the (hand) action recognition is based on PCA features from 3-D hand trajectories and Bag-of-Words of snippets of trajectory gradients. However, learning to classify and recognize the perceived action is, by itself, a challenging task. The work presented in [2], address this problem through maintaining a representation of relations between objects at decisive time points during manipulation tasks. Aksoy et al. suggest a graph representation constructed from tracked image segments such that topological transitions in the graph can be stored (and learned) in a transition matrix called the Semantic Event Chain (SEC). Thus, they learn the semantics of object-action relations through observation. As an alternative to learning complex actions (or activities), Tenorth and Beetz intro-
duced a complementary knowledge processing system, called KnowRob\textsuperscript{1} [124], which uses common-sense information on a larger scale, such as information from the Internet, in order to reason about a perceived scene and infer possible actions (or action primitives) for robot manipulation tasks.

2.2 Semantic World Modeling

Deriving a meaningful world model from sensor input data is by far not a new concept [22]. Early reported works have, however, mainly been focused on the creation of meaningful map representations, typically based on ultrasonic range finder sensor data, which were useful for robot navigation tasks [21, 22]. A variety of works that address the problem of modeling and representing objects within a map model of the environment, which is useful for robot navigation, have further been presented through the concept of semantic mapping ([11, 66, 96, 101, 132, 137]).

The work presented in this thesis instead focuses on semantic representation and modeling which emphasize the objects themselves. This type of semantic modeling has not received the same impact in the literature as, for example, the concept of semantic mapping. However, there are a few notable publications (which are also directly or indirectly related to perceptual anchoring, and which further are presented in Section 2.4.2). An early presented solution for semantic world modeling was through the use of a Markov logic network, which enables probabilistic data association for formulating an object identity [10]. Succeeding the work presented by Blodow et al., was an approach that suggested the use of probabilistic multiple hypothesis anchoring [30], and which emphasized upon data association concerning semantic world modeling. Wong et al. discussed the limitations in the use of multi-target tracking and association and suggested, instead, a novel clustering-based approaches for semantic world modeling from partial views [136]. An alternative scene graph based world model was, instead, suggested by Blumenthal et al. in [12], which introduced a graph structure that enables tracking of dynamic objects, incorporates uncertainties, and allows for annotations by semantic tags. A graph-based world modeling approach has, likewise, been proposed for the recent works presented in [45, 109], which also introduces the feasibility to exploit contextual information during 3-D object recognition.

2.2.1 Tracking and Data Association

In the literature, semantic world modeling has commonly been presented synonymously with data association approaches [4]. The data association problem, which was motivated by the needs of tracking objects over time, addresses the task of estimating object states based on measurements from percepts, which

\textsuperscript{1}Further introduced in Section 2.3.2
in practice is analogous with associating uncertain measurements to known tracks. The work presented by Bar-Shalom and Fortmann provides a comprehensive overview of the functionalities of data association and object tracking, as well as an overview of greedy Nearest-Neighbor (NN) methods together with an approximate Bayesian filter, namely the Joint Probabilistic Data Association Filter (JPDAF) method [4]. However, JPDAF is a suboptimal approximation method, which assumes a fixed number of tracked targets.

An alternative approach for multi-target tracking and association is the Multiple Hypothesis Tracking (MHT) approach [103], which formulates associating hypotheses linked in a tree structure hierarchy such that the branches of possible tracks for a corresponding measurement can be explored recursively. However, the branching factor for this type of approach will, inevitably, grow exponentially with the number of measurements that are maintained. The issue of intractable branching of the tree-structured tracks of the possible hypotheses was widely discussed by Wong et al., which, instead, introduced a clustering approach based on Markov Chain Monte Carlo Data Association (MCMCDA) [97] for the work on data association for semantic world modeling from partial views [136].

2.2.2 Characteristics of World Modeling

The authors behind the work on data association for semantic world modeling [136], further outlined the fundamental differences between data association and semantic world modeling. Wong et al. argued that unlike target tracking, for which most data association algorithms are designed, semantic world modeling has three distinguishing domain characteristics, viz.:

- Objects can have attributes besides location, and hence are distinguishable from each other in general (which likely makes data association easier).

- Only a small region of the world is visible from any viewpoint. Most data association methods operate in regimes where all targets are sensed (possibly with noise/failure) at each time point.

- Most object states do not change over short periods of time.

For the work presented in this thesis, we endorse the same assumptions by focusing on the semantic world modeling problem (rather than the data association problem). In particular, we commend the assumption that "...objects can have attributes besides location, and hence are distinguishable from each other in general".
2.3 Available On-line Resources of Information

In this section, we highlight large-scale visual datasets that are publicly available on-line. In particularly, large datasets that can facilitate (deep) learning at scale in the context of computer vision and object recognition and detection, such as the *ImageNet* dataset [29], and the *PASCAL* visual object classes dataset [31].

### 2.3.1 Large Image Datasets

The *ImageNet* database [29] consists of 500 – 1.000 collected images for each of the *noun* synsets found in the *WordNet* lexical database [32, 88], and today contains of 14 million images for over more than 20.000 synsets. Another prominent resource is the *Tiny Images* database [130], which consist of 80 million 32x32 pixel images collected by performing web queries for the noun synsets found within WordNet. Because of the tight coupling between images and lexical noun synsets, maintained both in the ImageNet and the Tiny Images database, this type of database can advantageously be explored for both *semantic object detection*, and *semantic object classification* tasks. The semantic information about the corresponding imagery of an object can then further be enriched through the use of maintained WordNet synsets together with other lexical resources. A prominent lexical resource for this purpose is the *ConceptNet*² semantic network [118], which consists of common sense knowledge about concepts, as well as relations among concepts.

### 2.3.2 Robotics with Cloud Resources

In recent years, several comprehensive surveys have been presented [60, 114], which aims to outline current and further trends on the topic of cloud robotics and automation. The specific sub-topic of cloud robotics that we primary have explored together with the work presented in this thesis is within the domain of *big data* (such as the datasets described in the previous subsection). There are, nevertheless, related works on cloud robotics with similar objectives as the objectives present in this thesis, as well as complementary frameworks and techniques that provide general cloud capabilities in robotics:

*RoboEarth*³: An early prominent work on cloud robotics is the *RoboEarth* project [134], which is working towards the goal of designing a knowledge-based system that provides web and cloud services such that a simple robot can benefit from the experience of other robots, and hence, be transformed into an intelligent one. One example of such an outcome of the RoboEarth project is a semantic mapping system [105] which enables

²[http://conceptnet.io/](http://conceptnet.io/)
³[http://roboearth.ethz.ch/](http://roboearth.ethz.ch/)
robots to explore the environment and share semantic meaningful representations, such as the geometry of the scene and the location of objects with respect to the robot. The RoboEarth project has further resulted in several spin-off projects, such as *RobotHow* and *KnowRob*:

- **RoboHow**\(^4\): Aims towards enabling robots to competently perform everyday human-scale manipulation activities, e.g., *making pancakes* [7]. To extend the robots’ repertoire of knowledge, in order to perform such complex everyday manipulation tasks, new skills are acquired using web-enabled information [125, 126], together with experience-based learning [77], as well as learning by observing humans performing the task [52].

- **KnowRob**\(^5\): Another extension of the RoboEarth project is the *KnowRob* project [124], which focuses on combining knowledge representation and reasoning methods with knowledge acquisition techniques for grounding knowledge that is useful for a various robot operation task. Combined knowledge from different sources, which are represented in a semantic framework of integrated information, provides, for example, a knowledge-based of specifications for robot motions [123].

- **Robot Operating System (ROS)**\(^6\): The ROS eco-system [102], is a framework around a collection of tools, libraries, and conventions which have been developed for the purpose of simplifying robotic software development. This modularized eco-system is, accordingly, also providing support for cloud capabilities. In the area of cloud robotics, an especially notable extension of the ROS eco-system is the *rosbridge protocol* that supports Web client-server communication, and which is developed as a part of the *Robot Web Tools* project [129].

- **Robo Brain**\(^7\): A large-scale computational system that focuses on the learning aspect [116]. The Robo Brain project emphasizes, in particular, learning from publicly available Internet resources, computer simulations, as well as real-life robot trials [53], where learned knowledge is accumulated into a comprehensive and interconnected knowledge-base.

For a subset of the publications included in this thesis, we have explored similar cloud capabilities. For example, we have utilized the Web services that support access to the common-sense knowledge found in the *ConceptNet* semantic network [118], in pursuit of discovering the correlation between the action applied to an object and the object per se.

---

\(^4\)http://robohow.eu/project
\(^5\)http://www.knowrob.org/
\(^6\)http://www.ros.org/
\(^7\)http://robobrain.me/
2.4 Perceptual Anchoring

Over the past decade, there have been a few reported reviews on the topic of perceptual anchoring. An in-depth study was presented in the year 2013 together with the Ph.D. dissertation by Daoutis on the topic of knowledge-based perceptual anchoring [25]. Another notable review that partly covered the field of perceptual anchoring up to the year 2012 is the report presented by Coradeschi et al. in [20]. However, the focus of previously reported works was either on specific sub-topics, such as knowledge-based anchoring in the context of cognitive robots [25], or on a more general higher level, such as symbol grounding in general [20]. In this section, we continue where the previous reviews left off and present current trends up to date in the research areas that are directly, or indirectly, related to the concept of perceptual anchoring.

2.4.1 The Evolution of Perceptual Anchoring

Since the initial formal definition of perceptual anchoring, presented by Coradeschi and Saffiotti in [18], the definition has undergone several extensions and refinements. Loutfi et al. introduced an extension to bottom-up anchoring together with multiple and non-vision-based modalities [84]. The first connection to a knowledge representation and reasoning mechanism was presented by Loutfi et al. in [83], whereby the properties of anchored objects could be reasoned upon. An extension that considers large-scale knowledge bases, such as a Cyc knowledge base [75], together with common-sense reasoning was later presented by Daoutis et al. in [26]. Another notable work is the introduction of probabilistic multiple hypothesis anchoring by Elfring et al. in [30], such that tracking-based data association is used to maintain changes in anchored objects.

To outline related work on the topic of perceptual anchoring, and to illustrate how the concept of anchoring has evolved since the first introduction ([18]), consider Table 2.1 together with the following summary:

- **Anchoring via Conceptual Spaces** – to facilitate the mapping between perceptual sensor data and symbolic values, i.e. symbolic *grounding* of sensory data, Chella et al. introduced the idea of using *conceptual spaces* in combination with perceptual anchoring [16].

- **Anchoring with Multiple Sensor Modalities** – the traditional anchoring definition was presented with the assumption that perceptual sensor data was given by a vision-based sensor. In response, a redefinition to bottom-up anchoring with multiple and non-vision based modalities was presented by Loutfi et al. in [84].

- **Anchoring for Human-Robot Interaction (HRI)** – there have also been presented studies where anchoring is conditioned by a spoken dialogue
that is supplemented by gestures either by the robot [87], or by human users [73, 74]. Another prominent system that is able to learn grounded language models with minimum user interaction has been presented in [17], which relies upon semantic categories to group sensory data, and where the meaning of words, subsequently, are acquired through examples and with the use of a semantic clustering algorithm.

*Cooperative Anchoring* – a first extension for cooperative anchoring was presented by [70]. The idea of cooperative anchoring and sharing distributed information about anchored objects across multiple agents in multi-robot system setups has later been explored in depth though the Ph.D. thesis presented by LeBlanc in [71].

*Knowledge Representation and Reasoning (KR&R) in the Context of Anchoring* – a first connection to a knowledge representation and reasoning mechanism was presented by [83], whereby the properties of anchored objects could be reasoned upon. An extension that considers large-scale knowledge bases, such as a Cyc knowledge base [75], together with common-sense reasoning in a distributed systems was later presented by Daoutis et al. in [26, 27].

*Probabilistic Anchoring* – as an alternative to traditional anchoring, an initial work on perception and probabilistic anchoring was presented by [10]. The idea of probabilistic anchoring was later further explored by Elfring et al. in [30]. A subset of the publications that are included in this thesis are inspired by the idea of integrating a probabilistic object tracking into the concept of anchoring, and the topic of probabilistic anchoring is, therefore, further presented in following Section 2.4.2.

*Context-aware 3D Object Anchoring* – a further extension of anchoring for the means of maintaining a probabilistic world model has recently been explored in [45, 109], which introduced an adaptable graph-based world model that is derived from continuous observation from a 3-D object recognition method. The graph structure of the model is, subsequently, exploited by a *Probabilistic Graphical Model* (PGM) in order to leverage contextual information during recognition.

### 2.4.2 Probabilistic Perceptual Anchoring

The importance of data association in relation to perceptual anchoring was widely discussed by LeBlanc in his Ph.D. thesis on the topic of cooperative anchoring [71]. Around the same era, and as an alternative to traditional anchoring, early work on perceptual and probabilistic anchoring was presented by Blodow et al. in [10]. This work was also the first reported work where
### Table 2.1: Time-line over the evolution of perceptual anchoring.

<table>
<thead>
<tr>
<th>Year</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>Original definition [18].</td>
</tr>
<tr>
<td>2004</td>
<td>Anchoring via Conceptual Spaces [16].</td>
</tr>
<tr>
<td>2005</td>
<td>Bottom-up Anchoring with Multiple Sensor Modalities [84].</td>
</tr>
<tr>
<td>2005</td>
<td>Anchoring Conditioned by Spoken Dialogue for Human-Robot Interaction (HRI) [87].</td>
</tr>
<tr>
<td>2008</td>
<td>Cooperative Anchoring [70].</td>
</tr>
<tr>
<td>2008</td>
<td>Knowledge Representation and Reasoning (KR&amp;R) in the Context of Anchoring [83].</td>
</tr>
<tr>
<td>2010</td>
<td>Probabilistic Anchoring [10].</td>
</tr>
<tr>
<td>2012</td>
<td>Large-scale Knowledge Bases together with Common-sense Reasoning in Distributed Anchoring [26, 27].</td>
</tr>
<tr>
<td>2013</td>
<td>Multiple Hypothesis Anchoring [30].</td>
</tr>
<tr>
<td>2018</td>
<td>Context-aware 3D Object Anchoring [45].</td>
</tr>
</tbody>
</table>

The history of objects was considered by using a Markov logic network for probabilistic data association. The history of objects was, however, maintained as computationally complex scene instances and the approach was, therefore, mainly intended for solving the problem of anchoring and maintaining coherent instances of objects in object kidnapping scenarios, i.e., when an object disappears from the scene and later reappears in a different location.

The idea of probabilistic anchoring was, subsequently, further explored by Elfring et al. in [30], which introduced probabilistic multiple hypothesis anchoring. This approach utilizes Multiple Hypothesis Tracking (MHT) [103], in order to maintain changes in anchored objects and thus maintain an adaptable world model. Like the work presented by [30], we
Chapter 2. Related Work

acknowledge that a proper data association is essential for object anchoring, and we support the requirements identified by the authors for a changing and adaptable world modeling algorithm, which are formulated to include: 1) appropriate anchoring, 2) proper data association, 3) model-based object tracking, and 4) real-time execution. However, contrary to their work, we do not encourage a highly integrated approach that supports a tight coupling between object anchoring, probabilistic data association, and object tracking. Instead, the approach presented in this thesis maintains a loose coupling, which is motivated by the fact that a MHT procedures will inevitably suffer from the curse of dimensionality [8]. Hence, a probabilistic anchoring approach, as presented by [30], will further propagate the curse of dimensionality into the concept of anchoring.

The limitation in the use of MHT for world modeling has, likewise, been acknowledged in a recent publication on the topic of data association for semantic world modeling [136]. While this work inherits the same problem formulation, it substantially differs in approach. The authors discuss and exemplify issues related to the use of a tracking-based approach for world modeling, such as intractable branching of the tree-structured tracks of possible hypothesis, and instead, suggests a clustering approach based on Markov Chain Monte Carlo Data Association (MCMCDA) [97]. In the same work on semantic world modeling [136], Wong et al. also pointed out some characteristics that differentiate world modeling from target tracking, which we further acknowledge in this thesis and have also outlined in previously presented Section 2.2.2.

The use of perceptual anchoring for the purpose of modeling a probabilistic world representation has, most recently and in parallel with the work presented in this thesis, also been explored in [45, 109]. In analogy to the method presented in this dissertation, Günther et al. suggested the use of a Support Vector Machine (SVM), trained on samples of object pairs manually labeled as "same object"/"different object", in order to approximating the similarity between two objects. The assignment of candidate objects to existing anchors was, subsequently, calculated using prior similarity values and a Hungarian method [68]. However, the work presented in [45, 109], focuses spatial features and introduces a distinction between unary object features, e.g., the position of an object, and pairwise object features, e.g., the distance between two objects, in order to build a graph-based world model that, thereupon, can be exploited by a Probabilistic Graphical Model (PGM) [63] in order to leverage contextual relations between objects to support 3-D object recognition. The method, presented in this work, takes instead advantage of both visual and spatial features in order to directly approximating the anchoring matching problem.
Chapter 3
Rudimentaries of Perceptual Anchoring

The concept of perceptual anchoring has been defined as the "...process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical objects" [18, 19]. The need for anchoring has arisen in synergy with the identification of the anchoring problem, which has been defined as the "...problem of how to perform anchoring in an artificial system" [19], and which has emerged from the need for robotic planning systems to plan and execute actions involving objects. Having identified and recognized the anchoring problem, in the first place, has further paved the way for a formal definition of the components and techniques involved in the process of anchoring objects. Those components and techniques are presented in detail in this section.

3.1 Components of Anchoring

Perceptual anchoring was initially defined in [18]. Since its formalization, a variety of aspects related to anchoring have been examined in the literature [84, 83, 26]. Notable extensions include a redefinition by Loutfi et al. to include bottom-up anchoring with multiple and non-vision based modalities [84]. The first connection to a knowledge representation and reasoning mechanism was presented by Loutfi et al., whereby properties of objects (anchors) could be reasoned about [83]. A further examination of anchoring includes common-sense reasoning in a distributed system by Daoutis et al. [26]. In all these works, variations of anchoring have been presented with a number of common prerequisite components, including:

- A symbolic system including a set $\mathcal{X} = \{x_1, x_2, \ldots\}$ of individual symbols (variables and constants), and a set $\mathcal{P} = \{p_1, p_2, \ldots\}$ of predicate symbols.
A perceptual system including a set \( \Pi = \{\pi_1, \pi_2, \ldots\} \) of percepts, and a set \( \Phi = \{\phi_1, \phi_2, \ldots\} \) of attributes. A percept \( \pi_i \) is a structured collection of measurements assumed to originate from the same physical object; an attribute \( \phi_i \) is a measurable property of percepts with values in the domain \( D(\phi_i) \). Let \( D(\Phi) = \bigcup_{\phi \in \Phi} D(\phi) \).

A predicate grounding relation \( g \subseteq \mathcal{P} \times \Phi \times D(\Phi) \), which embodies the correspondence between (unary) predicates and values of measurable attributes. The relation \( g \) maps a certain predicate to compatible attribute values.

An overview of the anchoring components and their relations is exemplified in Example 1.

**Example 1.** Consider the captured camera image with segmented image regions, seen in Figure 3.1. Each segmented region corresponds to an individual percept, which is captured by the perceptual system, see Figure 3.1 – № 1. We denote the percepts \( \pi_1, \pi_2 \) and \( \pi_3 \), which correspond to the observed physical objects ‘ball’ (with smiley), ‘ball’ (tennis ball) and ‘mug’, respectively. For each percept, a number of attributes are, subsequently, measured, e.g. color, position, etc. One such attribute is a color attribute measured as a segmented normalized color histogram (in the HSV color space)\(^1\) over the masked area of percept \( \pi_2 \), illustrated in Figure 3.1 – № 2. For clarity, we denote the measured color attribute as attribute \( \phi_{\text{color}}^2 \), which has values in a domain that is equal to the \( n \) number of histogram bins. Finally, the predicate grounding relation \( g \), illustrated in Figure 3.1 – № 3, for the aforementioned color attribute can be seen as the encoded correspondence between specific peaks in the color histogram and certain predicate symbols\(^2\), e.g.:

\[
g(\text{black}, \text{color}, \arg\max_{i=1...n}(\phi_{\text{color}}^3, i)) \quad \text{iff} \quad i = 3
\]

Moreover, the following definitions allow for the characterization of objects regarding both symbolic and perceptual properties:

\( ^1\) For a complete description of measured attributes used in this thesis, see Section 4.2.3.
\( ^2\) For a detailed description of predicate grounding, see Section 4.3.1.
3.2 ANCHORING FUNCTIONALITIES

An anchor is, subsequently, an internal data structure $\alpha^t_x$, indexed by time $t$ and identified by a unique individual symbol $x$ (e.g., ‘mug-2’, ‘ball-1’, etc.), which encapsulates the correspondences between percepts and symbols that refer to the same physical object. A **symbolic description** of an anchor $\alpha^t_x$ denotes the list of predicates that are relevant to the perceptual identification of an object, while a **perceptual signature** gives the values of the measured attributes of the percept(s) encapsulated in an anchor $\alpha^t_x$.

3.2 Anchoring Functionalities

The anchoring process was initially introduced as a *top-down* process where anchors were managed by three abstract functionalities: **find**, **re-acquire** and **track**. However, the anchoring process and corresponding abstract functionalities have, as stated in Section 3.1, undergone several refinements and extensions since the initial definition [18]. The notation used for the framework presented in this thesis is primarily based on the notation presented together with the extensions to *bottom-up* anchoring with multiple and non-vision based modalities, by Loutfi et al. in [84]. This definition introduced an distinction between *bottom-up* sensor-driven acquisition and symbolically triggered *top-down* acquisition, such that the creation of anchors is managed by:

- **Acquire** – initiates a new anchor whenever a percept is received which currently does not match any existing anchor. It takes a percept $\phi$, and returns an anchor $\alpha^x$ defined at $t$ and undefined elsewhere. In *bottom-up* acquisition, a randomly generated symbol $x$ is attributed to the anchor. Furthermore, information about the object and its properties are included...
into the world model used by the planner. In this way the object can be reasoned about and acted upon.

- **Find** – takes a symbol x and a symbolic description and returns an anchor $\alpha^x$ defined at t (and possibly undefined elsewhere). It checks if existing anchors that have already been created by the acquire satisfy the symbolic description, and in that case, it selects one. Otherwise, it performs a similar check with existing percepts (in case the description does not satisfy the constraint of percepts considered by the acquire). If a matching percept is found an anchor is created.

Once an anchor has been created, it is important to update the perceptual signature as the anchored object is re-observed (both when the object is continuously observed, and when the object is re-observed after some time). Traditionally, the process of maintaining anchors was managed either by a symbolically triggered **track** functionality, or by a perceptually triggered **re-acquire** functionality. However, the notation presented by Loutfi et al. in [84], further introduced a combined functionality for maintenance of anchors:

- **Track** – the track functionality takes an anchor $\alpha^x$ defined for time $t - k$ and extends its definition to t. The track assures that the percept pointed to by the anchor is the most recent and adequate perceptual representation of the object. We consider that the signatures can be updated as well as replaced, but by preserving the anchor structure, we affirm the persistence of the object so that it can be used even when the object is out of view.
A graphical depiction of the core anchoring functionalities, as described above, is illustrated in Figure 3.2. In this thesis, we mainly target bottom-up anchoring and the creation of anchors through the acquire functionality. We will, therefore, focus on the problem of sensor-driven acquisition where the initial matching procedure (for determining if a received percept(s) matches an existing anchor or not) is based on the perceptual signature. Moreover, we have reintroduced a distinction between functionalities for maintenance of anchors, such that: 1) re-acquire is consistent with the maintenance functionality introduced by Loutfi et al. in [84], while 2) track is a functionality that is highly integrated with the state of the art in high-level probabilistic object tracking methods ([94]).
Chapter 4
Method: An Architecture for Perceptual Anchoring

"Comprehension follows perception."

— Philip K. Dick

In this chapter, we embrace the fact that "... objects can have attributes besides location, and hence are distinguishable from each other in general" [136], and present a novel anchoring framework that is used to maintain, based on object attribute values, a consistent notation of objects – both perceptually and symbolically. The presented architecture has evolved over years of studies, together with the publications included in this thesis. The aspects addressed in this chapter are, therefore, limited to recurrent core components of the proposed framework, while different variations related to specific applications are further presented in Chapters 5 & 6. The majority of the contributions related to those components of the proposed framework have, in particular, been introduced together with three of the publications that are included in this thesis:

In Paper I, we have explored the benefits of integrating a database solution into the concept of perceptual anchoring. By utilizing both publicly available on-line resources and statistical information about previously stored anchors, we can support and elaborate the grounding of symbolic values to corresponding perceptual information, as presented in Section 4.3.1. Moreover, we have presented how a database query interface that explores the strong coupling between semantic symbols and perceptual data, found within the concept of perceptual anchoring, can be used in order to reduce the search-space of possible anchor candidates to consider within the anchoring matching function (exemplified in Section 4.4.1).
### Question(s):

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

In **Paper II**, we have introduced an unconventional anchoring matching approach for determining how to anchor objects in a bottom-up anchoring fashion. In this context, two specific contributions have been introduced: 1) the theoretical procedure for comparing the attribute values of an unknown candidate object against the attribute values of previously stored anchored objects (outlined in Section 4.4.1), and 2) a method for combining attribute similarity distance values (given by the former procedure) and approximating the anchoring matching function by a learned model (introduced in Section 4.4.3). In addition, we have fine-tuned and integrated a Convolutional Neural Network (CNN) for the purpose of classifying objects at the perceptual level (described in Section 4.3.2).

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

In **Paper IV**, we have presented our work on facilitating instance level object recognition though the use of *binary descriptors*, extracted from visual key-point features (presented in Section 4.2.4). In an attempt to both efficiently represent and match such binary visual feature descriptors, we have also introduced a *summative approach*, called $f$-sum (presented in Section 4.4.1).

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 4.1 Architecture Overview

An overview of the anchoring framework which constitutes the backbone of the work presented in this thesis, can be seen in Figure 4.1. The system architecture of the framework is a modularized architecture consisting of several integrated nodes (individual programs), which utilizes the libraries and communication protocols that are available though the Robot Operating System (ROS)

\[1\]

[102]. In summary, the overall system architecture is composed of four central subsystems, namely: 1) a *perceptual system*; 2) a *symbolic system*; 3) an *anchor management system*; and 4) a *database system*. Moreover, both the *symbolic system* and the *anchor management system* of this architecture are fully integrated with a generic database interface, which provides an effortless communication with the subordinated *database system*, illustrated in Figure 4.1 – № 4.

\[1\]http://www.ros.org/
4.1. ARCHITECTURE OVERVIEW

Figure 4.1: A system architecture overview of the anchoring framework used for the work presented in this thesis. The system architecture is a modularized architecture consisting of four main subsystems, namely: 1) a perceptual system responsible for processing sensor data, segment objects of interest, and measure attribute values from prior segmented objects, 2) a symbolic system that establishes the percept-symbol correspondence for detected objects, 3) an anchor management system that creates and maintains anchors, and 4) a database system that facilitates long-term maintenance of anchors.

4.1.1 System Description Outline

Based on the framework overview, seen in Figure 4.1, the structure of the system description, presented in this section, is divided as follows:

- **Section 4.2:** presents the perceptual system with the purpose of segmenting and detecting objects on the perceptual level. This section further displays the various types of feature attribute values that are extracted and, subsequently, used to match and maintain anchored objects.
Section 4.3: covers the symbolic grounding of predicate symbols to perceptual data (as a result of the perceptual system). This subsection further presents how we utilize recent developments in deep learning techniques for classifying objects on a category level.

Section 4.4: presents the anchor management system with the two bottom-up anchoring functionalities acquire and re-acquire. In this subsection, we further introduce both the theoretical procedure for comparing the attribute values of an unknown candidate object against the attribute values of previously stored anchored objects. We also establish the formal use of a classification algorithm for combining attribute similarity distance values (given by the former procedure) to determine if a candidate object truly match a previously stored anchored objects.

4.1.2 Notes on Anchoring Notation

In addition to the definitions presented in Section 3.1, we will further define the following assumptions:

Assumption 1. An individual symbol $x$ is unique for an anchor altogether, and thereby also unique for all the percepts, attributes and symbols that are encapsulated in the anchor.

The individual symbol $x$ is, therefore, further used to index a particular percept, attribute or predicate symbol that belongs to an anchor $\alpha_x$.

Assumption 2. In similarity with the works presented in [83, 84], we assume that at an arbitrary time $t$, there can exist one or several types of percepts that originate from the same physical object.

For the notation used in this thesis, a superscript notation is, therefore, used to denote an individual type of percept, e.g., an arbitrary visual imagery percept is denoted $\pi_{\text{visual}}$, while $\pi_{\text{visual}}^x$ of an anchor $\alpha_x$ denotes a visual percept anchored in an anchor $\alpha_x$.

4.2 Perceptual System

An efficient perceptual processing pipeline is a necessity for retrieving and storing measurements over time of an object (which further is a requirement for anchoring that object to its symbolic representation). To support efficient detection and segmentation of objects at the perceptual level, the presented framework also relies upon publicly available core libraries and systems, such as the Point Cloud Library (PCL)$^2$, and the Open Computer Vision library (OpenCV)$^3$. An overview of the perceptual system and the perceptual processing pipeline is presented in Figure 4.1 – № 1.

$^2$http://pointclouds.org/
$^3$http://opencv.org/
4.2. PERCEPTUAL SYSTEM

4.2.1 Input Sensor Data

An essential precondition of perceptual anchoring is the sensory input data, which, subsequently, is processed in order to extract percepts, i.e., the structured collection of measurements that is assumed to originate from the same physical object. For the framework presented in this thesis, we presuppose visual sensor data as input. It should, however, be noted that the perceptual anchoring problem has previously been extended to cover other types of sensor modulates [84]. The work presented in this thesis is, nevertheless, purely based on visual sensor data. More specifically, the suggested anchoring framework is based on the rich information given by an RGB-D sensor. An example of such an RGB-D sensor is the Microsoft Kinect V2 sensor, which is illustrated in the upper part of Figure 4.1 – № 1. Hence, both the RGB color information and the per-pixel depth information are available for each frame, which subsequently is projected into dense 3-D point cloud data. To facilitate the integration and to establish the connection between the ROS environment and the Kinect RGB-D sensor, we have used the ROS-Kinect2 bridge [135], developed at the University Bremen, for the majority of the works presented in this thesis.

4.2.2 Object Segmentation

The initial step of the suggested processing pipeline is an object segmentation procedure, which is performed with the purpose of detecting the objects of interest in the scene. This object segmentation procedure is based on point cloud data, which is provided as input data by the RGB-D sensor. More specifically, the used segmentation procedure presumes an organized structure of the point cloud data (i.e., the organization of point cloud data is identical to the rows and columns of the imagery data), in order to efficiently segment 3-D objects via the following steps:

- Estimate 3-D surface normals based on integral images [51]. This function uses the algorithm for calculating average 3-D gradients over six integral images, where the horizontal and vertical 3-D gradients are used to compute the normal as the cross-product between two gradients.

- Planar segmentation based on the calculated surface normals.

- Object segmentation through clustering of the remaining 3-D points (i.e. points which are not intersecting detected surface planes). This segmentation uses a connected component segmentation, presented by [131], where a Euclidean comparison function is used to connect the points that constitute the cloud cluster of an individual object.

The resulting output of the object segmentation procedure is \( \mathbf{m} \) point cloud clusters (where \( \mathbf{m} \) varies between frames). An example of segmented objects is
Figure 4.2: An example of segmented objects. Each segmented object is represented by the outer contour and the 3-D points of the corresponding point cloud cluster (projected from 3-D point cloud data to 2-D imagery data).

illustrated in Figure 4.2. For consistency with the definition of anchoring, we will denote segmented clusters as percepts: \( \{ \pi_{1}^{\text{spatial}}, \pi_{2}^{\text{spatial}}, \ldots, \pi_{m}^{\text{spatial}} \} \), which each corresponds to the 3-D point cloud data of an individual object.

After the segmentation of the point cloud clusters, the same RGB-D data is also used for segmenting the corresponding visual 2-D imagery data of each detected and segmented object. This image segmentation process is entirely based on the prior point cloud clusters and a projection between the 3-D point cloud frame and the 2-D visual RGB frame of the RGB-D sensor. Also, we will here denote visual data as percepts \( \{ \pi_{1}^{\text{visual}}, \pi_{2}^{\text{visual}}, \ldots, \pi_{m}^{\text{visual}} \} \), which each corresponds to the visual 2-D imagery data of a segmented object.

### 4.2.3 Feature Extraction

Segmented percepts are, subsequently, forwarded to a feature extraction procedure. The first step of this procedure is to extract a position attribute \( \phi_{y}^{\text{pos}} \in \mathbb{R}^{3} \), which is measured as the point at the geometrical center of each segmented percept \( \pi_{y}^{\text{spatial}} \), where \( \pi_{y=1,2,...,m}^{\text{spatial}} \in \{ \pi_{1}^{\text{spatial}}, \pi_{2}^{\text{spatial}}, \ldots, \pi_{m}^{\text{spatial}} \} \). From the same percept, \( \pi_{y}^{\text{spatial}} \), is further a size attribute \( \phi_{y}^{\text{size}} \in \mathbb{R}^{3} \) measured as the 3-D bounding box around each percept \( \pi_{y}^{\text{spatial}} \).

Correspondingly, a color attribute \( \phi_{y}^{\text{color}} \) is extracted for each visual percept \( \pi_{y}^{\text{visual}} \), which is measured as a color histogram (in the HSV color space) over the 2-D imagery area.
4.2.4 Extract Binary Descriptors

In order to identify an object on the instance level, a distinct signature of each object must be extracted. A prominent approach to identifying objects on an instance level, which further has been addressed in Paper IV, is through the use of distinct visual key-point features that, subsequently, are encoded as feature descriptors. Visual key-point features, in relation to anchoring, has been researched in previously presented works on anchoring [26, 27, 28], where vector-based SIFT key-point features [85] were extracted and used for training a Bag-of-Words (BoW) model. This model was later used for identifying an encountered object in a household experimental setup with a moving robot. However, vector-based visual key-point features, such as SIFT [85], are computationally demanding to extract and match, and will, therefore, become a computational bottleneck if used in a larger scale.

As an alternative to vector-based visual features, a number of computationally efficient binary-valued visual features have, instead, been proposed, such as BRIEF [15], ORB [108], BRISK [76] and FREAK [3]. In Paper IV & Paper V, we have explored the use of such binary-valued visual features in order to identify instances of objects. In relation to anchoring, a binary descriptor attribute $\phi^{\text{desc}}_y$ (based on visual key-point features extracted from the corresponding visual percept $\pi^{\text{visual}}_y$), is derived through the following steps:

1. Detect distinct image key-points, e.g., through the use of the ORB key-point detector (Oriented FAST and Rotated BRIEF) [108], which uses the FAST algorithm [107], extended by intensity centroids [106], for the orientation and a greedy search algorithm to improve the rotation invariance of the BRIEF detector [15].

2. Extract binary feature descriptors, e.g., by the use of the FREAK (Fast Retina Keypoint) algorithm [3], in which a cascade of binary strings is computed as the intensities over retinal sampling patterns of a key-point patch.

The resulting descriptor attribute $\phi^{\text{desc}}_y$ is then a set of binary strings (one binary string extracted for each detected visual key-point), where each binary string is of a fixed length, e.g., 64 B. An exemplification of detected key-point features is illustrated in Figure 4.3.

4.2.5 Used Attributes in Relation to Publications

It is worth noting that all extracted and measured attributes, as outlined in this Section 4.2 (and partly presented in following Section 4.3), have not been used in all publications that are included in this thesis. Different combinations of attributes, and subsets thereof, have been used for various purposes for particular publications. For a complete summary of which attributes that have been used in which publication, we refer to Table 4.1.
Figure 4.3: An illustration of distinct ORB key-point features. Each detected key-point is represented by a circle (in the middle column). The diameter of each circle represents the size of the adjacent area of the key-point, while the line within each circle represents the orientation of the key-point. The correlation between the detected key-point features, between two scenes, is further illustrated in the rightmost part figure.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>$\phi^{\text{class}}$</th>
<th>$\phi^{\text{color}}$</th>
<th>$\phi^{\text{desc}}$</th>
<th>$\phi^{\text{pos}}$</th>
<th>$\phi^{\text{size}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paper I</td>
<td>✓</td>
<td>✓</td>
<td>□</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Paper II</td>
<td>✓</td>
<td>✓</td>
<td>□</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Paper III</td>
<td>✓</td>
<td>□</td>
<td>□</td>
<td>✓</td>
<td>□</td>
</tr>
<tr>
<td>Paper IV</td>
<td>□</td>
<td>□</td>
<td>✓</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>Paper V</td>
<td>□</td>
<td>✓</td>
<td>✓</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

Table 4.1: A summary of attributes used for various presented publications.
4.3 Symbolic System

Operating between the perceptual system and the anchor management system, is a *symbolic system*, which consists of both a *predicate grounding* procedure and a *object classification* procedure, illustrated in Figure 4.1 – № 2.

4.3.1 Predicate Grounding

Establishing the percept-symbol correspondence is a fundamental prerequisite for the concept of object anchoring, as the correspondence between percept-symbol further permits the exploration of symbolic values in order to search for corresponding perceptual information. In the work presented in Paper I, we emphasize the importance of establishing a reliable percept-symbol correspondence by introducing an improved symbolic system that practically handles the grounding of predicate symbols to corresponding attribute measurements.

The *predicate grounding* procedure is responsible for grounding a measured attribute $\phi_y$ (of the collection $\Phi_y$, that originates from the same physical object) to a predicate grounding symbol $p_y$, e.g. a certain peak in a color histogram, measured as a $\phi_y^{\text{color}}$ attribute, is grounded to the symbol ‘red’. More specifically, for the grounding of color predicates, $p_y^{\text{color}}$, we rely upon color theory and that a color tone is expressed as a mixture of the base color mixed with either *tint* (mixture of white) or *shade* (mixture of black) [89]. An elemental color term, e.g. ‘red’, can therefore be defined as a set of color tones of different color shades of the (base) color term, i.e. *shades of red*. This information is widely available through on-line resources, e.g. Wikipedia\(^4\). Given the sets of different color shades of 13 elemental color terms: \{‘white’, ‘gray’, ‘black’, ‘magenta’, ‘pink’, ‘red’, ‘brown’, ‘orange’, ‘yellow’, ‘green’, ‘cyan’, ‘blue’, ‘violet’\} $\in \mathcal{P}$, our approach for grounding $p_y^{\text{color}}$ predicates utilizes a *Support Vector Machine* (SVM) classifier [14], which was trained with sets of HSV color values for each color shade. Hence, the color predicate grounding is given by the average prediction of the $n$ most dominant peaks of the HSV color histogram that compose a *color attribute*, $\phi_y^{\text{color}}$, as illustrated in Figure 4.4.

Moreover, consider that the goal is to ground size predicates: \{‘tiny’, ‘small’, ‘medium’, ‘large’, ‘huge’\} $\in \mathcal{P}$, to a *size attribute*, $\phi_y^{\text{size}}$, which expresses the volume of the 3-D bounding box around a segmented object. Since the size relations among objects depend upon both the objects per se and the specific domain in which the objects are found, this grounding process can not rely upon a pre-trained model for the mapping between *size predicate symbols* and measured *size attributes*. For example, a ‘spoon’ found in a *kitchen* domain can be considered a ‘small’ object compared to a ‘plate’, while the same ‘plate’ can be considered a ‘tiny’ object compared to a ‘couch’ in a *household* domain. Hence, for grounding a *size predicate*, $p_y^{\text{size}}$, we need to

\(^4\)https://en.wikipedia.org/wiki/List_of_colors_by_shade
Figure 4.4: The procedure for grounding color predicates. Based on a segmented visual percept (including a bit binary image mask), a color histogram in HSV color space is first calculated as a color attribute. Next, dominant peaks of this color histogram are classified by an SVM classifier that has been trained by examples of different shades of elemental colors. Finally, the color predicate symbol is given as the best prediction of the SVM classifier.

consider the data of all previously perceived objects. Assuming a normal distribution of the volumes of all previously perceived objects, the approach for grounding size predicates is based on the standard deviation and the z-score, as exemplified in Fig. 4.5, and where the volume of the 3-D bounding boxes of all previously anchored objects are accessed through the use of a database query interface, illustrated by outgoing arrows from Figure 4.1 – № 4.

Figure 4.5: A size predicate is grounded based on the volume of the size attribute, and the distribution of volumes of all previously perceived objects.

4.3.2 Object Classification

The final procedure of the symbolic system is through an object classification procedure, which is initiated in addition to the traditional predicate grounding procedure with the goal of symbolically associating a category label with each object. For this classification procedure, we have exploited recent advancements in deep learning for image classification, which was facilitated by the integra-
4.4 Anchor Management System

The last subsystem of the proposed framework architecture is the anchor management system, illustrated in Figure 4.1 – № 3, which is responsible for the administration of the fundamentals of anchoring, i.e., creating and maintaining anchors. Based on the result of an initial matching function, an anchor
(stored in an anchor-space) is either created or maintained through two core functionalities: acquire or re-acquire. This system further utilizes a reinstated track functionality, which has been re-introduced together with the integration of a object tracking procedure. This integration of object tracking into anchoring has been explored both on the perceptual level and on the higher symbolic level – further addressed in Chapter 5.

4.4.1 Matching Function

The entry point into the anchor management system is via a matching function. This matching function follows a bottom-up approach to anchoring, described in [84], where the system constantly receives candidate objects and invokes a number of different matching algorithms (one matching algorithm for each measured attribute) in order to match the perceptual signature of an unknown candidate object against the signature of an existing anchor. More specifically, a set of attributes $\Phi_y$ of an unknown candidate object is compared against the set of attributes $\Phi_x$ of an existing anchor, $\alpha_x$. The combined result of each invoked matching algorithm, subsequently, determines if an anchored object has previously been perceived or not.

In detail, a classification attribute $\phi_{\text{class}}^y$ and symbol $p_{\text{class}}^y$ of an unknown candidate object is compared against the classification attribute and symbol of a previously stored anchor, $\alpha_x^x$, according to:

$$d_{\text{class}}(\phi_{\text{class}}^x, \phi_{\text{class}}^y) = \begin{cases} \exp \left( -\frac{|\phi_{\text{class}}^x - \phi_{\text{class}}^y|}{\phi_{\text{class}}^x + \phi_{\text{class}}^y} \right) & \text{if } p_{\text{class}}^x \equiv p_{\text{class}}^y \\ 0 & \text{else} \end{cases} \quad (4.1)$$

The $d_{\text{class}}$ can be interpreted as the exponentially decaying relative $L^1$-distance between the two attribute values $\phi_{\text{class}}^x$ and $\phi_{\text{class}}^y$. This means that we exponentially penalize the distance between two objects in the class attribute space.

Moreover, the color histogram of a color attribute $\phi_{\text{color}}^y$ of a candidate object is compared (assuming normalized color histograms) according to the color correlation:

$$d_{\text{color}}(\phi_{\text{color}}^x, \phi_{\text{color}}^y) = \frac{1}{2} \left( 1 + \frac{\sum_{i=1}^{n} (\phi_{\text{color}}^x_i - \mu_x)(\phi_{\text{color}}^y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n} (\phi_{\text{color}}^x_i - \mu_x)^2 \sum_{i=1}^{n} (\phi_{\text{color}}^y_i - \mu_y)^2}} \right) \quad (4.2)$$
Where \( n \) is the number of histogram bins, the index \( i \) gives the \( i \)-th histogram bin value of \( \phi^\text{color}_x \) and \( \phi^\text{color}_y \) (respectively), and \( \mu_x \) and \( \mu_y \) are the color mean value of each histogram, given according to:

\[
\mu_x = \frac{1}{n} \sum_{i=1}^{n} \phi^\text{color}_x(i), \quad \mu_y = \frac{1}{n} \sum_{i=1}^{n} \phi^\text{color}_y(i)
\]

The distance between a position attribute \( \phi^\text{pos}_y \), and the position \( \phi^\text{pos}_x \) of a previously stored anchor \( \alpha^x \), is calculated according to the \( L^2 \)-distance (in 3-D spatial space). Inspired by the work presented by [10], this distance is then mapped to a normalized similarity distance according to:

\[
d^\text{pos}_{x,y}(\phi^\text{pos}_x, \phi^\text{pos}_y) = e^{-L^2(\phi^\text{pos}_x, \phi^\text{pos}_y)}
\]  \( (4.3) \)

Furthermore, the size attribute \( \phi^\text{size}_y \) of a candidate object is compared according to the generalized Jaccard similarity distance (for the bounding boxes in 3-D space):

\[
d^\text{size}_{x,y}(\phi^\text{size}_x, \phi^\text{size}_y) = \frac{\sum_{i=1}^{n} \min(\phi^\text{size}_x(i), \phi^\text{size}_y(i))}{\sum_{i=1}^{n} \max(\phi^\text{size}_x(i), \phi^\text{size}_y(i))}
\]  \( (4.4) \)

Motivated by the importance of the time \( t \) within the concept of anchoring, the difference in time since last recorded observation of a previously stored anchor \( \alpha^x \), defined at \( t - k \), is finally mapped to a similar normalized distance according to:

\[
d^\text{time}_{x,y}(t, t - k_x) = \frac{2}{1 + e^{t-(t-k_x)}} = \frac{2}{1 + e^{k_x}}
\]  \( (4.5) \)

The difference in time, \( d^\text{time}_{x,y} \), can also be interpreted as a numerical description of the characteristic that "...most object states do not change over short periods of time", as discussed by [136].

All given similarity distance values (Eqs. (4.1) to (4.5)) are, consequently, given in the interval \([0.0, 1.0]\), and all distance values are, therefore, also commensurable, i.e., none of the similarity distance values have a greater (or a lesser) influence on the combined result.

**f-sum: A Summative Approach for Matching of Binary Descriptors**

In an attempt to improve the matching of binary descriptors, we have in Paper IV presented a summative approach for fast matching of binary descriptors, called f-sum. Given a binary descriptor attribute, \( \phi^\text{desc}_y \), which consists of \( n \) binary strings of \( l \) length, the principal objective with the suggested f-sum approach is, subsequently, to create a 2-D array which maintains a frequency count that represents how often a particular value occurs for each byte, for each binary descriptor string. To exemplify, consider that we are given a binary
CHAPTER 4. AN ARCHITECTURE FOR PERCEPTUAL ANCHORING

descriptor attribute of dimension \( n \times l \), where \( n \) is the number of binary strings, and \( l \) is the length in bytes of each binary string, e.g., 32 or 64. The resulting 2-D array \( f\text{-sum}^{2d}_{y} \), will then have a dimension of \( p \times l \), where \( p \) is maximum value of a byte, i.e., 255. This frequency count is further a weighted frequency count depending on the size of \( n \) such that the larger the \( n \) the less significance each individual byte has on the frequency occurrence. Thus, the frequency occurrence is multiplied by \( 1/((l \times n)) \) for each cell, \( f\text{-sum}^{2d}_{y,i,j} \), so that each cell value is given in the interval \([0.0, 1.0]\).

In the context of the anchor matching, comparing a summarized 2-D array \( f\text{-sum}^{2d}_{y} \) (which has been summarized based on the binary descriptor attribute, \( \phi_{\text{desc}}^{y} \), extracted for an unknown candidate object), against the summarized 2-D array \( f\text{-sum}^{2d}_{x} \) maintained in an anchor \( \alpha^{x}_{t} \), is thereupon done by measuring the \( L^{1} \)-distance between both arrays, according to:

\[
d^{\text{desc}}_{x,y}(f\text{-sum}^{2d}_{x}, f\text{-sum}^{2d}_{y}) = \sum_{i=0}^{p} \sum_{j=0}^{l} |f\text{-sum}^{2d}_{x,i,j} - f\text{-sum}^{2d}_{y,i,j}| \tag{4.6}
\]

Reduced Anchoring Search-Space through Database Queries

As the anchoring problem moves towards real-world situated scenarios, and away from "toy examples", a more substantial amount of anchors needs to be maintained and managed. This shift of scenario domain has introduced new areas of concern. An issue that is particularly evident in the context of matching anchors at scale in a bottom-up fashion, is the problem of increasing complexity within the anchoring matching function, which must match the perceptual signature of an unknown anchor against the signature of all previously created and maintained anchors, for every time instance.

In Paper I, we have addressed this issue by presenting our work on a database-centric approach to perceptual anchoring. The main novelty, introduced in this context, is a database query interface that explores the strong coupling between semantic symbols and perceptual data, found within the concept of perceptual anchoring, in order to reduce the search-space of possible anchor candidates to consider within the anchoring matching function. To practically demonstrate our approach, we refer to the grounding of predicate symbols and the object category classification, as presented in Section 4.3. Consider that we further divide the total set of predicate symbols, \( \mathcal{P} \), into the following attribute specific subsets of symbols:

- \( \mathcal{P}_{\text{class}} = \{ 'mug', 'spoon', \ldots 'tomato' \} \);
- \( \mathcal{P}_{\text{color}} = \{ 'white', 'gray', \ldots 'violet' \} \);
- \( \mathcal{P}_{\text{size}} = \{ 'tiny', 'small', \ldots 'huge' \} \).
The size of each subset is, consequently: \( n_P^{\text{class}} = 101 \), \( n_P^{\text{color}} = 13 \), and \( n_P^{\text{size}} = 5 \), respectively. By utilizing this divide set of predicate symbols, and by formulating database queries that are composed of the conjunction of symbols from each attribute specific subset of symbols, e.g., \{\&\&: [{"p.class": "mug"}, {"p.color": "white"}, {"p.size": "medium"}]\}, we can in theory reduce the anchoring search-space to:

\[
\frac{1}{n_P^{\text{class}}} \times \frac{1}{n_P^{\text{color}}} \times \frac{1}{n_P^{\text{size}}} = \frac{1}{101} \times \frac{1}{13} \times \frac{1}{5} = 0.015\% 
\]

This theory is, however, only valid for the assumption that all the attributes and corresponding predicate symbols of all previously stored anchors are equally distributed. In reality, this is rarely the case. For example, a scenario might contain several ‘mugs’ of the same ‘white’ color and of the same ‘medium’ size. In this case, a database query can only provide a subset of possible matching anchor candidates, while individual instances of ‘mugs’ are only distinguishable by their attribute values, e.g., they are different in positions. In order to truly match unknown candidate objects against previously stored anchors, we must, therefore, rely upon the notation presented in Eqs. (4.1) to (4.6). This problem of ambiguous results, which arises from matching anchors based on symbolic values, is further addressed in Chapter 6.

### 4.4.2 Creating and Maintaining Anchors

The anchoring matching function, described in Section 4.4.1, is for matching the perceptual signature of an unknown candidate object against the perceptual signature of an existing anchor, \( \alpha^x \). This matching process is, subsequently, repeated for all previously stored anchors \( \alpha^{x|x \in X} \), and based on the the overall result of the matching function an anchor is created or maintained (in an anchor-space that is comprised by a database solution, illustrated in Figure 4.1 – № 4), through one of the two principle functionalities:

- **Acquire** – initiates a new anchor whenever a candidate object is received that does not match any existing anchor \( \alpha^x \). This functionality defines a structure \( \alpha^x_t \), indexed by time \( t \) and identified by a unique identifier \( x \), which encapsulates and stores all perceptual and symbolic data of the candidate object.

- **Re-acquire** – extends the definition of a matching anchor \( \alpha^x \) from time \( t - k \) to time \( t \). This functionality assures that the percepts pointed to by the anchor are the most recent and adequate perceptual representation of the object.

As suggested by Loutfi et al. [84], in bottom-up acquisition of anchors, a unique individual symbol \( x \) should further randomly be generated and attributed to the anchor. To generate a unique individual symbol for each anchor,
we have enhanced the traditional `acquire` functionality by utilizing the deep learning classifier (presented in Section 4.3.2), such that a unique symbol \( x \) is generated based on the classification symbol \( p_{\text{class}} \), e.g., for an object classified as a ‘cup’, a corresponding unique symbol could be generated as \( x = \text{‘cup-4’} \).

An example of anchored objects together with a unique symbol, \( x \), which has been generated based on the classification symbol, \( p_{\text{class}} \), for each anchored object, can be seen in Figure 4.7.

![Figure 4.7](image_url)

**Figure 4.7**: An example of anchored objects together with a unique symbol, \( x \), generated and attributed to each anchored object.

However, combining all similarity distance values, given by Eqs. (4.1) to (4.6), in order to determine whether a candidate object has previously been perceived (or not), is not a trivial task. Especially not in the context of bottom-up anchoring in real-world situated scenarios with unlimited possibilities of objects together with continuous distance values.

To address the issue of interpreting the result of the initial anchoring matching function, various approaches have been studied together with the different publications included in this thesis. The work presented in Paper IV and Paper V, was mainly evaluated though the use of a simulated anchor-space together with datasets consisting of 2-D images of known objects. Hence, the anchoring functionalities were determined in a traditional “winner-takes-all” manner through the use of Eq. (4.6). A rule-based approach together with similarity distance values according to Eqs. (4.1) and (4.3) was, instead, used for the work presented in Paper III. The distinction between the `acquire` of a new anchor or the `re-acquire` of an existing anchor was, consequently, decided by two threshold values, which were determined through trial-and-error and set to \( \theta_{\text{class}} = 0.3 \) and \( \theta_{\text{pos}} = 0.95 \), i.e., if \( d_{x,y}^{\text{class}} > \theta_{\text{class}} \) or \( d_{x,y}^{\text{pos}} > \theta_{\text{pos}} \) for an existing anchor \( \alpha_x \) according to Eqs. (4.1) and (4.3), then the `re-acquire` functionality was initiated.
For the work presented in Paper II, we have further shed some light upon this issue of combining all attribute similarity distance values to determine whether a candidate object has previously been perceived (or not), and introduced the idea of learning to combine the distance values and, consequently, learning to approximate the anchoring matching problem – further addressed in Section 4.4.3. In Paper I, we have, likewise, explored the idea of learning the anchoring matching function. However, in Paper I, we have, instead, approached the problem from the stance where sparse training data is available.

### 4.4.3 Learning the Anchoring Functionalities

A benefit of using perceptual anchoring is that the percepts pointed to by the anchor are the most recent and adequate perceptual representation of an object. For the evaluations presented in both Paper I & Paper II, we have exploited this updated perceptual representation, found within an anchor, in order to collect a human annotated ground truth dataset. The collected dataset was, subsequently, used in order to train different learning algorithms to initiate the appropriate anchoring functionality for different situations.

#### Data Collection

The data collection, for presented evaluations, was conducted through the use of an introduced human-annotation interface, which was queued with segmented perceptual sensor data given by the perceptual processing pipeline (presented in Section 4.2). By utilizing this interface, all data about unknown candidate objects, together with the perceptual data of possible matching existing anchored objects, could be presented and visualized for the human user. The user could, thereby, provide feedback about the action that the human counterpart would consider to be the most appropriate anchoring action for each presented candidate object (i.e., acquire a new anchor for a queued object, or re-acquire an existing anchor). The procedure for collecting our ground truth data is described and exemplified in Figure 4.8.

Behind the scene of the proposed interface the data that, in reality, was collected and stored was similarity distance values, provided by Eqs. (4.1) to (4.5). Together with each set of similarity distance values (as result of comparing the attributes of an unknown candidate object against the attributes of an existing anchored object), an annotated label was also stored (which indicated whether the user considered an existing object a matching object, or not). By collecting data based on the human judgment and the feedback provided by the user, we were further able to confront the user with typical problematic anchoring scenarios and gather samples for such problematic scenarios, e.g., ambiguous situations where an identical (but physically different) instance of an object was introduced while the similar counterpart was still observed. It should, however,
Figure 4.8: A depiction of the proposed human-annotation interface that was used in order to collect ground truth data of anchored objects. In conjunction with changes in the scene, as illustrated by № 1, the human user has the possibility to freeze the execution of the framework and select segmented candidate objects, e.g., the moved ‘ball’ as illustrated in № 2. The framework is responding by displaying an updated representation of a number of already anchored objects, shown in № 3, which perceptually and symbolically best match the selected object (e.g., select all already anchored ‘yellow balls’). The user can then provide positive feedback about a matching anchored object (by selecting the representation of the matching anchored object), or negative feedback (by clicking somewhere else on the screen).

be noted that the user referred to in this section has mainly been the author of this thesis (though with some "user" support from senior colleagues).

Evaluated Supervised Classification Algorithms

Through the use of the proposed human-annotation interface, we were able to collect a dataset of a total of 5400 samples for the evaluation presented in Paper II. During the data collection, several typical problematic anchoring scenarios (e.g., scenarios where ambiguous objects are introduced in the scene, existing objects that are disappearing and reappearing in the scene, etc.), were executed in order to cover a broad range of different situations. Moreover, the data collection was conducted on several occasions to capture changes in the environmental conditions, e.g., changes in lighting conditions.

Given the collected data, which was comprised of sets of similarity distance values together corresponding annotated labels, our approach for learning how to correctly anchor objects (and thereby learn to invoke the correct anchoring functionality – acquire or re-acquire), was through the evaluation of different supervised classification algorithms. More specifically, for this evaluation we used and trained the following classification algorithms (parameters used for each classifier were, initially, determined through trial-and-error):

- **Support Vector Machine (SVM)** [14], with ν-Support Vectors (trained with ν = 0.1), and with a Histogram intersection kernel function
- **Multi Layer Perceptron (MLP)**, with back-propagation training, two hidden layers and a layer configuration, according to: 5 – 10 – 15 – 2
4.4. ANCHOR MANAGEMENT SYSTEM

- **k-Nearest Neighbor** (k-NN), trained and tested with $k = 3$

- **Normal Bayes Classifier** (Bayes) [36]

The collected dataset was randomly divided 70/30 into training/test samples, giving us a total of 3780 training samples and 1620 testing samples. The resulting average classification accuracy, together with F1 score, for each trained classifier can be seen in Figure 4.9.

![Figure 4.9: Resulting average classification accuracy together with F1 score for each trained and tested model for our approach to learning the anchoring functionalities through supervised classification.](image)

Examining the results, presented in Figure 4.9, it is noticeable that the best average classification accuracy of 96.4% was achieved by the use of a SVM classifier. The highest average F1 score (for a true match) of 94.4% was, likewise, achieved with the same SVM classifier. It should, however, also be noted that the differences in classification accuracy between the MLP classifier and the SVM classifier are close to insignificant (only 0.2%). Nevertheless, the best trained resulting SVM model was, subsequently, formally integrated as a part of the initial matching function such that the predicted result of the SVM model was used to determine if an unknown candidate object matches an existing anchor (i.e., if the anchor should be re-acquired as an existing matching anchor), or if no current anchors match the candidate object (i.e., if a new anchor should be acquired).

**Evaluated Semi-Supervised Learning Algorithms**

Motivated by the need for an autonomous agent to adapt to an unknown environment, e.g., in case the agent change domain of operation, a similar learning approach was further explored for the evaluations presented in Paper I. For this evaluation, we collected a similar dataset through the use of the same human-annotation interface. However, in this case, a limited dataset of only 602 samples of annotated data was collected, and our approach for learning how to correctly anchor objects was, consequently, through the use of different semi-supervised classification algorithms. More specifically, we trained and
validated the following variations of classification through Support Vector Machine (SVM) [14] algorithms (where a Radial Basis Function (RBF) was, for consistency, used as kernel function for all validated algorithms):

- Supervised Support Vector Machine (SVM) classification (used for comparison)
- Sparse Quasi-Newton optimization for Semi-supervised Support Vector Machines (QN-S3VM) [37, 38] classification
- Semi-supervised Contrastive Pessimistic Likelihood Estimation (CPLE) [82] classification

By the results, presented in Figure 4.10, it is seen that the best average classification accuracy was achieved with the use of QN-S3VM classification. However, more interestingly, with the use of as few as 20 labeled samples, an average accuracy of around 85% was attained with both QN-S3VM classification and CPLE classification with optimistic assumptions (i.e., optimistically assume the best case assignment for unknown labels). Nevertheless, it is also evident that there is a rather high standard classification deviation of about ±5% in case of only using 20 labeled samples during training. By increasing the number of samples to 40 labeled samples, the classification deviation is reduced to about ±2%, while the average accuracy is increased to about 90% in case of the QN-S3VM model. Furthermore, once approaching pure supervised learning (> 200 labeled samples), it is, once again, seen that the QN-S3VM model reports best results, where an overall best average classification accuracy of 96.1% was achieved through training the QN-S3VM classifier with 400 labeled samples.

Figure 4.10: Resulting average classification accuracy together with the standard classification deviation for each trained and tested model for our approach of learning the anchoring functionalities through semi-supervised classification algorithms.
4.5 Discussion

Throughout this chapter, we have introduced all the fundamental components that together compose the proposed anchoring framework. We have, initially, presented a perceptual system, which can handle and process the constant stream of input RGB-D sensor data in order to segment and detect objects in dynamic and uncontrolled scenarios. Next, we have introduced a symbolic system that is utilized for the purpose of establishing and modelling the percept-symbol correspondence between segmented sensor data and symbolic values. We have further shown how available third-party resources can be used to support the grounding of symbols to perceptual sensor data. In particular, we have outlined how the use of a Convolutional Neural Network (CNN), trained with samples from an extensive image database (ImageNet), has been integrated and used in order to classify objects and to associate a symbolic category to a perceived object. Finally, we have introduced an anchor management system, which comprises a novel anchoring matching function for administering bottom-up matching in order to determine if an anchor has previously been perceived (or not).

It should, however, be noted that the presented techniques and methods for, initially, processing the input sensory data should be considered replaceable black-box approaches, which have been used for the means of segmenting the perceptual data that originates from an object of interest in the scene. For example, the used object segmentation method could be replaced with a convolutional network-based semantic segmentation approach [81]. This, however, requires an adequate dataset of pixel-wise mask annotations for training the network to detect the objects of interest, which is something that is not always publicly available and must, therefore, further be addressed, e.g., through weakly supervised learning [61].

In this chapter, we have further highlighted the challenge of matching and accurately maintaining anchored objects in a bottom-up fashion, and without prior knowledge about the objects per se. This problem boils down to the task of combining and interpreting the result of matching anchors based on continuous attribute values (which initially is handled by an anchoring matching function), in order to truly determine if an anchor object has previously been perceived or not. This challenge is particularly problematic in the context of bottom-up anchoring of objects in real-world scenarios with, presumably, endless possibilities of potential objects. To address this challenge, we have introduced a novel anchoring matching approach based on classification and learning from examples in order to determine whether a perceived object has previously been observed, and thereby learning to invoke the correct anchoring functionality (acquire or re-acquire). To evaluate the proposed approach, we have demonstrated how a human-annotation interface has, firstly, been used in order to collect a large dataset of samples for which the human counterpart have provided the ground truth about what he/she would consider as the
appropriate anchoring action for presented candidate objects in various anchoring scenarios. The collected dataset has, subsequently, been used for the training and validation of various classification algorithms in the interest of introducing a learned model that replaces traditionally hand-coded anchoring matching approaches (e.g., the rule-based approach previously introduced in Paper III).

In order for an anchoring framework to adapt and learn to adjust to new environments without extensive training, we have further demonstrated that the problem of invoking correct anchoring functionality, likewise, can be approximated by semi-supervised learning from a sparse number of annotated samples. A truly practical framework must, however, also account for scalability as more and more objects are encountered over time of operation. To address this issue, we have, in addition, proposed that the maintenance and storage of anchors should be supported by a database solution. As we have demonstrated in this chapter, a database-centric anchoring approach also introduces the capability of formulating database queries based on the symbolic values in order to retrieve corresponding maintained perceptual data. Hence, in order to practically anchor objects at scale in scenarios with several thousands of object instances, we will hereby conclude this chapter by suggesting the following combined anchoring procedure:

1. Reduce the search-space of possible anchor candidates by matching anchors through the use of symbolic values and a database query interface (as described in Section 4.4.1).

2. For ambiguous candidates, proceed by matching anchors bottom-up through the use of attribute values and proposed attribute matching algorithms (as presented in Section 4.4.1).

3. Resolve ambiguities through classification (as suggested in Section 4.4.3), in order to determine if a candidate object matches an existing anchor object, and, subsequently, maintain the anchor of an object by invoking the appropriate anchoring functionality acquire or re-acquire (as described in Section 4.4.2).
Chapter 5
Extension: Tracking of Anchors Over Time

“When the going gets weird, the weird turn pro.”
— Hunter S. Thompson

A necessity for maintaining a consistent notation of an anchored object – perceived within a dynamic and changing environment – is to support the anchoring procedure with state of the art in tracking algorithms to track anchored objects in space and time. The integration of object tracking into object anchoring has been explored in two of the publications included in this thesis:

✓ In Paper III, we outline our work on tracking the changes of anchors using a particle filter-based tracking approach directly on the lowest perceptual level (presented in Section 5.2.1). Moreover, by recording and maintaining the trajectory of anchored objects, we can also reveal additional properties of objects, e.g., how objects are used by recognizing the actions applied to the objects. In Paper III, we put focus on how we are able to learn actions, applied to anchors, and how that information can be used together with other sources of information, e.g., commonsense knowledge available in the ConceptNet semantic network [118], to reason about uncertainties the inevitably emerge in the tracking process while tracking objects in dynamic scenes (described in Section 5.4).

<table>
<thead>
<tr>
<th>Question(s):</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
</table>

✓ In Paper II, we have, opposite previously work, explored high-level object tracking by presenting the integration of a probabilistic inference sys-
tem into the concept of object anchoring (presented in Section 5.2.2). By representing the relation between objects, expressed in formal logic, we were also able to take advantage of the combined framework to reasoning about the spatial relationship between objects, as presented in Section 5.5.

<table>
<thead>
<tr>
<th>Question(s):</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
</table>

5.1 Extension of the Anchoring Definition

A limitation in the original anchoring definition [18, 19] (cf., Chapter 3), is that the definition, traditionally, has assumed unary perceptual-symbol correspondences and denoted an anchor as the latest perceptual update of an object. This definition would not permit the maintenance of the history of anchors, as the perceptual and symbolic information of an anchor change over time for every new perceived matching instance of an object. To circumvent this problem, we have further introduced an extension of the anchoring definition and introduced two different types of attributes, first presented in Paper III:

- **Static attributes**, $\phi$, which are unary within the anchor (according to the traditional definition), and which combined identifies an anchor.

- **Volatile attributes**, $\phi_t$, which are individually indexed by time $t$, and which are maintained in sets of attribute instances $\varphi$, such that $\phi_t \in \varphi$.

5.1.1 Maintaining the Historical Trace of an Anchor

Using the notation above, we were able to both maintain a persistent notion of an anchor (according to the traditional definition), and maintain the historical trace of an anchor. More specifically, we were, for example, considering a classification attribute of type $\phi^\text{class}$ as a static attribute (since this attribute was classifying the object), while a position attribute of type $\phi^\text{pos}$ was considered as a volatile attribute (since this attribute was part of a movement history of an anchor), and which, therefore, was maintained in a set of 3-D points (each index by time $t$), such that $\phi^\text{pos}_t \in \varphi^\text{trajectory}$. To further exemplify, other types of volatile attributes for objects could be $\phi^\text{gas}_t \in \varphi^\text{smell}$ in the case of olfactory concepts as described in [84], or in the case of color of certain objects $\phi^\text{color}_t \in \varphi^\text{state}$ for representing fruits as discussed in [83].

Introducing this distinction of attributes further introduce a marginally different definition of the anchoring matching procedure (cf., Section 4.4). For example, the normalized similarity distance between a volatile position attribute $\phi^\text{pos}_y$ of a candidate object and last updated position $\phi^\text{pos}_{t-k} \in \varphi^\text{trajectory}$ of a previously stored anchor $\alpha^x$, is, instead, calculated according to:

$$d^\text{pos}_{x,y}(\phi^\text{pos}_{t-k,x}, \phi^\text{pos}_y) = e^{-L^2(\phi^\text{pos}_{t-k,x}, \phi^\text{pos}_y)}$$ (5.1)
5.2 Variations of Object Tracking in Conjunction with Anchoring

Alongside the publications presented in this thesis, different approaches to object tracking in association with perceptual anchoring have also been investigated. In this section, we summarize studied strategies and discuss the strength and weaknesses of each explored approach. Furthermore, a depiction of how each studied object tracking approach has been integrated with the anchoring framework (presented in Chapter 4), can be seen in Figure 5.1.

![Diagram of anchoring framework](image)

**Figure 5.1:** The anchoring framework combined with different object tracking approaches: (left) object tracking on perceptual level through the use of adaptive particle filter-based tracking algorithm based on point cloud data, and (right) high-level object tracking through the use of an inference system based probabilistic logic programming for joint relational probabilistic data association and object tracking.

### 5.2.1 Object Tracking on Perceptual Level

In Paper III, we have considering the possibility to track objects directly on the perceptual level through the use of perceptual sensory point cloud data, as illustrated in Figure 5.1 (left). The same input RGB-D stream of sensor data, used for segmenting and detecting objects (presented in Section 4.2), was
in this case further used for tracking the movements of objects. This object tracking procedure was, therefore, dependent upon the same segmented spatial percepts: \( \{ \pi_1^{\text{spatial}}, \pi_2^{\text{spatial}}, \ldots, \pi_m^{\text{spatial}} \} \), to initiate the tracking. More specifically, the object tracking procedure utilizes a particle filter-based tracking algorithm that adaptively samples particles based on an error estimate using the Kullback–Leibler Divergence (KLD) [34, 35], which is implemented and included in the Point Cloud Library (PCL) [112]. An example of segmented point cloud clusters with the particles from the adaptive particle filter-based object tracking algorithm can be seen in figure Figure 5.2. This integration was further facilitated by sharing the unique identifier \( x \), for each anchor \( \alpha^x \), such that the identifier was provided as an associated identifier for each particle filter for directly track an anchor on the perceptual level.

![Figure 5.2: A depiction of adaptive particle filter-based object tracking based on point cloud data. The particles from the particle filter are illustrated as an overlay over the segmented point cloud clusters.](image)

This type of particle filter algorithm is, however, designed for the tracking of the cluster of 3-D points that constitutes one single object of interest. To circumvent this problem of single object tracking, a pool of particle filters (one filter for each detected object) was, therefore, employed in the system setup presented in Paper III. Although this approach provided an adequate tracking procedure for table-top scenarios with few moving objects, the procedure was not able to handle, with the same frame rate, more complex scenarios that entail tracking of multiple objects of a higher order.
5.2.2 High-Level Object Tracking

To avoid the problem with a limited frame rate as result of tracking objects on the perceptual level (as introduced in previously Section 5.2.1), an alternative approach of tracking objects on a higher symbolic level was, instead, investigated through the integrating of a probabilistic inference system into the anchoring framework. An integration presented in Paper II, which is illustrated in Figure 5.1 (right). The inference system, used for this work, was based on Dynamic Distributional Clauses (DDC) [94], which is formulated as an extension of the logic programming language Prolog [119]. In addition, the inference system uses Joint Probabilistic Data Association Filter (JPDAF) [4] for both tracking the states of perceived objects (stored and maintained in a temporary world model), as well as expressing the relation between occluded objects and occluding counterpart objects.

The probabilistic approach, presented in Paper II, was largely inspired by previously presented work on probabilistic multiple hypothesis anchoring for semantic world modeling, presented by [30]. Nonetheless, contrary to the work presented by Elfring et al., we did not encourage a highly integrated approach that supported a tight coupling between object anchoring and probabilistic object tracking. Instead, our approach was to maintain a loose coupling, which was motivated by the fact that Multiple Hypothesis Tracking (MTH) procedures [103], will inevitably suffer from the curse of dimensionality [8]. A truly probabilistic anchoring approach [30], will, therefore, further propagate the curse of dimensionality into the concept of anchoring. A limitation in the use of MHT for world modeling that further has been discussed in a recent publication on the topic of data association for semantic world modeling [136].

More specifically, our presented approach was to propagate both perceptual and symbolic information about a created or maintained anchor (as result of either one of the two principle functionalists acquire or re-acquire), such that both the updated 3-D position attribute, $\phi_{t,x}$, and the higher level symbolic information, $p_{x}^{\text{color}}$ and $p_{x}^{\text{size}}$, composed the observation model for updating the associations between perceived objects and internal representations of objects. By administering this loose coupling, we were able to both update and track the most recently updated representation of perceived objects, stored and maintained in a temporary world model (TWM) through the inference system, as well as maintaining updated representations of all perceived objects, stored and maintained in a permanent world model (PWM) through the anchoring framework.

5.3 Anchoring Track Functionality

Despite the use of different object tracking approaches in combination with object anchoring, as outlined in the previous Section 5.2, the updated tracked state of an object must, subsequently, be fed back to the anchor management
system in order to maintain an updated consistent representation of each anchored object. Traditionally, there has further been a symbolically triggered third anchoring functionality; a track functionality [18], which for the extension to bottom-up anchoring [84] was formally integrated with the re-acquire functionality. Together with the work presented in Paper II and Paper III, we have once again suggested two separate functionalities and reinstated a track functionality, such that:

- Track – extends the definition of an anchor $\alpha^x$ from time $t-1$ to time $t$. This functionality is directly responding to the state of the probabilistic object tracker, which assures that the percepts pointed to by the anchor are the adequate perceptual representation of the object, even though the object is currently not perceived.

For the approach used in Paper III, a measured position attribute $\phi_{pos}^{t,x}$, of a tracked segmented percept $\phi_{x}^{spatial}$, was directly forwarded to the track functionality of the anchor management system, as illustrated in Figure 5.1 (left). This volatile position attribute was, subsequently, appended to the set of attributes $\phi_{x}^{trajectory}$ of corresponded anchor $\alpha^x$ (identified by the shared unique symbol $x$), such that $\phi_{pos}^{t,x} \in \phi_{x}^{trajectory}$.

A similar track functionality was implemented for handling the feedback from the inference system that was integrated for the work presented in Paper II, as illustrated in Figure 5.1 (right). Recall that both perceptual and symbolic information about a created or maintained anchors, as result of either one of the two principle functionalists acquire or re-acquire, was propagated to the inference system for the purpose of updating the particles of the state models of corresponding updated anchored objects. The 3-D position $\phi_{pos}^{t,x}$ of an anchor $\alpha^x$ is, therefore, already updated, and it would be redundant to further feeding back the update from the inference system to the track functionality. However, the inference system is also updating the particles of each state model based on a random distribution, even though there is no perceptual update about an object at time $t$, e.g., when an object becomes occluded by another object. It is this information – and this information only – that is fed back to the track functionality. In details, a position attribute $\phi_{pos}^{t,x}$ (calculated as the mean 3-D position of all the particles of a tracked object), is defined at time $t$ and given for all anchored objects in the scene that are presently not perceived through sensory data, e.g., in case of object occlusions.

### 5.4 Learning Movement Actions from Maintained Anchors

Based on the maintained historical trace of an anchored object, we have in Paper III explored the possibility to extract and learn additional properties of objects. In particular, by analyzing the tracked trajectory of anchored objects,
5.4. LEARNING MOVEMENT ACTIONS FROM MAINTAINED ANCHORS

maintained in sets of volatile position attributes $\phi^{\text{trajectory}}$, we were able to extract and use subsets of information in order to learn different movement actions (represented by an attribute set $\phi^{\text{action}}$), which apply to various anchored objects. In this context, we have evaluated different types of sequential representational learning algorithms, such as conditional Restricted Boltzmann Machines (cRBM) [122], Long Short-Term Memory (LSTM) [50], etc., in order to learn the movement actions applied to anchored objects that have been tracked in space and time.

5.4.1 Evaluation of Sequential Learning Algorithms

To collected the background data for our suggested learning approach, we were anchoring objects involved in five different classes of movement actions: ‘pouring’, ‘stirring’, ‘putting’, ‘grating’ and ‘slicing’. The anchored information was, subsequently, extracted from the maintained anchor-space and fed to the learning algorithm. Resulted confusion matrix for the average action prediction for both evaluated cRBM and LSTM learning algorithms, can be seen in Table 5.2a and Table 5.2b, respectively. Through comparison of the results of both evaluated sequential learning algorithms, we concluded that an average classification accuracy of 94.83% was achieved with the use of the LSTM algorithm.

Table 5.1: Confusion matrices for the average classification prediction for the learning of object movement actions by: (a) cRBM, and (b) LSTM.

<table>
<thead>
<tr>
<th>True label</th>
<th>Poured</th>
<th>Pute</th>
<th>Stirred</th>
<th>Grated</th>
<th>Sliced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pour</td>
<td>22</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Put</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stir</td>
<td>2</td>
<td>5</td>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Grate</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>Slice</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>36</td>
</tr>
</tbody>
</table>

(a) cRBM.

<table>
<thead>
<tr>
<th>True label</th>
<th>Poured</th>
<th>Pute</th>
<th>Stirred</th>
<th>Grated</th>
<th>Sliced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pour</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Put</td>
<td>0</td>
<td>14</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stir</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Grate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Slice</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>36</td>
</tr>
</tbody>
</table>

(b) LSTM.

5.4.2 Using Action Learning to Improved Anchoring

By deploying a trained sequential learning algorithm (as presented above), we have, in addition, introduced how the learned generalized representation of an action can be used as a feedback loop to the anchor management system, as illustrated in Figure 5.3a, in order to improve the anchoring process per se. The need for this feedback loop was motivated by errors and ambiguities that inevitably emerge within the object classification procedure (as described in
as a result of motion blur, partly occluded objects in the scene, and the human interference. Errors that also affect the maintained trace of an anchored object, which, subsequently, affects the classification result of the suggested action learning approach. To resolve these problems, we have suggested that an early prediction of the movement action of an object should be established. The result of this prediction is not only a grounded symbolic name for a novel action, $p_{\text{action}}$, but the same prediction is also an estimation of possible object candidates that afford that action. Moreover, to verify and query that an object affords a particular action, we have suggested the use of common-sense information available in the ConceptNet\textsuperscript{1} semantic network [118]. To exemplify, we refer to the result of a query for the concept ‘stir’, presented in Figure 5.3b, where it is evident that e.g. a ‘wooden spoon’ afford a ‘stirring’ action. Consequently, if the framework establishes an early prediction for a novel ‘stirring’ action, it is also possible to strengthen the beliefs of a potential ‘wooden spoon’ object being involved in the same action.

\textbf{Figure 5.3:} Conceptual illustration of how learned action representations are used to improved anchoring. (a) An overview of the integration of learned action representations, which permit feedback to the anchor management system. This feedback loop further utilizes the common-sense knowledge of ConceptNet to query and verify that an anchored object indeed affords an action. (b) The resulting top 10 responses for a query of the concept ‘stir’ in ConceptNet.

\textsuperscript{1}http://www.conceptnet.io/
5.5 Reasoning About Relations Between Anchored Objects

Despite the accuracy of the proposed anchoring approach, presented in Section 4.4.3, there are scenarios where the pure anchoring framework fails to correctly acquire or re-acquire an object, e.g., when an object is occluded and moved by another object. For a changing and adaptable system to handle the world modeling of such scenarios, the system must further incorporate model-based object tracking in order to manage the data association for objects that are not directly perceived by the input sensors. In this section, we will exemplify how our approach of integrating Dynamic Distributional Clauses (DDC) [94] (used by the inference system) into the anchoring framework can handle such scenarios with occluded objects, and as a subsequent result further improve the anchoring accuracy.

5.5.1 Proof-of-Concept

Recall that the mean position in 3-D space of the particles for each object that is not directly perceived at time \( t \), e.g., objects occluded by another object, was fed back to the reinstated track functionality of the anchor management system (cf., Section 5.3). Through the use of this feedback loop, the position of an anchor was, subsequently, updated to the most probable position according to the inference system. To demonstrate how the anchoring framework benefits from this feedback loop from the inference system (and vice versa since the inference system is dependent on observations from the anchoring framework), was the particles (given by the inference system, as described in Section 5.2.2), visualized concurrently with the output of the anchoring framework, as exemplified in Figure 5.4.

By comparing resulting anchored objects, seen in Figure 5.4, it is evident that there is an essential difference in resulting anchors. In the case of purely anchor objects using the anchoring framework (Figure 5.4 – 2nd row from top), it can be seen that the initial ‘apple-1’ object (seen in Figure 5.4 – № 1) is lost while the object is occluded and moved by the ‘skin-1’ object. Subsequently, than the system re-observe the apple object, the anchoring framework can not determine if the object is new ‘apple’ or the previously observed ‘apple-1’, and as a result acquire an new anchor ‘apple-3’ (seen in Figure 5.4 – № 2). In the case of using both the anchoring framework and the inference system (Figure 5.4 – bottom row), and the position of the tracked ‘apple-1’ object (seen in Figure 5.4 – № 3) is fed back to anchor management system while the object is moved, it can, however, be noticed that the apple object, instead,

\(^2\)Rather than using a dedicated classifier for recognizing different human body parts, we have, instead, fine-tuned used GoogLeNet model to recognize human ‘skin’ objects as one of the object categories (see Section 4.3.2).
Figure 5.4: A depiction of how suggested integrated framework benefits of combined object anchoring and probabilistic object tracking. Rows in order from the top: 1st) representing screen-shots of a scenario where a human hand is occluding an apple while the apple is moved, 2nd) corresponding resulting anchored objects while only using the anchoring framework (note that the original ‘apple-1’ object is lost while it is occluded and moved by the ‘skin-1’ object, and a new ‘apple-3’ object is, therefore, acquired in the end of the scenario), 3rd) plotted particles given by the inference system during execution of suggested integrated approach, and 4th) corresponding resulting anchored objects of the anchoring framework supported by the feedback of the inference system (note that in this case is the position of ‘apple-1’ object tracked while it is occluded and moved by the ‘skin-1’ object, and the ‘apple-1’ object is, accordingly, re-acquired in the end of the scenario).

is correctly re-acquired as ‘apple-1’ once the object is re-observed (seen in Figure 5.4 – № 4).
5.6 Discussion

In this chapter, we have shown how we are able to improve the overall anchoring process by integrating a probabilistic object tracking procedure into anchoring. The motivating for this integration is to support the anchoring process in cases where the sensor data for a particular object breaks off, e.g., when object occlusions occur. Initial studies have been conducted with the use of object tracked based on 3-D point cloud data at the lowest perceptual level. Conclusions were, however, made that such point cloud-based tracking approached also introduced a system bottleneck and the approach was, subsequently, abandoned in favor of the integration of an inference system that supported high-level probabilistic tracking and reasoning. For this particular integration, a loosely coupled integration of the systems was favored in order to prevent the curse of dimensionality from propagating from the probabilistic inference system into the perceptual anchoring framework. Nonetheless, in order to retain the combined integrated framework in a coherent cognitive state, a tight feedback loop was further supported for maintaining the consented tracked positions of objects. As a result of this integration, we have further presented the proof-of-concept of how this integrated framework is used to model and manage a consistent world model of perceived objects in dynamic scenarios.

Another aspect that has been covered in this chapter is the possibility to learn additional attributes from the historical trace of tracked and maintained anchors. In order to preserve the history of an anchor, we have further presented an extension of the anchoring definition, which enables the maintenance of the historical trace of anchors as the perceptual and symbolic information of an anchor change over time for every new perceived matching instance of an anchored object. This extension was motivated by the limitation in the original anchoring definition [18], which traditionally has assumed unary perceptual-symbol correspondences and denoted an anchor as the latest perceptual update. Based on the maintained movement history of an object, we have demonstrated how of sequential learning algorithms, such as Long Short-Term Memory (LSTM) [50], are used in order to learn actions that applies to an object. The prediction of perceived action, together with the use of commonsense knowledge (ConceptNet), was, subsequently, used in order to improve the overall anchoring process by establishing an early prediction about the perceived action and which type of objects that might afford the action.

A conventional approach for learning both the interaction that involves objects [69, 98], as well as object affordances [64, 65], from RGB-D data, is to utilize a tracking algorithm for tracking the hand movement of a human user. In the publications presented in this section, we have only considered the tracking of anchored objects, while a human ‘hand’ object might constitute one such anchored object. We have, however, not considered an approach which distinctly tracks the human hand movement. The integration of such human
hand tracking approaches can, therefore, be another area of future work. For example, a possible future direction is to explore the spatial-temporal relationships between objects and human hand actions to learn the function of objects, as suggested by the authors of the work on inferring object affordances from human demonstration [62].
Chapter 6
Application: Anchoring to Talk About Objects

Spoken dialogue is a natural and effortless medium of communication between humans. This together with the evolution of complementary interacting technologies (e.g., gesture recognition), makes spoken dialogue a more functional solution for human-robot interaction. Alongside the ambition to move perceptual anchoring into real-world applications, we further want to enable a more flexible way to talk about objects and to retrieve anchors in a top-down fashion based on the request from the human user. In Paper V, we have presented how we can use perceptual anchoring together a multi-modal dialogue system, and the use of available semantic resources (e.g., the WordNet lexical database [32, 88]), in order to allow for flexible dialogues about anchor objects.

<table>
<thead>
<tr>
<th>Question(s):</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
</table>

This work was largely inspired by previously presented studies where perceptual anchoring is conditioned by a spoken dialogue that is supplemented by gestures either by the robot [87], or by human users [73, 74].

6.1 Top-Down Anchoring

Up until this chapter, we have exclusively been discussing bottom-up anchoring in this thesis. However, the anchoring problem was originally presented as a top-down problem [19], i.e., based on a symbolic description, find the best matching object maintained in an anchor. A symbolically triggered top-down acquisition of an anchor is, accordingly, managed by the following abstract functionality:
Find – takes a symbol $x$ and a symbolic description and returns an anchor $\alpha^x$ defined at $t$ (and possibly undefined elsewhere).

For the work presented in Paper V, we have studied how anchored objects could be referred to in the dialogue of human-robot interaction. In order to find an anchored object in a top-down fashion based on the interaction was, therefore, a similar find functionality required – further described in Section 6.2.1.

### 6.2 Integration of a Multi-Modal Dialogue System

For the work presented in Paper V, we have introduced a multi-modal dialogue framework deployed in a Q.bo\(^1\) mobile robot, which is further connected to an anchoring framework that maintains the states of physical objects in the environment. The centre of attention for this study was, however, the integration between an semantic system and a dialogue system (as illustrated in Figure 6.1), and how integrated system could be used to maintain a fluent dialogue about anchored objects in human-robot interaction scenarios. Nonetheless, an early prototype of the anchoring framework, as presented in Chapter 4, was used in the background to process the stream of visual sensory input data, and consequently anchor perceived object. More specifically, the anchoring framework was, in this case, utilizing binary descriptor attributes (extracted from visual key-point features) together with color attributes in order to detect and maintain objects. The fundamental anchoring approach was, consequently, a similar bottom-up approach, as used throughout all the works presented in this thesis. However, the top-down approach used for the particularly work presented in Paper V, was, instead, to endorse the use of a semantic system that was implemented as an extension to the anchoring framework, and which periodically invoked an anchoring find functionality.

#### 6.2.1 Semantic System

The core of the semantic system, seen in Figure 6.1 – No. 1, is an ontology, which is maintained as a database collection. This ontology is stored in the form of hierarchies in a finite search-space, where the relation between the various nodes is stored in the form of subsumption relations. Thus, a field in the collection contains a sequential list of all the nodes that have to be traversed in order to search the association between semantic concepts. As an intermediate layer between the dialogue system and anchoring system, this system must be able to receive, interpret and respond to both changes in semantic properties of anchors, as well as user requests. This interchange of information between systems was facilitated through three complementary functionalities:

\[^1\text{http://thecorpora.com/}\]
6.2. INTEGRATION OF A MULTI-MODAL DIALOGUE SYSTEM

Figure 6.1: A system overview of presented integration between a multi-modal dialogue system and a semantic system (coupled with an anchoring system).

🔍 **Search** – initiates a search of a lexical database in order to update the ontology whenever a user is requesting something outside the boundaries of existing knowledge. For this purpose, the WordNet lexical database was used [32, 88]. This functionality is created in such a way that all the hypernyms and hyponyms/troponyms of a noun/verb are explored recursively until there are no more nodes to explore. The results are then stored as local copies in the ontology.

🔍 **Request** – receives and responds to user requests from the dialogue system. Based on requested objects (nouns) and/or activities (verbs), a search for possible candidates in the ontology is conducted. If a match for the request is available in existing knowledge, a reply is sent immediately. Otherwise, an additional search is initiated for the requested verb and/or noun. In the latter case, the results of the search are also synchronized with the regularly invoked find function.

🔍 **Find** – the purpose of this function was twofold: 1) update the ontology of the semantic system based on perceived objects, 2) update semantics of the anchor system based on new knowledge as a result of a dialogue with the user. Upon receiving semantic properties of perceived objects, the semantic knowledge is updated such that the ‘is-a’ hierarchies of the ontology are consistent with perceived objects at current time t. Furthermore, a comparison is made between the existing knowledge and semantic properties of perceived objects such that the updated knowledge could be sent as a response back to the anchoring system.
6.2.2 Dialogue System

The dialogue system of the combined framework was orchestrated by the IrisTK multi-modal multi-party authoring platform\(^2\) [117], seen in Figure 6.1 – № 2. IrisTK is an XML-based dialogue platform that was designed for quick prototyping and development of multi-modal event-based dialogue systems. This platform is inspired by the notion of state-charts, developed in [48], and used in the UML modeling language. The state-chart model is an extension of Finite-State Machines (FSM), where the current state defines the effect of events in the system. However, whereas events in an FSM triggers merely a transition to another state, state-charts may allow events to also result in actions. Another notable difference is that the state-chart paradigm will enable states to be hierarchically structured, which means that the system may be in several states at the same time with generic event handlers in an upper state and more specific event handlers in the sub-state of the system’s current state. Also, the transition between states can be conditioned - depending on global and local variables, as well as event parameters.

6.2.3 Establish Joint Communication Routines

The most significant gap to bridge, for this system integration, was to establish a joint communication routine between the individual systems. The anchoring framework, and subsequently also the semantic system, uses the ROS communication protocol [102], while the dialogue system uses a designated protocol that is specific to dialogue. This disparity in protocols is a common problem when it comes to integrating different platforms to build a complete interactive system. Since semantic queries that the dialogue manager extracts from the user utterance might need to be sent to the semantic system (e.g., if the user asks whether “there is something to eat”), a translation module was needed in order to propagate semantic queries over the ROS network, sending requests to anchoring framework, and reversely, receiving responses back for processing by the dialogue system (e.g., to generated a verbal output “sorry, we do not have any food”). To address this problem of divergences in communication protocols, a IrisTK-ROS translation module was implemented and used. This module manages, initially, the translation between XML messages (used by IrisTK platform) and ROS messages (used by the anchoring framework). Translated messages was, subsequently, propagated and republished to respectively local network environments through the use of the WebSockets transport layer and the rosbridge_server\(^3\) package.

\(^2\)http://www.iristk.net/

\(^3\)http://wiki.ros.org/rosbridge_server
6.3 Fluent Spoken Dialogue about Anchored Objects

The experimental setup used for the evaluation of the suggested integration between the anchoring framework and the IrisTK platform, was based upon simulated anchors created from a catalog of images obtained from an on-line grocery store available in Sweden. Simulated anchors were used as it was possible to scale-up the anchor-space without having to solve various issues related to object recognition and in order to create reproducible experiments. The simulated anchor-space was, in this case, populated with a total of 6830 anchors, where each anchor $\alpha_x$ consisted of a descriptor attribute (extracted from each image using detected visual key-point feature in combination with the FREAK [3] algorithm), a color attribute (which represents the most dominant color in the image), and a semantic category (e.g., ‘pasta’, ‘milk’, ‘coffee’, etc.), which the on-line grocery store already had associated with each image.

For the IrisTK platform, requests for anchors according to a predefined grammar, shown in Table 6.1, were used. Each acceptable grammar have one key noun (e.g. ‘food’, ‘drink’, etc.), which was used as parameter for the request against the semantic system of the anchoring framework, illustrated in Figure 6.1 – № 1. In this evaluation, the ontology of the semantic system consisted of the taxonomy of the semantic categories of simulated anchors. As described in Section 6.2.1, the WordNet lexical database was further used to find the connection between a requested noun and the taxonomy of the semantic categories of simulated anchors. The results of searching WordNet were also stored as a local mirrored copy to be used for recurrent requests.

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Key noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>“I would like something to eat”</td>
<td>‘eatable’</td>
</tr>
<tr>
<td>“I am thirsty, is there something to drink?”</td>
<td>‘drink’</td>
</tr>
<tr>
<td>“Do you know if we have any beverage?”</td>
<td>‘beverage’</td>
</tr>
<tr>
<td>“I would like a fruit”</td>
<td>‘fruit’</td>
</tr>
<tr>
<td>“Find me some food”</td>
<td>‘food’</td>
</tr>
<tr>
<td>“Do you know if we have any solid food?”</td>
<td>‘solid_food’</td>
</tr>
</tbody>
</table>

Table 6.1: Acceptable queries used for dialogues about anchored objects.

Assuming that a large anchor-space, in combination with generic semantic queries (as exemplified in Table 6.1), could result in hundreds or thousands of matching candidates that have a semantic connection to a requested noun. Hence, the focus for this evaluation was to explore how a growing anchor-space (together with generic queries), would affect a fluent dialogue. To fur-
ther process matching candidates, and to find the most prominent candidate, a WordNet similarity software package⁴ was, in addition, used together with the shortest path algorithm:

\[ y(s_1, s_2) = \frac{1}{\text{length}(s_1, s_2) + 1} \]  

Where \( s_1 \) and \( s_2 \) are concepts and the length between the concepts is the shortest path between the concepts in the ‘is-a’ hierarchies of WordNet, and where anchor(s) with the highest similarity score, \( y \), between its semantic category and the requested noun was considered as the most prominent candidate(s). However, many anchors can share the same semantic category, and different semantic categories can have the same similarity score. A given similarity score can, therefore, not guarantee a distinct result. Such indistinguishable ambiguous results are further discussed in succeeding Section 6.3.1.

### 6.3.1 Semantic Ambiguities

In Paper V, we have further shed some light upon the issue of ambiguities that emerge from the use of top-down functionalities triggered by semantic symbols. The purpose of this experiment was to monitor the growth of ambiguities (both for anchors, \( \alpha^n \), as well as semantic categories of anchors) in the resulting best candidates of a WordNet similarity search. Ambiguities were, in this case, counted as the number of anchors and semantic categories with the same highest similarity score to a requested noun (which randomly was selected from nouns seen in Table 6.1). For this experiment, a simulated randomly growing anchor-space with 128 possible semantic categories was used, and all results were presented as the average result over 100 simulated tests.

![Figure 6.2: Ambiguities in resulting best candidates with respect to a growing anchor-space.](https://code.google.com/p/wordnet-blast/)
The results, shown in Fig. 6.2, confirms a growing ambiguity in relation to an increasing anchor-space. Even an anchor-space of only five anchors might, in this case, introduce candidate anchors with the same semantic category (or candidate anchors of semantic categories with the same distance to a requested noun). It is, therefore, of importance to look further into the ‘is-a’ hierarchy of WordNet to resolving this problem. Hence, in order to keep a fluent dialogue, we concluded that a common ancestor in the ‘is-a’ hierarchy might be a better answer to a user request. For example, instead of trying to find the uttermost prominent candidate for a requested noun ‘eatable’, a response with the common parent, such as “there is some ‘food’ in the ‘kitchen’”, might be an equally (or better) response to such request.

6.4 Discussion

In this chapter, we have presented our work on integration a multi-modal dialogue system with the anchoring framework. The aspiration for this integration was to maintain a fluent dialogue about anchored objects in human-robot interaction scenarios. Generating natural language descriptions of real-world objects is a core facet of human-robot interaction. However, reaching the target of integrating a combined multidisciplinary human-robot interaction system is not a trivial task. Evaluating a complex and multidisciplinary human-robot interaction system is, likewise, a challenging task. An additional complexity arises from the multitude of components and variables involved in the system, which makes an overarching evaluation of the final system either non-informative or impossible. Many similar systems presented in the literature focuses, therefore, only on sub-solutions to the visionary target. Nonetheless, a prominent system that can learn grounded language models with minimum user interaction has been presented in [17]. This system relies, similar to the system presented in this chapter, upon semantic categories to group sensory data. The meaning of words is acquired through examples together with the use of a semantic clustering algorithm. The integrated system, presented in this chapter, was instead relying on a third part lexical database (WordNet) to extend natural language descriptions of available semantic categories.

An imminent dilemma that arises from human-robot interaction about the symbolic properties of anchored objects, which we have stressed in this chapter, is the problem of symbolic ambiguities when referring to anchored objects. A knowledge-oriented architecture that aims to address this problem has been presented in [73, 74]. This system includes a novel component for natural language interpretation that relies on structured symbolic models of the world in order to handle ambiguities while referring to objects through three classes of utterances (commonly found in human-robot interact), namely: statements (i.e., new facts), orders (or desires), and questions.
Chapter 7
Conclusions

In this chapter, we present a summary of the challenges and contributions addressed in this thesis. We further outline a critical assessment of the presented work. Finally, we conclude this thesis by discussing possible directions for future work.

7.1 Challenges and Contributions

In this thesis, we have presented our sensor-driven approach for connecting abundant perceptual data and high-level, symbolic knowledge by using and extending the concept of perceptual anchoring. More specifically, we have introduced a bottom-up anchoring approach for handling both visual and spatial perceptual data that emerges from the use of a RGB-D sensor, introduced in Chapter 4. In order to address research question Q2 (cf., Section 1.3), we have demonstrated how the use of available third-part resources, e.g., the use of a Convolutional Neural Network (CNN) trained with samples from a massive image database, is integrated in order to facilitate the classification of objects at the perceptual level. Furthermore, with regard to research question Q1, we have presented a novel matching function in the context of anchoring. Contrary to previous works on anchoring, which traditionally have addressed the matching of anchors through a simplified approach based on the use of symbolic values, we have in this thesis explored the idea of matching anchors based on continuous attribute values measured from percepts. Nonetheless, interpreting the result of the matching procedure to determine if an anchor has previously been perceived (or not), is not a trivial task. Especially not in the context of bottom-up anchoring in real-world scenarios with unlimited possibilities of objects. In this thesis, we have addressed this problem through the use of traditional classification algorithms and learning from examples. Different classification approaches were trained with human-annotated samples of anchored objects in order to learn how to determine if an object matches an existing anchor and
if the object should be *re-acquired* as the already existing matching anchor, or if a new anchor should be *acquired* for the object.

A deterministic anchoring approach, such as the approach presented in Chapter 4, is based on perceptual observations in order to deterministically update and maintain modeled objects. In dynamic and changing scenarios, situations may arise where the stream of perceptual sensor data for a particular object breaks off, e.g., when object occlusions occur. It is, therefore, important to couple an anchoring framework with an appropriate probabilistic procedure that can speculate about the actual state of the real world, given the compromised stream of perceptual data. In Chapter 5, we propose an extension of the anchoring framework through the coupling of probabilistic object tracking. In response to research question Q3, we have studied two different approaches for coupling object tracking to anchoring: 1) object tracking based on 3-D point cloud data at the lowest perceptual level, which further introduced a system bottleneck and which, therefore, was abandoned in favor of; 2) high-level object tracking through the integration of an *inference system*. Motivated by the limitation in the original anchoring definition, which prohibited the modeling of the history of an object, we further introduce an extension to the anchoring definition that accounts for the maintenance of the historical trace of an anchor. Based on the historical trace of an anchored object, together with the use of common-sense knowledge (related to research question Q2), we have further demonstrated how additional properties of an object can be learned, e.g., the property of which *action* an object can *afford*.

Finally, to demonstrate the variety of application domains derived from object anchoring, we present, in Chapter 6, an integration between an anchoring framework and a multi-modal dialogue system. A typical limitation of dialogue systems is that a predefined grammar is required. In Chapter 6, we present how such predefined grammar together with associated symbolic *noun* triggers were used in a *top-down* manner to request an answer based on the semantic properties of anchored objects, and hence, sustain a fluent dialogue about objects in human-robot dialogues. In order to search for the semantically best matching candidate among anchor objects, the hierarchies of the *WordNet* lexical database were explored. We further shed some light upon the problem of *ambiguous results* that emerge from a symbolically triggered *top-down* approach, which in this case arose from several matching candidates with the same similarity distance in the hierarchies of the *WordNet* database. Such ambiguities, which surface from the use of a *top-down* approach, also motivated the choice of a *bottom-up* approach, as principally used throughout the work presented in this thesis.

### 7.2 Critical Assessment

The primary underlying objective of this thesis is to advance the field of perceptual anchoring by introducing a practical anchoring procedure that is suitable
for autonomous agents operating in real-world scenarios. Even though we have taken a leap towards this goal with the work presented in this dissertation, there are still a number of issues and limitations that must be addressed before the overarching objective can be considered fulfilled.

The first point of limitation concerns the use of visual key-point features for identifying objects at the perceptual level. Throughout the works presented in this thesis, different attributes and different combinations of attributes are used, as summarized in Table 4.1. In early presented works (Paper IV & Paper V), we assumed that an object could be identified solely based on visual binary descriptors together with the use of suggested f-sum approach. This approach was later (Paper I, Paper II & Paper III), discarded in favor of other attributes, both visual and spatial. The suggested f-sum algorithm was, in this case, rejected based on two fundamental limitations:

1. Not all objects have a texture that produces distinct key-point features, and which, subsequently, results in a unique descriptor for an object.

2. The suggested summative approach was based on the entire set of descriptor strings that originate from the same physical object, and the method required, therefore, the full view of an object in order to match the object accurately. A full visible view of an object is not always the case, especially not in dynamically changing real-world scenarios.

The second point of limitation concerns the processing of used perceptual input sensor data. In this thesis, we have investigated how both visual and spatial attributes, extracted from the perceptual data provided by a RGB-D sensor, can facilitate the anchoring of objects in a bottom-up fashion. However, the used initial object segmentation and detection procedure (provided together with the PCL library [112]), has been used as an black-box approach for the means of segmenting and detecting objects of interest in the scene. Considering that this procedure has produced an acceptable stream of perceived segmented objects, we have not investigated whether this procedure truly harnesses the structured information of organized 3-D point cloud data most optimally and efficiently. Without further studies, we cannot dismiss that a suboptimal performance of this object detection procedure might result in a system bottleneck of the presented framework. In the context of sensor-driven bottom-up anchoring, such a bottleneck at the perceptual level could be especially problematic if perceptual data is lost between frames of input sensor data.

Another issue concerns the limited use of mobile autonomous agents. The anchoring problem has emerged from the need for robotic planning systems to plan and execute actions involving objects [19], and the anchoring problem has, therefore, mainly been investigated from the perspective of an autonomous mobile robot, e.g., in robot navigation scenarios. For the publications presented in this thesis, the anchoring problem has primarily been investigated from the
perspective of static system setups, while it remains for future work to investigating how presented anchorage strategy applies to a mobile system setup (e.g., similar to the work presented in [45, 109]). It should, however, be noted that the bottom-up anchoring approach, presented in this thesis, continuously maintain a world model of all the objects perceived over time, which makes the approach relevant for any autonomous agent that requires an updated model of the perceived environment. A mobile robot platform, on the other hand, commonly affords to handle only a single object at a given time (e.g., planning and executing an action involving a single object), and that it, therefore, could be more intuitive to design a mobile robot according to a goal-oriented top-down anchoring approach. A mobile platform further entails an additional level of complexity that inevitably stems from uncertainties and noise in localization and navigation. Nevertheless, in the interest of reducing this complexity, the distinction between mobile versus static could be abridged, to a certain degree, by approximating a static world coordinate frame with the use of registration algorithms that align the input sensor data with a global world coordinate frame. For example, a prominent method for registering sensor data to a global world coordinate frame, and which is based on visual key-point features (similar to the visual key-point features introduced in Section 4.2.4), has been introduced through the algorithm for simultaneous localization and mapping using ORB key-point features (ORB-SLAM) [91, 92].

A final issue regarding limitations concerns experimental evaluations. Although datasets for evaluating individual subtasks of anchoring are commonly and publicly available (e.g., the ImageNet database [29] for object classification tasks), there exist no datasets, to the best of our knowledge, that provide the combined ground truth data for all the steps and components that are involved in the process of anchoring. As a result, specific datasets of anchored objects have been collected and, subsequently, post-processed and annotated for the various works presented in this thesis. In Paper I & Paper II, we have introduced and used a human-annotation interface for the purpose of supporting the collection of ground truth data about anchored objects, and thereby circumvent the need for post-processing and annotating of the collected datasets. However, this interface was introduced for the specific purpose of collecting datasets to learn how to invoke the correct anchoring functionalities (acquired or re-acquired). Besides, the datasets were mainly collected by a single user (the author of this thesis), which might have resulted in a learned model that is biased towards how that individual user perceived the environment. In order to truly benefit from this interface, the interface must first be adapted and improved such that all intermediate data of all the procedures of the systems involved in the anchoring process are captured and collected, and the interface must, secondly, be used to collect more extended datasets, which also are collected with the help of more users.
7.3 Future Work

A direction of future work that, in particular, could be of interest to further explore is the integration of various learning approaches at different levels of perceptual anchoring. Machine learning is one of today’s most rapidly growing technical fields [57], and it would be ludicrous not to exploit future trends and developments in machine learning algorithms also within the context of perceptual anchoring. For example, machine learning approaches can further be applied to support both the anchoring process at the symbolic level, e.g., learning strategies for acquiring the grounding of predicate symbols to corresponding perceptual data, as well as on the perceptual level, e.g., learning approaches for processing the stream of perceptual sensor data.

Together with the work presented in Paper III, we have outlined our initial work on the topic of learning additional object properties based on the maintained tracked historical trace of an anchored object. More specifically, we have explored how sequential learning algorithms (e.g., LSTM [50]), can be used to learn the actions from maintained trajectories of anchored objects. The possibility of learning additional properties of objects, through the use of various learning approaches and based on maintained anchored objects per se, is undoubtedly an interesting direction of future work.

The integration of object tracking into anchoring is also a subject for further investigation. In Paper II, we have introduced the integration of high-level probabilistic object tracking into the anchoring framework. However, we have, in this thesis, only addressed how we maintain an uni-modal probability distribution in an anchor. A refinement to likewise handle multi-modal probability distributions, e.g., multi-modal distributions of the positions of unobserved objects, would render our anchoring approach truly probabilistic, allowing us to keep track of multiple hypotheses (similar to previous work on probabilistic anchoring [30]).

Based on the critical assessment (presented in the previous Section 7.2), there are also a few possible areas for future works at the technical level. In particular, the initial perceptual pipeline and the segmentation and detection of objects (as presented in Section 4.2), is one such area that needs additional attention. As stated by the authors behind the work on semantic world modeling [136], there are some distinct domain characteristics to take into consideration while representing a world model, such as the fact that the state of most objects does not change between frames. A facet of future work is to exploit this characteristic at the lowest perceptual level, such that state changes are tracked directly between 2-D visual frames of used RGB-D sensory data, and thereby reduce 3-D object procedure to only consider a subset of the environment for which the state has changed between frames. An alternative future direction for improving 3-D object detection is to follow the lines of the research presented in [45, 109], where contextual relations inherent to human-made spaces are explored in order to support 3-D object recognition, e.g., a long, thin object to
the left of a ‘plate’ is more probable to be a ‘fork’ than a ‘spoon’. Alternatively, there have been proposed object segmentation and detection approaches based on 2-D depth images, together with the use of a Convolutional Neural Network (CNN) [46]. Following this trend of using a CNN approach for segmenting and detecting objects is, likewise, a possible future direction of work for the presented anchoring framework.
References


REFERENCES


REFERENCES


Publications in the series
Örebro Studies in Technology


