The Attentive Robot Companion

Learning Spatial Information from Observation and Verbal Interaction

Leon Ziegler

Declaration of Authorship

According to Bielefeld University's doctoral degree regulations §8(1)g:
I hereby declare to acknowledge the current doctoral degree regulations of the Faculty of Technology at Bielefeld University. Furthermore, I certify that this thesis has been composed by me and is based on my own work, unless stated otherwise. Third parties have neither directly nor indirectly received any monetary advantages in relation to mediation advises or activities regarding the content of this thesis. Also, no other person's work has been used without due acknowledgment. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged. This thesis or parts of it have neither been submitted for any other degree at this university nor elsewhere.

Leon Ziegler	Place, Date

The Attentive Robot Companion

Learning Spatial Information from Observation and Verbal Interaction

Leon Ziegler

March 2015

A doctoral thesis presented for the degree of
Doctor of Engineering (Dr.-Ing.) at

Faculty of Technology
Bielefeld University
Applied Informatics
Inspiration 1
33619 Bielefeld
Germany

Reviewers

Dr.-Ing. habil. Sven Wachsmuth
Prof. James J. Little

Examination Board

Prof. Dr. Philipp Cimiano
Dr.-Ing. Kirsten Bergmann

Defended and approved on Monday, June 22, 2015

Printed on permanent paper as per ISO 9706.

Acknowledgments

This thesis would not have been possible without the support of so many people. At this point I would like to take the opportunity to express my appreciation and say “thank you”. First of all, I owe the possibility to even study and choose a field I am truly interested in to the overwhelming support of my parents and family who always believe in my abilities and supported me not only financially but also with their encouragement in numerous other ways. Especially during my time as a PhD student I have to thank my lovely girlfriend Julia, who never tired of providing me with optimism and confidence. It helped a lot to know that someone is always there for support and assistance outside the university. You are the one that had to suffer from the very time consuming and resource demanding endeavor that is a doctoral thesis. But also my friends, especially my former roommates deserve a special “thank you”.

In my professional environment, special thanks go to Sven who has been a great supervisor and always provided valuable input and discussions throughout the process of this thesis. Furthermore, I would like to thank Jim Little to also instantly agree on reviewing my thesis.

Thank you Gerhard, Franz, and Britta to accommodate me in the Applied Informatics Group and proving such a wonderful working environment. The same holds for all my colleagues who always ensured a communicative, friendly, and amusing atmosphere. Especially Frederic, Florian, and Marco helped including me in the group when I started my PhD, and also Johannes for sharing experiences in study and research since school. They also accompanied me through many years of participating in RoboCup. I am very

thankful for the opportunity of being a part of Team ToBi.

Speaking of which, I want to thank all other ToBis for their hard work and commitment during the various competitions. In particular, I want to thank Sven (as the team leader), Frederic (for the early years), as well as Sebastian, Matthias, and Lukas (for the more recent years). It has been a great time not only meeting all of you but also working with you – and supervising at least some of you. Thank you all, it has been a great experience and a great success!

Furthermore, I thank Jens and Florian for sharing an office and the fruitful collaborations and discussions. Also thanks to my student assistants Michael, Lukas, Phillip, and Tobias.

Finally, I very much value the consent of my sister Lena, as well as, Gwendolyn and Johannes to proofread my thesis.

Thank you.

Abstract

This doctoral thesis investigates how a robot companion can gain a certain degree of situational awareness through observation and interaction with its surroundings. The focus lies on the representation of the spatial knowledge gathered constantly over time in an indoor environment. However, from the background of research on an interactive service robot, methods for deployment in inference and verbal communication tasks are presented. The design and application of the models are guided by the requirements of referential communication. The approach here involves the analysis of the dynamic properties of structures in the robot's field of view allowing it to distinguish objects of interest from other agents and background structures. The use of multiple persistent models representing these dynamic properties enables the robot to track changes in multiple scenes over time to establish spatial and temporal references. This work includes building a coherent representation considering allocentric and egocentric aspects of spatial knowledge for these models. Spatial analysis is extended with a semantic interpretation of objects and regions. This top-down approach for generating additional context information enhances the grounding process in communication. A holistic, boosting-based classification approach using a wide range of 2D and 3D visual features anchored in the spatial representation allows the system to identify room types. The process of grounding referential descriptions from a human interlocutor in the spatial representation is evaluated through referencing furniture. This method uses a probabilistic network for handling ambiguities in the descriptions and employs a strategy for resolving conflicts. In order to approve the real-world applicability of these approaches, this system was deployed on the mobile robot BIRON in a realistic apartment scenario involving observation and verbal interaction with an interlocutor.

Contents

List of Tables **V**

List of Figures **VII**

Glossary **XI**

1. Introduction **1**
1.1. Robot Companions in the Home 4
1.2. From Vista Space to Environmental Space 8
1.3. Research Questions 9
1.4. Scenario & System Foundation 11
1.5. Outline 14

2. Analysis and Statement of the Research Problem **15**
2.1. Functional Requirements 16
2.2. The Choice of Scope 17
2.3. Knowledge Representation 18
2.4. Applying a Situation Model in Interaction 20
2.5. Summary: Contribution of this Thesis 22

3. Partitioning the Workspace **25**
3.1. The Geometric Foundation 28
3.2. Detecting Roles in Articulated Scenes 32
3.2.1. Benefits of Observed Functional Roles 33

3.2.2. The Challenge in Segmenting a Scene 35
3.2.3. The Articulated Scene Model 37
3.3. Anchoring and Integrating Egocentric Models 41
3.3.1. A Twofold Spatial Representation 42
3.3.2. Registering a Scene Model with the Current View . . 44
3.3.3. Generating a Valid Model for the Current View 45
3.3.4. Applications Exploiting the Model's Potential 51
3.4. Focusing the Robot's Attention 55
3.4.1. The Interlocutor's Viewing Direction 56
3.4.2. Detecting Interaction Spaces for Manipulation 59
3.4.3. Repositioning for Observation 61
3.5. Evaluation 63
3.5.1. Quantitative Evaluation 63
3.5.2. Event Detection with ASM 75
3.5.3. Robot Behavior Performance 79
3.6. Summary 79

4. Applying Semantics **81**
4.1. Furniture Categorization 87
4.1.1. Implicit Shape Model 88
4.1.2. Ray-Based Hough Space Voting 91
4.1.3. Evaluation 96
4.1.4. Summary & Discussion 99
4.2. Classification of Household Objects 102
4.2.1. Boosted Classification 104
4.2.2. Evaluation 109
4.3. Room Categorization 116
4.3.1. Generation of Training Data 117
4.3.2. Anchoring of Features 118
4.3.3. Evaluation 121
4.4. Summary 131

5. Perception and Communication **135**
5.1. Benefits of Combining Perception and Communication 140
5.2. Reference Frame Selection in Human Communication 141
5.2.1. Conducting an Online Study 144
5.2.2. Empirical Results 144

5.3. A Probabilistic Model 146
5.3.1. Visual Analysis 146
5.3.2. Maintaining and Updating the Spatial Network 148
5.3.3. Resolving Conflicts 154
5.3.4. Adaptation to Personal Preferences 156
5.3.5. Application in Human-Robot Interaction 158
5.4. Evaluation 161
5.4.1. Online Evaluation 161
5.4.2. Real-World Evaluation 166
5.5. Summary 174

6. Discussion & Conclusion **177**

Bibliography **181**

Appendices **203**

A. Situation cases for ASM evaluation **205**

B. Results from Multi-View ASM Evaluation **207**
B.1. Evaluation of simple ASM 208
B.2. Evaluation of naive matching ASM 209
B.3. Evaluation of multi-view ASM 210

C. Models for 3D ISM Training **213**

D. Results from Evaluation of Household Object Classification **215**

E. Questionnaire for RSM evaluation **219**

List of Tables

3.1. Categories for pixel-wise evaluation 68
3.2. Categories for event evaluation 77

4.1. Confusion matrix of the voting scheme evaluation 97
4.2. Results of the furniture categorization 98
4.3. Recognition results on real-world indoor scenes 99
4.4. Confusion matrix of E-SAMME-ALL condition 112
4.5. Confusion matrix of the E-SAMME-3D condition 113

D.1. Confusion matrix of the E-SAMME-2D configuration 216
D.2. Confusion matrix of the E-SAMME-OBJ-T configuration . . 216
D.3. Confusion matrix of the E-SAMME-OBJ-S configuration . . . 217
D.4. Confusion matrix of the SVM-SURF configuration 217

List of Figures

1.1. The humanoid robot Nao 4
1.2. First generation of BIRON 5
1.3. Cosero 6
1.4. BIRON II 13

3.1. SLAM example 26
3.2. Articulated Scene Model 33
3.3. Scene segmentation processing pipeline 35
3.4. Twofold spatial representation 43
3.5. Registering a scene model 45
3.6. Naive model matching 47
3.7. Merging premises 49
3.8. Rear projection to view frustum 50
3.9. Scene from different perspectives 51
3.10. Movement strategies for a mobile robot 54
3.11. Multi-modal anchoring 57
3.12. *SeAM* Layers 60
3.13. Viewpoint calculation 63
3.14. Schematic visualization of settings (I) 65
3.15. Schematic visualization of settings (II) 66
3.16. Required information for quantitative analysis 67
3.17. Schematic visualization of settings (II) 69
3.18. Quantitative results from additional test cases 70
3.19. Comparison of naive matching and merging algorithm 72

3.20. Evaluation results from additional settings 74
3.21. Evaluation results from additional settings 74
3.22. Scenario for the qualitative evaluation 76

4.1. Virtual scans of furniture meshes 89
4.2. Clustering of vote rays 93
4.3. Intersection of spheres and vote rays 94
4.4. Recognition results on real world indoor scenes 100
4.5. SAMME error plot for changing number of classes 107
4.6. Objects used for evaluation of E-SAMME 110
4.7. Recognition results on household objects 112
4.8. Feature appearance in E-SAMME 113
4.9. Feature test error comparison for object recognition 114
4.10. A reconstruction of a living room. 119
4.11. Feature test error comparison 122
4.12. Single feature comparison 124
4.13. Feature appearance in E-SAMME 125
4.14. Confusion matrices from training on room database 126
4.15. Graph of base classifier usage 127
4.16. MLP comparison 128
4.17. Confusion matrices of training on IKEA room database . . . 130

5.1. Leonardo 137
5.2. Reference object with located object for RF selection 142
5.3. Vehicle and opposite objects 143
5.4. Percentage use for each reference frame 145
5.5. Furniture Segmentation 147
5.6. Furniture graph example 149
5.7. Details of the probabilistic model. 150
5.8. A sequence of graph configurations for backtracking 155
5.9. Selected graph configurations after three descriptions 156
5.10. Selected graph configurations after four descriptions 156
5.11. Sequences of graph configurations with RF preferences 157
5.12. Room for online study 162
5.13. Probability distributions of furniture in online study 164
5.14. Correct matches in online evaluation 165
5.15. Describing spatial relations in a real-world apartment 167

5.16. Furniture layout for evaluation 168
5.17. Probability distributions of furniture in real-world study . . . 170
5.18. Correct matches in evaluation 171
5.19. Amount of correct matches by groups 172
5.20. Amount of vertices that were assigned a wrong label 173

Glossary

AdaBoost

A supervised machine learning meta algorithm for combination of several weak classifiers to a single strong classifier.. 84, 85, 102, 104, 105, 109, 179

ASM

Articulated Scene Model. 32, 37, 40, 41, 47, 49, 50, 52–55, 62–65, 67–70, 73–75, 77–79, 86, 146, 159

Base classifier

One of the simple classifiers used in boosting for generating an ensemble classification scheme.. 102–106, 109, 114, 128, 132

BIRON

Bielefeld Robot Companion. 4, 5, 11, 19, 55, 139, 158, 166, 167

BIRON II

Bielefeld Robot Companion V2. 11, 12, 18, 19, 67

BonSAI

Biron Sensor and Actuator Interface. 13, 61, 76, 79

BoW

Bag of Words. 83, 103

BoW

Bag of Words. 111, 120, 123

BRISK

Binary Robust Invariant Scalable Keypoints. 103

DTree

Decision Tree. 102, 103, 110, 114, 121, 127, 128

E-SAMME

Exhausive SAMME. 106, 109–111, 113, 120–122, 128, 129

Environmental space

The psychological space that is projectively larger than the body and surrounds it. It is too large to apprehend directly without considerable locomotion. 8, 116

Figural space

The psychological space that is projectively smaller than the body and can be directly perceived from one place without appreciable locomotion. 8

FPFH

Fast Point Feature Histogram. 103, 110, 123, 125, 127, 131

FPR

False Positive Rate. 72

FREAK

Fast Retina Keypoint. 103, 110

Geographical space

The psychological space that is projectively much larger than the body and cannot be apprehended directly through locomotion. 8

Home tour

A scenario in which a human introduces a new robot to her apartment and shows it around to familiarize it with this new environment. 4, 5, 33

HRI

Human-Robot Interaction. 2, 4, 6, 7, 20, 55, 135–137, 139, 140, 175, 178, 179

ICP

Iterative Closest Point. 43, 72, 117, 158

ISM

Implicit Shape Model. 87, 88, 90, 91, 96–100, 132, 148, 169, 178

KinFu

Kinect Fusion. 117, 118, 121

Lost key scenario

A recurring scenario for demonstrating various aspects of this thesis. A mobile robot is able to tell where certain objects are, just by observing the human's actions. 10, 41, 78, 179

MLP

Multilayer Perceptron. 102, 103, 110, 121, 127–129

Opposite object

The intrinsic left/right axis of opposite objects is primarily assigned in a way that corresponds to standing in front of the object. 143

ORB

Oriented FAST and Rotated BRIEF. 103, 110, 113

PCL

Point Cloud Library. 59

Point cloud

A set of data points in some coordinate system. In this theses this term always refers to a set of points in a three-dimensional cartesian coordinate system. 18, 19, 27, 34, 36, 43–45, 49, 54, 58, 59, 70, 72, 77, 86, 88, 89, 99, 100, 103, 109, 117–119, 121–125, 127, 147, 158

RANSAC

Random Sample Consensus. 59

RBPF

Rao-Blackwellized Particle Filters. 26

RF

Reference Frame. 139–146, 149–153, 157, 158, 160–166, 169, 171–173, 175

RSB

Robotics Service Bus. 13

SAMME

Stagewise Additive Modeling using a Multi-Class Exponential Loss Function. 104–106

SeAM

Semantic Annotation Mapping. VII, 59–61, 79, 80

SHOT

Signature of Histograms of Orientations. 89, 90, 92, 96, 103, 110, 113, 123, 125, 127, 131

SIFT

Scale-Invariant Feature Transform. 103, 110

Situation awareness

An awareness about the geometrical, functional, and social situation an agent is located in. 1, 2, 82, 135

Situation model

This term is used in psychology as means to express the multi-dimensional representation of the situation under discussion. 2, 3, 5, 6, 9–11, 14, 15, 22, 25, 54, 55, 61, 81, 82, 101, 114, 131, 133, 135, 139, 140, 177, 179, 180

SLAM

Simultaneous Localization and Mapping. 26, 43, 59, 60, 67, 79

SPD

Scene Plane Descriptor. 116, 120, 122, 123, 127, 129, 131

Superpixel

A set of individual pixels of a digital image representing an image segment of arbitrary criterion. 36

SURF

Speeded Up Robust Features. 103, 110, 111, 113

SVM

Support Vector Machine. 103, 110, 111, 114, 121, 127

Vehicle object

The intrinsic left/right axis of vehicle objects is primarily assigned in a way that corresponds to sitting in the object. 143

Vista space

The psychological space that is projectively as large or larger than the body but can be visually apprehended from a single place without appreciable locomotion. 8, 116

Chapter 1

Introduction

The development of mechanical and digital hardware is progressing rapidly, so researchers are trying to bring robotic applications into human living and working environments. Personal robots with a human-like *situation awareness* who are able to perform seamlessly as companions in everyday situations are the subjects of many utopian visions. Considering the rapid aging of populations in Europe and many other countries, personal assistive robots are considered a key technology for prolonging the independence of elderly people. According to Schaal (2007), even more functions relevant to our society will be fulfilled by robots, like in education, health care, rehabilitation, and entertainment. However, we have learned that the progression from static and well-defined environments in laboratories or industrial settings to dynamic, uncertain and very complex domains is extremely hard. There is still a long way to go before real personal robots become mature enough to function among us.

> "One reason for this gap is that it has been much harder than expected to enable computers and robots to sense their surrounding environment and to react quickly and accurately."
>
> (Gates, 2007)

An awareness of what the environment looks like is crucial for an artificial agent. In recent years, advances in technologies for sensing and interpreting the surrounding's spatial properties enabled researchers to develop robotic systems that were able to perform in highly complex real-world scenarios (Thrun et al., 2006). But not only the spatial structure of the environment

is important. For a personal robot, it is at least evenly important to know what the structure's function is and what situation it represents. This enables it not only to perform the specific tasks it is asked to do, but it is also a prerequisite for successful *Human-Robot Interaction (HRI)*. Especially in terms of appropriate communication about items in the environment, a sophisticated *situation model* is essential. The term *situation model* is used in psychology as means to express the multi-dimensional representation of the situation at hand (van Dijk and Kintsch, 1983; Johnson-Laird, 1983). Zwaan and Radvansky (1998) state that the model contains at least five dimensions of situations: time, space, causality, intentionality, and protagonist (reference to the main individuals under discussion).

However, it is not enough to build up an isolated knowledge base of facts about the physical environment. In communication, dialog turns are linked across interlocutors, and the meaning of the conversational content depends on the interlocutors' implicit consensus, not on explicit definition (Sacks et al., 1974; Brennan and Clark, 1996). This means that a model for *situation awareness* always depends on the context of the current situation and the alignment in communication — in other words, the *common ground* between the interaction partners. According to Branigan et al. (2000) a coordination of interlocutors occurs when they share the same representation at some level. So, Pickering and Garrod (2004) argue that the "Alignment of situation models [...] forms the basis of successful dialogue". Whereas the alignment is not *per se* necessary for successful communication, alternatives would be very inefficient in terms of production and comprehension of utterances.

From a usability point of view, the components of a system not only have to operate as the developer conceptualizes them, meaning that they fulfill their functions and are technically stable, the system also has to be both easy and safe to use, as well as socially acceptable (e.g. Dix et al., 2004; Nielsen, 1993).

If robots are supposed to actually be involved in our society in the future like Schaal suggests, they need to be accepted by children and adults. This can only be realized if they comply with certain social behaviors and standards that we as humans find acceptable. Dautenhahn (2007) formulates a set of social rules for robot behavior containing different paradigms regarding the social relationship of robots and people. This includes a means of communication that aligns to the communication partner and the context

of the interaction. This exposes a need for a representation that comprises interlocutor-specific and context-specific semantic knowledge — the *situation model*.

Now the question is: Which information should be available in a *situation model*, and how should this information be represented? Also, which mechanisms are needed in order to apply the knowledge in real-world situations? These questions outline the work I will present in this thesis, though it is not possible to answer them in a comprehensive way. Instead I will take a closer look at three different aspects of building a consistent *situation model*. These aspects focus on the space, time, and protagonist dimensions of Zwaan and Radvansky (1998)'s definition. The causality and intentionality dimensions will only be covered marginally in the enclosing high-level applications.

The basis for such a model is a geometric description of the surroundings. I will explore possibilities for representing the data in a way that allows an appropriate level of detail for the task at hand and enables inference about the functional roles of certain structures through observation. The aspect of learning (in terms of knowledge acquisition) is very important for successful generation of an adaptive model. This is also true for the interpretation of the surrounding that can not directly be inferred from observation. However, it is important for a personal robot to also have semantic knowledge about different areas of its working environment in order to act appropriately. In order to enrich the *situation model* with the according information, I will present an approach for applying semantics to the enclosed areas of an apartment. Nevertheless, a situation model should not consist purely of visually perceived information. The communication with a human interlocutor provides useful information as well. Not only about the situation itself, but also about the way this information is represented in the interlocutor's mental *situation model*. In order to align to the partner on a communicative level, it is important to establish methods to access the *situation model* in a way that supports this alignment process. This thesis is embedded in the research program of the collaborative research cluster called "Alignment in Communication" at Bielefeld University. The program involves many interdisciplinary projects which collaborate to reach two goals: different kinds of alignment phenomena and their implications on conversation and *situation models*. Wachsmuth et al. (2013) give an overview of selected topics within the cluster.

1.1. Robot Companions in the Home

There were many robotic platforms developed in recent years that aim to lead the way for future personal robot companions. Many of them focus on technical design and appearance in order to support the research on motion and *HRI*, like the adult-sized futuristic looking robot HRP-4 (Kaneko et al., 2011), the infant-sized iCub (Metta et al., 2010), and the anthropomorphic robot head Flobi (Lütkebohle et al., 2010).

There are a few robots that made the transition into the real-world like the impressive robotic car Stanley, winner of the DARPA Grand Challenge developed by Thrun et al. (2006) or the TOOMAS shopping guide (Gross et al., 2009). However, all of these robots have a very distinct task to fulfill and their hardware and software design is highly optimized for the task.

Other obvious examples are vacuum cleaning, floor washing, and lawn mowing robots that have been available for purchase for several years now. But there are also commercial robots on the market that serve still a very limited, but social function. The robotic seal Paro is used in care facilities with elderly people or other patients in order to increase their social interaction, similar to animal-assisted therapy (Wada and Shibata, 2007). Comparable effects were found with the toy dinosaur Pleo in children's play (Fernaeus et al., 2010).

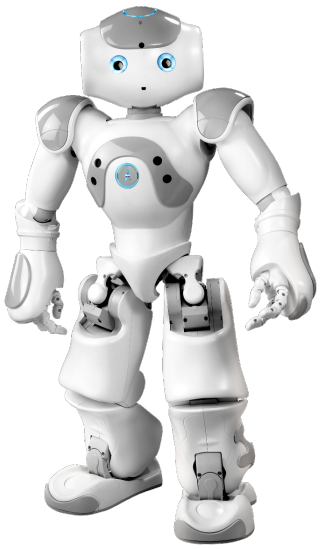


Figure 1.1.: Nao by Aldebaran Robotics. Image taken from: Bader et al. (2013).

Another commercially available robot is the Nao by Aldebaran Robotics (Gouaillier et al., 2009) (see Figure 1.1). It has been designed for a much wider range of applications than the afore-mentioned robots. In practice, however, it is mostly used as a toy or a research platform because the software still lacks essential abilities to truly understand its surroundings and its communication partner.

Most of the basic research in robotics is done using platforms not designed for end users, but to support the research itself. The first generation of our

research platform, *Bielefeld Robot Companion (BIRON)*, was introduced by Haasch et al. (2004). It was a modified PeopleBot from ActiveMedia equipped with a pan-tilt camera, a pair of microphones, and a laser range finder (see Figure 1.2).

At the time, it was used in a *home tour* scenario which involved sensing of humans (Fritsch et al., 2004), sensing the environment using the laser range finder for obstacle avoidance, and recognition of human speech (Wachsmuth et al., 1998) coupled with a basic dialog management system. Based on contemporary standards the *home tour* scenario is a comparatively simple challenge. The robot behaves purely reactively. It passively follows a human to new locations and is introduced to new facts about the environment, which basically consist of references from labels to coordinates. It does not learn anything new about its environment except when notified by the human.

Figure 1.2.: First generation of *BIRON*.

Today comparable research projects go beyond the *home tour* scenario and progress to more complex scenarios. Those either require a more sophisticated *situation model*, more powerful perception, or a dialog system that handles more complex interactions. Further, most projects involve a pro-active robot behavior like in the scenario for Dora The Explorer, first introduced by Hawes et al. (2010). Dora is driven by a motivational system that triggers an active exploration behavior to fill gaps in the spatial knowledge of the environment. Meanwhile, the robot tries to do a categorical labeling of rooms by analyzing functionally important objects, as well as considering ontology-driven inference on the results of this uninformed search. The architecture is composed of reactive goal generators which create new goals that pass a collection of filters for a first selection step. A management mechanism then determines which of the remaining goals to pursue (Sjoo et al., 2010). It contains goal generators for frontier-based exploration, view planning, and a visual search using the pan-tilt-zoom camera. The spatial information is stored using a framework

for cognitive spatial mapping (Pronobis et al., 2009). The map is assembled by so-called "places" that define the spatial relations representing the structure of the environment. A "place" is a collection of arbitrary distinctive features that can be complex or abstract in nature. Also, there is a concept called "scene" for segmentation of space and grouping of similar feature values. The map is only a topological representation of the environment and does not require a maintenance of a global spatial consistency.

Meger et al. (2010) pursued a similar goal with the visual searching platform Curious George. This robot won the first place in the 2007 and 2008 robot league of the Semantic Robot Vision Challenge (Helmer et al., 2009). As the competition requires the robots to identify the objects from instantly-learned categories using web imagery, Curious George is able to download the required data from web services like Google Image Search. For exploration, the team implemented a frontier-based strategy as proposed by Yamauchi (1997). For visual search they do not use a 2D occupancy grid, but a 3D representation of the environment, the result of a horizontal surface-finding algorithm developed by Rusu et al. (2009d) as a package for the Robot Operating System (ROS) (Quigley et al., 2009). As an attention system, they implemented the *saliency map* approach proposed by Itti et al. (1998).

One of the most advanced robot platforms in terms of real-world applicability in household scenarios is probably Cosero (Stückler et al., 2014) (see Figure 1.3). The team NimbRo@Home from the University of Bonn won the RoboCup@Home competition (Wisspeintner et al., 2009) in 2011, 2012, and 2013 using the Cosero platform and its predecessor Dynamaid. It is equipped with a height-adjustable torso on an omni-directionally moving base and two anthropomorphic arms. The human-like appearance is meant to support *HRI*. To represent the environment, the deployed system uses a global occupancy map refined through the so-called 3D surfel grid approach (Stückler and Behnke, 2014). This global representation is used mainly for planning in navigation, while an egocentric 3D representa-

Figure 1.3.: Cosero

tion of the current situation is used mainly for local planning and grasping. For people awareness they augment the global environment representation with person hypotheses (Stückler and Behnke, 2011) which in turn profits from semantic knowledge about the surrounding structure retrieved from this representation.

Although these robots are already quite sophisticated, they still lack a knowledge representation that is powerful enough to handle future tasks of a truly personal service robot. The current representations do not generalize to arbitrarily different tasks than those described in the research publications.

Further, large parts of the gathered information is not preserved long-term. Most of it is only kept for intermediate usage and only very high-level representations are preserved for later reference (Dora The Explorer is an exception here). Another shortcoming of the robotic systems described here is that there seem to be no strategies to align the *situation model* to the communication partner. This is certainly a requirement for future personal robots. It is impractical to keep up the command-like communication pattern that current artificial systems require in order to understand the interlocutor. *HRI* will be based on natural language in the future, which will require alignment strategies in the robotic systems that are able to match internal representations to instances of differently represented information.

That a robot like Cosero, which lacks these abilities, is so successful in the RoboCup@Home competition shows that research must still evolve in these areas. The tasks assigned to the robots are not designed to require such abilities[1]. This is probably because research has not come far enough yet for enabling the participating teams to perform real natural language *HRI* or accessing a generalizable multi-purpose knowledge base. The "Enduring General Purpose Service Robot" task goes in this direction, but from personal experience, I can report that the last years' commands were all solvable using standard tools. Most tasks require pre-knowledge of allocentric information like labeled locations and areas. Accordingly, the majority — if not all — allocentric knowledge is provided beforehand and the robot just has to build up egocentric representations to carry out commands at specific locations.

[1]The 2014 rulebook for the RoboCup@Home competition can be found at `http://www.robocupathome.org/rules` (visited March 1, 2015)

It seems there is a lack of widespread, functioning solutions for an integrated approach to gathering and maintaining knowledge with varying spatial scope. This is one of the reasons why it might be valuable to shift the attention in research from processing robotics problems in the easily-perceivable space in the direct vicinity of the robot to a more comprehensive view of the wider environment.

1.2. From Vista Space to Environmental Space

In psychology, it is mutually agreed that cognitive functions differ when applied to different scales of space, as discussed by Montello (1993). He argues that in human psychology the representation of space is scale-dependent. Applied to actual tasks this means that comparably small scenes such as those in manipulation tasks are represented differently than those in tasks like navigating to another room, which requires representation of a much wider area. Montello (1993) distinguishes four major classes of psychological spaces. The *figural space* is "projectively smaller than the body and can be directly perceived from one place without appreciable locomotion". The *vista space* is "projectively as large or larger than the body but can be visually apprehended from a single place without appreciable locomotion". The *environmental space* is "projectively larger than the body and surrounds it. It is too large to apprehend directly without considerable locomotion". Usually it requires the integration of information over a significant period of time to fully perceive this space. *Geographical space* is "projectively much larger than the body and cannot be apprehended directly through locomotion".

For robotics, this means that it might also be advantageous to make a similar distinction. In previous work, Swadzba (2011) explored ways to model the *vista space* of a mobile robot. In this thesis, I will proceed to a more comprehensive view of the spaces relevant for a personal robot in an apartment environment. However, although I will present approaches for integrating representations of different scopes, the focus will lie on *vista space* and *environmental space* representations. *Geographical space* is out of scope for a domestic service robot, and *figural space* is explicitly covered by another project within the collaborative research cluster in which this thesis is embedded (cf. Meier et al., 2011; Li et al., 2012).

As the representation of space should be applied in a real-world scenario,

it must be handled as a continuous model, although the different scopes are represented in different ways. It is impractical to model scopes of a scene in completely isolated representations preventing a bidirectional interchange or collaboration.

Ruetschi and Timpf (2005) argue for a similar distinction between spaces with different scopes in the real-world scenario of wayfinding in public transport. They found that humans use the *network space*, which is "a mediated space, presenting itself by means of maps and schedules, but also by audible announcements and tardiness. It exhibits a network structure". In addition, they use the *scene space*, which is "directly experienced but documented only implicitly and within itself. [...] It exhibits a hierarchical structure". These spaces have a *geographical scale* and an *environmental* or *vista scale*, respectively, following Montello's definition. Further, they state that *network space* and *scene space* are linked in many ways and interact closely in the application domain of public transport.

So in addition to the definitions mentioned in Section 1, a *situation model* should support representation of different scopes of the space surrounding an agent in a scope-dependent, but continuous way. A robotic system that implements such a *situation model* needs strategies for cooperation with different scope-dependent representations.

1.3. Research Questions

Although we have seen many very impressive performances of robots in recent years in a wide spectrum of application scenarios, when looking at an individual system, the capabilities are very limited. Usually, complete systems are designed to function in exactly one scenario. They may represent the optimal system for solving the task at hand, but apart from that, most systems are completely useless. There are different reasons for this. One is certainly that many researchers perform basic research on delimited fields, which is good because most basic capabilities a true personal robot requires are still far from solved. Another aspect might be the lack of applicability of many software components to realistic circumstances, or to some extent, the inability to adapt to situations other than the one they were optimized for. This leads to another aspect that is typically underestimated: The integration of context to analyses and mechanisms. An object recognition

component could largely profit from knowing which functional role the currently perceived scene has, and a pro-active knowledge gathering behavior might not be appropriate in the middle of the night.

With the advancements in available functionality, middleware implementations and system coordination approaches over the last years, more and more focus is applied to system integration aspects. With the availability of more complex (in terms of number of available functions) and more compatible systems, there is also a growing demand for more unary solutions rather than multiple island solutions — especially in knowledge representation techniques. This demand leads to the following research questions which represent the basic skeleton of this work.

Question 1: How to represent spatial knowledge?
Which frames of reference should be used (egocentric, global)? How can structural information or instances and their relations be represented? Which data structures should be used? How can world knowledge and inferred knowledge be combined?

Question 2: What and when to represent?
Which level of detail should be applied and how does this depend on the situation? How can the relevance of certain data be judged before insertion?

Question 3: How to solve temporal integration?
How does the update process work? Which additional dimensions are required in the situation model? How can the temporal aspect of the representation be exploited?

Question 4: How to include context information?
How do components benefit from context knowledge? How can background knowledge be referenced on later occasion? Can the spatial layout of the knowledge representation facilitate the selection of peripheral information?

All of these questions need to be answered when trying to build a *situation model* for a personal robot companion that is applicable in arbitrary situations. Certainly there are more aspects to this topic (semantic ontologies,

logical inference, intentions, etc.), which cannot be answered in the scope of this thesis. These questions indicate, however, that the pure representation is not the only key to a successful *situation model* development, but also that strategies for handling data processing are required.

Nevertheless, the posed questions underly the work described in this thesis, and in the following chapters, I will propose answers to these questions.

1.4. Scenario & System Foundation

The different aspects covered in the following chapters will be linked to a mutual scenario in order to demonstrate the various applications of the described solutions. I will refer to this scenario as the *lost key scenario*. It is an analogy of a situation in which a person can not remember where she/he last placed a key ring, asking someone for help finding it. In concrete terms, the mobile service robot of a homeowner observes the actions and utterances in its surroundings to build up a *situation model*. For this, it pro-actively moves around the apartment and closely inspects presumably relevant events or locations. At some point, when someone asks it about a certain object, it is able to report the location or the last performed manipulation of the target object. This scenario requires the afore-mentioned aspects investigated in this thesis. It requires a *situation model* implementation with long-term capabilities in representing distinct structures and actions. To maximize the informative content and minimize the effort, the robot must select the most relevant events to observe and represent those in an efficient way. It must be able to link possibly ambiguous verbal references to spatial structures by aligning descriptions to actions and to the *situation model*. Further, a verbalization of the found results must be available which supports the alignment to the communication partner. The hardware and software components needed to enable such a scenario as a prerequisite for the implementations done for this thesis will now be explained.

The Hardware Platform

The *Bielefeld Robot Companion V2 (BIRON II)* hardware platform (see Figure 1.4) we use, based on the research platform GuiaBot™ by Adept

MobileRobots[2], is customized and equipped with sensors that allow analysis of the current situation in a human-robot interaction. The platform used here is the second generation of the *BIRON* platform series, which has been continuously developed since 2001. It comprises two piggyback laptops to provide the computational power and to achieve a system running autonomously and in real-time for HRI. The robot base is a PatrolBot™ which is 59cm in length, 48cm in width, and weighs approx. 45 kilograms with batteries. It is maneuverable with 1.7 meters per second maximum translation and 300+ degrees rotation per second. The drive is a two-wheel differential drive with two passive rear casters for balance. Inside the base, there are two laser range finders that add up to a 360° degree laser scan with a scanning height of 30cm above the floor. To control the base and solve navigational tasks, we rely on the ROS navigation stack[3].

For room perception, gesture recognition and 3D object recognition, the robot has two ASUS Xtion Pro Live RGB-D sensors[4] for real time 3D image data acquisition: one facing down (objects) and an additional one facing towards the user/environment. The object recognition system is supported though high quality 2D imagery from a Sony Alpha 5100 consumer camera. A high-resolution webcam is used for facial recognition. The corresponding computer vision components rely on implementations from Open Source libraries like OpenCV[5] and PCL[6].

Additionally, the robot is equipped with the Katana IPR 5 degrees-of-freedom (DOF) arm; a small and lightweight manipulator driven by 6 DC-Motors with integrated digital position encoders. The end-effector is a sensor-gripper with distance and touch sensors (6 inside, 4 outside) allowing it to grasp and manipulate objects up to 400 grams throughout the arm's envelope of operation. The on-board microphone has a hyper-cardioid polar pattern and is mounted on top of the upper part of the robot. For speech recognition and synthesis, we use the Open Source toolkits CMU Sphinx[7] and MARY TTS[8]. The upper section of the robot also houses a touch screen

[2] `http://www.mobilerobots.com/`, (visited: March 1, 2015)
[3] `http://wiki.ros.org/navigation`, (visited: March 1, 2015)
[4] `http://www.asus.com/de/Multimedia/Xtion_PRO_LIVE/` (visited: March 1, 2015)
[5] `http://opencv.org/` (visited: March 1, 2015)
[6] `http://pointclouds.org/` (visited: March 1, 2015)
[7] `http://cmusphinx.sourceforge.net/` (visited: March 1, 2015)
[8] `http://mary.dfki.de/` (visited: March 1, 2015)

($\approx$ 15in) as well as the system speaker. The overall height is approximately 140cm.

For real-world applications, the robot can be deployed in a laboratory apartment in the new CITEC building of Bielefeld University. This so-called *Intelligent Apartment* measures 60 square meters and has three rooms, including a kitchen, a living room, a gym, and a bathroom. It contains plenty of hidden technology, but looks like a regular apartment.

Figure 1.4.: *BIRON II.*

System Architecture

To model the robot behavior in a flexible manner, we use the *Biron Sensor and Actuator Interface (BonSAI)* framework. It is a domain-specific library built on the concept of *sensors* and *actuators* that allow the linking of perception to action (Siepmann and Wachsmuth, 2011). These are organized into robot *skills* that exploit certain *strategies* for informed decision making (Lohse et al., 2013). *BonSAI* supports modeling of the control-flow using State Chart XML. The coordination engine serves as a sequencer for the overall system by executing *BonSAI skills* to construct the desired robot behavior. This allows the robot to separate the execution of the skills from the data structures they facilitate thus increasing the re-usability of the skills.

The robot's architecture relies on the lightweight and flexible middleware *Robotics Service Bus (RSB)* for inter-component communication (Wienke and Wrede, 2011). *RSB*-enabled components communicate using a message-oriented, event-driven pattern over a logically unified bus that is organized through hierarchical scopes.

1.5. Outline

The thesis is structured as follows: Chapter 2 takes a more detailed look at the problem at hand. I will analyze the research questions, formulating actual approaches to be further developed in subsequent chapters. Also the main contributions of this thesis will be formulated. After this I will proceed semantically, starting with basic requirements and ending with complex conversational aspects. Chapter 3 deals with the structural representation of the robot's surroundings (Research Question 1). The chapter will discuss spatial and temporal integration aspects from the research questions (mainly Questions 2 and 3) and present a pro-active robot behavior that utilizes the developed models. A more semantic view of the surrounding structures is employed in Chapter 4. Here, a grounding of certain entities and areas to general semantic categories which expose certain functional properties is described. Further, the integration of peripheral information into the decision making process will be explored. This mainly refers to Research Questions 3 and 4. Chapter 5 takes a look at instances and their relations in the *situation model*, suggesting how to align these to the model of a interlocutor in a communicative situation (Research Question 1). It also demonstrates how different cognitive functions can benefit from one another, mainly contributing to Question 4. Finally, I will give a closing overview in Chapter 6 including a discussion of the approached contribution and an outlook to future perspectives for implementing *situation models* for personal robots.

Chapter 2

Analysis and Statement of the Research Problem

In the introduction I promised to explore ways of representing a *situation model* in an artificial robotic system. As stated above, there are many aspects to this topic that cannot all be answered comprehensively. For this thesis, I chose referential communication as a guiding theme to develop representations and strategies to contribute to a universal *situation model*. In concrete terms, the following chapters cover aspects that enable an interactive robot to ground references to specific objects in a scene in multiple modalities. For that reason, it gains spatial knowledge as a persistent model in a way that allows it to ground these references. Mainly three different aspects of building a consistent model will be pursued: Representing the spatial layout of the environment, applying semantics to geometric structures and areas of the environment, and deploying and aligning the model in human-robot interaction. This chapter aims to analyze the implications of the different available choices regarding these three aspects. It also contains conclusions from experiences that had been made during the work with mobile robots by myself and others. By dealing with these questions, a more fine-grained statement about what the research problem is will emerge. Ultimately, this analysis leads to specifically formulated goals and contributions of this thesis.

2.1. Functional Requirements

With the vision of a multi-purpose, generally-deployable robotic system in mind, it becomes quite clear that, to build a coherent system, it is unrealistic to just combine the many island solutions that currently exist for isolated problems. Today many researchers focus on their specific problems and invent representations that perfectly fit their requirements. This leads to a variety of very distinct solutions with no avenue of collaboration or exchange, instead resulting in huge overhead to maintain all the different representations. It would therefore be desirable to build a central, comprehensive representation that can be used by all components of the robotic system. The problem is that the requirements of the solutions for the various problems of such a general-purpose system are very divergent. It would require a very flexible and powerful representation with many support mechanisms to serve all the posed requirements. A thesis like this can not claim to find the ultimate solution for this problem, but I do propose a representation designed to function as a basis for several software components in a robotic system. Considering the guiding theme of this thesis, components enabling referential communication will be considered for design decisions. Further it may be extended to other problems not considered in this thesis.

But what are the general functional requirements of such a representation apart from the task-specific ones? First, it must grant direct access to the data. This means a component must be able to easily receive the required data without having to transform or remap the information in order to fit them to the internally used format. To a certain degree, this also requires the components to adjust their formats to those supported by the central representation. Otherwise, the divergence in the representations on component level would just be transferred to the central representation, yielding no gain for the system.

Further, the representation must support the efficient analysis of the data. The representation's data structures must be chosen so that the components can implement fast algorithms on them. However, if multiple data structures are maintained, they also need to be closely linked to quickly transfer data.

The representation itself needs to be resource-efficient to guarantee low latency when components access the data. This means that transformation or search tasks within the descriptions must be implemented efficiently. This calls for a sparse representation of the spatial data. The implementation

should support the representation of every kind of data in a level of detail appropriate for the task at hand. This reduces the memory load and thereby the latency.

2.2. The Choice of Scope

One of the most obvious differences in the many representations used across different components is the scope of the spatial description. In object manipulation tasks, the spatial representation has a totally different scope than in a path-planning task for navigation. In general, they can be divided into an *allocentric scope*, which defines a view on the scene from a global perspective, and the *egocentric scope*, which defines a view from the personal perspective. The latter supports a description of the scene relative to the point of view of the agent that generates it, usually representing the field of view of the perceptual system. Whereas the allocentric scope may support representations of the complete known environment from a global coordinate system, or just a subspace (e.g. the intermediate surrounding of the robot).

From participating in several RoboCup@Home[1] competitions, I can report that nearly all competing robotic systems use an allocentric representation for long-distance navigation and storing positions of relevant objects and locations. Meanwhile, obstacle avoidance and manipulation tasks are nearly exclusively done exploiting the egocentric scope. This is not surprising because localization tasks usually profit from relating entities (including the self) to landmarks, which is particularly convenient in allocentric representations. On the other hand, avoidance and manipulation tasks rely on the relation of the self to structures in the immediate environment. For this, egocentric representations are most suitable.

A general representation should cope with these different scopes. It must enable the components to choose the scope of their spatial descriptions, but must also maintain links and relation between allocentric and egocentric views. The argument for a sparse representation applies here as well. Not every part of the environment needs to be represented egocentrically, and less so from multiple points of view. Similarly, it may not be necessary to

[1] `http://www.robocupathome.org/`

keep the egocentric representations updated all the time. Depending on the task at hand, they may only become relevant in certain situations.

2.3. Knowledge Representation

In order to identify a preliminary set of data structures for the general spatial representation scheme, I will have a look at the software components currently running on the *BIRON II* platform (see Section 1.4). One of the most fundamental components of a mobile robot's system is the navigation. For localization and mapping of landmarks in the form of physical obstacles, it uses a probabilistic occupancy grid representation of obstacles in the environment (Moravec, 1988). It is an allocentric representation that depicts the spatial layout of the complete environment known to the robot. A semantic mapping approach for probabilistically labeling areas in the environment based on certain semantic properties uses a similar representation (Ziegler, 2010). The resolution of those representations is adjustable, but in practice, a rather coarse resolution is chosen (usually $\sim$ 5cm cell size) because the associated tasks do not require more detail.

For a persistent storage of locations and objects, the system uses a plain database containing descriptions of the entities in global coordinates that relate to the occupancy grid representation.

The person tracking component contains an allocentric representation, as well. Person hypotheses are also maintained on an instance level with global coordinates. However, these hypotheses are fed with information from detectors building upon egocentric representations. Human torsos are detected using an egocentric *point cloud* representation from a depth camera. Legs are detected from a polar representation of distance measures from a laser scanner.

More egocentric representations are used in recognition and manipulation. Geometric analyses for finding candidates and obstacles for grasping also use a *point cloud* representation from a depth camera. However, the content of the the *point cloud* has a higher resolution than that to detect torsos and is limited to the maximum range of the robot's arm, whereas the torso detector needs a significantly longer sight. The visual object recognition component cooperates closely with the 3D geometrical analysis component and works on 2D imagery taken from the robot's visual sensors.

Summarizing these insights, one can identify a set of data structures that would satisfy the demands of most components of the current *BIRON II* system.

Allocentric areal representation. This could be a probabilistic grid structure or, alternatively, a hierarchical quadtree representation (Hunter and Steiglitz, 1979). A three-dimensional voxel grid or octree (Meagher, 1982) representation would be imaginable, as well.

Allocentric instance representation. In the current *BIRON* system this is just a plain database of instance descriptions, but a network structure would be possible as well.

Egocentric areal representation. An obvious data structure for this is a *point cloud* or depth image.

This selection of data structures deliberately misses representations for the 2D imagery and polar range descriptions mentioned above. There are several reasons for this. First of all, a general representation needs to represent the lowest common denominator for all named requirements. But it certainly cannot universally manage all types of representations that used internally across the components of a system. A compromise must be found. Secondly, the relevance of certain data structures in a central representation is proportionate to the persistence in their demands. Both data structures in negotiation require no persistence in the ways they are used in their respective software components; their data is processed and directly forgotten. Only the results of the analysis may be relevant for future reference, but these can be represented using the identified set of data structures. The same holds true for egocentric instance representations that may be relevant in the specific execution of, for example a manipulation task, but to persistently represent these instances, the egocentric frame is probably unnecessary. Nevertheless, if new requirements occur that demand persistence for these structures, an extension mechanism that allows one to link-in arbitrary structures to the default representation would be imaginable.

Persistence is a central topic for a general knowledge representation of a multi-purpose robotic system. It allows the system to use the representation as a spatial memory. The current *BIRON* system only has a limited spatial memory. As far as I can tell, this also applies for most artificial robotic

systems that participated in the RoboCup@Home competition in the last years. From my experience, it is sufficient for the robots to maintain the allocentric representations for later reference. Since there is no task where the robot has to re-visit a previously analyzed scene for a second time, there is no need to reference previously gathered egocentric knowledge at a later occasion. In situations when egocentric representations are required — like when grasping objects — the scene is analyzed bottom-up, and the data is discarded as soon as the robot finishes its task at this location. In real-world applications that go beyond those in the RoboCup competition, this is of greater importance. A robot needs to transfer knowledge from one location to a different situation in the future. This is especially important for information that cannot be re-generated in a bottom-up manner. Nonetheless, this also reduces the cognitive effort of the system by eliminating the need to repeatedly analyze the same scene from scratch.

For a general persistent knowledge representation, this means it needs to maintain several egocentric representations for later reference. They need to be linked in a way that allows the system to compare these models with each other and with the allocentric representation (cf. Section 2.1). This is particularly important for supporting the inference of referenced objects in communication. Further, methods for spatial and temporal integration, which are self-evident for allocentric representations, also need to be implemented for these egocentric structures.

2.4. Applying a Situation Model in Interaction

In the previous sections, I mainly discussed functional properties of a spatial representation for multi-purpose service robots. However in interaction situations, methodical aspects of such a representation become particularly relevant. In *HRI*, the communication is not purely auditory, although in many robotic systems the communication is limited to the speech modality. Similarly, the interpretation of action should not be purely visual. A robotic system that aims to understand humans in a way that promotes their acceptance in society will have to cope with multiple modalities in interactions. Referencing objects in a dialog is a common example of this. To correctly ground the sentence "*This* is the object I mean", the system needs to either interpret a gesture or it must know certain properties of the objects in the

near vicinity of the interlocutor in order to narrow down a probable target object. Similarly, to interpret the sentence, "I mean the chair in front of the cupboard", one requires a rough concept of which objects might be meant by "chair" and "cupboard", especially if the current scene contains multiple instances with these labels. Also the spatial relation meant by "in front of" needs to be interpreted regarding different perspectives or reference frames. Also the context of an interaction might be important for the correct interpretation. For example, the sentence, "Please bring me the book", might relate to the novel the interaction partner is currently reading and is located in the bookshelf — if this conversation takes place in the living room. However, if this sentence is said in the kitchen while cooking dinner, it might relate to the cookbook lying open on the table.

For general spatial representation, this means it must support the inference about multiple aspects represented in the system. For example, a component for gesture recognition that works on a 3D egocentric representation may also reference the allocentrically represented surrounding of the robot in order to correctly interpret the gesture.

Especially when grounding utterances, a close collaboration of the different representations is crucial for the success. Although the utterance was perceived correctly on a linguistic level, the content might still be ambiguous. Including semantic information about the context (e.g. in which room is the interaction and what is its function?) might improve the interpretation process. In referential communication "perspective-taking" is a key concept for enhancing the process of associating the described relations to instances. This involves both egocentric and allocentric representations. The same is true when using different reference frames in the descriptions. In turn, the production of signals to the interlocutor profits from close collaboration of the different representations in the same way. However, not only the representation is the key to successful resolving ambiguities. It requires a sophisticated algorithm, that can handle multiple hypotheses in a probabilistic way and include a variety of evidences in the process of finding the most likely interpretation of the ambiguous utterance. One of the evidences may also be a linguistic world knowledge regarding preferences of humans in speech production in various situations.

These arguments again promote a good interconnection of the different types of representations. It must be easy to transfer information from one representation into the other. More importantly, it becomes obvious that

references should play a significant role in a persistent spatial representation. Both spatial relations for interpreting speech and action, as well as temporal references to the status of a past scene to detect change, are of great importance. Only this allows inference about the dynamic properties of certain structures and, therefore, communication about events that were not directly observed.

2.5. Summary: Contribution of this Thesis

These analyses allow a more precise formulation of the topics explored in this thesis. This, in turn, helps define the specific goals to pursue. The research problems identified in the previous sections closely relate to the research questions identified from the semantic analysis of the *situation model* in Section 1.3.

Research Question 1 addresses the representation of spatial knowledge. As discussed in Section 2.1, a complex robotic system for multiple purposes should contain a central representation that handles the spatial information for the individual software components. This storage should represent the data sparsely to minimize computational overhead. Also, it should manage different types of representations that are well connected and allow components to use the type of representation that suites their algorithms best. These types differ in the spatial scope and the data structures they use. Specifically, three types of representations were identified: An areal- and an instance-based representation with an allocentric scope, and an egocentric representation for describing structures in the robot's field of view. Chapter 3 will explore the realization of such a representation.

The requirement that the representation should be sparse has implications on Research Question 2 (What and when to represent?). The existence of multiple types of representations within the central storage enables the developer to choose an appropriate level of details for the different representations. These can be chosen according to the application they are used for and the resolution required by the algorithm using it. As discussed before, there should be a set of egocentric representations in addition to the allocentric ones. Consequently, there need to be strategies that decide when new egocentric models need to be introduced and when they need to be merged or deleted. These will be discussed in Chapter 3.

As seen in the analysis of application in interaction, temporal integration and temporal references are crucial mechanisms for spatial representation in a robotic system. The same issue is addressed by Research Question 3. To detect change that is not directly observable, the maintenance of a history of certain structures or properties is important. In the detection process references to situations in the past will be established which need to be represented by the central spatial storage. This aspect will be discussed in Chapter 3. In Chapters 4 and 5, the aspect of spatial and temporal integration while updating the instance based representation will be discussed.

Research Question 4 focuses on the integration aspect of multiple types of data in the interpretation process. The analysis of the application in interaction suggests that the different representations need to collaborate closely to enable the interpretation process to integrate context data. A multi-cue classification process is described in Chapter 4 that relies on this collaboration and the interconnection aspect of the representation. This system explores a boosting-based classification approach that uses a variety of different features to classify different room types. It gathers a vast amount of visual cues and uses them to label different parts of the environment according to their function. Using the allocentric areal representation, these labels are published as context information for other interpretation processes. The system described in Chapter 5 does not focus so much on using the central spatial representation; it rather explores an approach for resolving ambiguities using a variety of evidences from multiple modalities. It incorporates an allocentric probabilistic network approach for tracking multiple hypotheses for interpretation.

Chapter 3

Partitioning the Workspace - Spatial and Temporal Integration of Informative Local Observations

For building an informative *situation model*, the first requirement is to have an idea of the general spatial structure surrounding the robot. This has already been discussed in the previous chapters. It does not necessarily mean that a complete detailed three dimensional representation of the environment needs to be tracked through 3D Slam or similar approaches. Several approaches for reconstructing a robot's environment have been presented which typically build up a comprehensive allocentric representation (cf. Wiemann, 2013). However, for specific atomic tasks like grasping an object the representation of spatial structures is often strictly limited to the relevant parts. Typically, only the target object and possible obstacles in the close neighborhood are represented in an egocentric fashion (cf. Rusu et al., 2009c). Particularly, in the field of domestic service robotics a large set of assumptions about the setting can be applied, for example about the size of the work space, the number of entities inside this space, structural properties of the floor and walls, etc. However, these might not be true for other fields of robotic research like rescue or outdoor scenarios. Depending on the task at hand it might suffice to know the rough layout – in domestic robotics of the apartment and a small set of more detailed areas, which are relevant for typical tasks and interactions. The scope of such a representation would be located between those of the two extremes described above.

On the one hand, it should support purposive vision, but for multiple tasks. On the other hand, it should also allow to be used for tasks with a more allocentric scope that require a minor level of detail. This chapter deals with the question of how to realize this kind of sparse spatial representation of the working environment of a domestic service robot. The considerations also include ideas for updating knowledge from spatial integration of different views and also temporal aspects of these integration measures.

Most commonly, the environment for mobile robots is represented in a form that is highly optimized for the task at hand. Besides, in most cases these tasks are tightly coupled with a specific robotic ability. For example, navigational tasks are almost exclusively solved with variants of *Simultaneous Localization and Mapping (SLAM)* (e.g. Leonard and Durrant-Whyte, 1991), which have proven to be a very effective solution to localization and navigation problems (Montemerlo et al., 2002; Grisetti et al., 2007; Kuo et al., 2009; Ma et al., 2009). One of the most prominent variants is the *Rao-Blackwellized Particle Filters (RBPF)* method (Murphy, 1999) in conjunction with an occupancy grid representation of obstacles in the environment (Moravec, 1988). Figure 3.1 shows an example of an occupancy grid generated by a *SLAM* application.

Figure 3.1.: An sample grid map generated using *SLAM*.

This approach works very well for compact platforms that solve – compared to everyday tasks of humans – relatively simple navigation tasks. But as platforms become more articulated and thereby more non-compact (e.g. WillowGarage PR2, Meka M1, Fraunhofer Care-O-Bot, see Sec. 1.1)

the existing solutions may not be suitable anymore for newly arising problems. Hornung et al. (2012) describe a complex task for the articulated mobile robot PR2 in which it has to navigate through a highly cluttered environment while carrying large objects. It thereby has to plan complex trajectories through narrow passageways considering its current body configuration to avoid collisions. Their approach utilizes a multi-layered 2D grid map which represents obstacles projected to various heights that are relevant for specific body parts of the articulated robot. These projected layers are created from a full 3D occupancy map represented as an efficient octree-based grid (Hornung et al., 2013). This is a pragmatic solution that uses existing, well tested techniques for solving a new problem. However, using this kind of allocentric representation for a task that involves locomotion with extensive examination of self-to-object relations, might not be the best choice. Collision checks are computationally more expensive and less precise than they could be when using a egocentric representation that only contains the direct neighborhood, but in a higher resolution. Depending on the arm control strategy this would possibly also reduce constant coordinate transformations.

For representing space for navigation robotics has focused on using global coordinate systems. Apart from the just mentioned method there are several other solutions for 2D mapping and also different approaches for modeling compact representations of the environment in 3D. These include volumetric representations (Thrun et al., 2000; Nguyen et al., 2007), as well as raw *point cloud* representations with overlayed octree formalisms for efficient search (Nüchter et al., 2007).

Other approaches that do not only consider navigational goals but also higher interaction tasks that involve planning, overlay the raw structural information about the environment with an hierarchical but allocentric semantic representation. Zender et al. (2008) anchor nodes of a conceptual graph structure in a topological map of the known environment, Philippsen et al. (2009) define a *Mobile Manipulation Database*, which is a large graph structure representing the world as objects, locations and properties connected by topological and semantic links.

In robotics, the role of egocentric models as a means of universal representation of spatial information acquired through locomotion has been underestimated so far. Currently they are only used as task-specific representations in domains like grasping, obstacle avoidance or attention. Most

of the previous work that combines allocentric tasks with problems that were classically solved with egocentric methods choose a unified allocentric representation. There are several semantic map approaches like Nüchter and Hertzberg (2008). They build up a complete 3D representation of the whole environment and use it for navigation and detection of objects alike. Vasudevan et al. (2007) describe a hierarchical solution containing an allocentric topological representation of places combined with local probabilistic object graphs. Elfring et al. (2012) introduces a probabilistic world model that anchors instances allocentrically and tracks them over time.

In this chapter I will be approaching the question of how to partition the workspace of a mobile robot in a way that gives semantic meaning to relevant structures and thereby explore a way of representing those structures in an egocentric way to facilitate self motion. However, these egocentric models will still be a part of a more enclosing spatial representation that relates these models to an allocentric description of the environment. Therefore I will first discuss spatial representations in humans and machines in more detail. Subsequently, the set of semantic roles I will focus on in this context will be described. This also includes the first naive computational model realizing the assignment of roles to structures. After that, these deliberations will be incorporated in a more application centered system view. This includes anchoring the egocentric models in an allocentric world model as well as computational strategies for utilization of the sophisticated model in real-world applications. Finally, some results from systematic evaluations will be presented.

3.1. The Geometric Foundation

In order to find a valuable strategy for representing spatial knowledge in an artificial system it might be useful to understand the spatial representations in the human mind. Using this information as an inspiration, I will formulate a proposal for an enclosing spatial representation system and analyze ways for populating it with useful spatial knowledge.

Spatial Representation in Human Cognition

While allocentric representations play a major role in navigation and spatial memory of humans (Burgess et al., 2004), egocentric representations have a

special role in self-motion as well, as found by Hartley et al. (2004). They argue that allocentric representations in human spatial memory could not be built-up or acted upon without interaction with egocentric systems. Wang and Simons (1999) show that visual tasks are significantly impaired if self-motion is disturbed, e.g. by causing visual change through moving a human subject in a wheelchair. They suggest that

> "the representation [...] of viewpoint changes is not environment-centered. The representation must be viewer-centered and the difference between observer and display movements results from a difference in the nature of the transformation. Apparently, view-dependent layout representations are transformed or updated using extra-retinal information to account for observer movements."

The encoding of this spatial information in the human mind has been studied by Mou et al. (2006). They show that in persistent encoding both egocentric and allocentric representations play an important role. Allocentric representations relate objects to visual landmarks, the egocentric subsystem computes and represents self-to-object relations, which are also used for locomotion, especially when the allocentric information are inaccurate.

Other findings in human cognition research suggest that we as humans consider egocentric representations of obstacles in our immediate vicinity for interaction. Wang and Spelke (2000) argue that

> "human navigation [...] depends on the active transformation of a representation of the positions of targets relative to the self."

They conducted several pointing experiments with human subjects, testing different conditions while the subjects were remaining oriented or were being disoriented. From these experiments they conclude that the distance and direction of target objects in *intermediate-sized environments* is represented in an egocentric way and is updated over locomotion. However, in addition there seems to be an enduring allocentric representation of environment geometry which is used for (re-)orientation.

Research on human spatial working memory also suggests the existence of allocentric and egocentric systems that collaborate in spatial tasks, regardless whether these tasks are based on visual or auditory perception

(Stark, 1996; Roskos-Ewoldsen et al., 1998; Hartley et al., 2004; Lehnert and Zimmer, 2006). Further, it is common sense in brain research that several regions in the brain take on different representations for the execution of complex tasks. While the hippocampus provides an allocentric spatial map of the environment (O'Keefe and Dostrovsky, 1971; Squire, 1992), the parietal and prefrontal cortex are presumed to process egocentric spatial information (Stein, 1989; Colby and Goldberg, 1999; Lee and Kesner, 2003).

Enclosing Spatial Representations

These findings at least suggest that a heterogeneous representation of the geometric structures and their properties surrounding an agent might be a valuable approach for robotic systems as well. As seen in Chapter 2 this also makes sense from a functional and application-oriented point of view. There are approaches to this already being used in existing systems. But it seems that the conjunction of the different representations is often not modeled explicitly or even not at all. Also, egocentric representations are mostly of a short-term nature, so that they only exist in the moments of benefit. They are not preserved for later use.

When we go back to the PR2 system presented in Hornung et al. (2012) (the complete system is better described in Chitta et al. (2012)) it becomes clear that the actual grasping task is executed in an egocentric way, but is completely decoupled from the navigation system which works allocentrically. The system creates an egocentric representation of the tabletop scenario in front just when it starts executing the grasping task. The only exchange between the two representations mentioned is the transfer of recognized object models from the egocentric representation to an environment-centered semantic representation of known entities which is based on the navigation map. However, as already discussed, all navigational tasks are purely allocentric, regardless whether it is a long-distance planning tasks, or a short-distance 3D collision avoidance task.

In Ziegler (2010) an allocentric semantic map representation has been explored, which has been used for navigation and attention tasks in a unified hierarchical representation, also containing global object locations which have been egocentrically detected. Regardless of the representation paradigms that have been used in the different approaches, it could be shown that robotic systems can benefit strongly from semantic knowledge about

certain areas or structures in the environment. This is particularly true for the object searching task described in the work mentioned above, but one can think of many other applications as well (clean up tasks, inference in planning tasks, grounding in referential communication).

In this chapter I argue for a closer integration of allocentric and egocentric representations in an enclosing structure. This approach will be developed further in the following sections (specifically Section 3.3).

Populating the Spatial Representation

In order to generate knowledge about the environment to facilitate these integrated robotic strategies, much work has been done in segmenting objects from a scene based on classical tracking or classification approaches. Many of those applications expect a detailed model (e.g. CAD) of the object to be segmented (Albrecht and Wiemann, 2011). Algorithms pursuing this approach make use of decomposition (Gelfand and Guibas, 2004) of the scene, rely on local hierarchical features combined with a matching algorithm (Steder et al., 2009), or combine 3D perception for detection and 2D vision for recognition (Pangercic et al., 2011). However, there has been done research done on segmenting structures with more instance independent, but category specific properties (Sturm et al., 2010). This approach can apply certain movement properties to structures in the environment just by observation. The authors thereby utilize a fixed set of template models for observed tracks in order to classify fairly specific movement models. This requires pre-learned knowledge of the world.

In contrast, there are other approaches targeting a bottom-up segmentation of the scene, rather than a semantic interpretation of the objects. Campbell et al. (2010) suggest a model-free approach for segmenting and 3D model creation through observation of multiple frames. Their approach expects the object to stand still while the camera is actively moved around the object. This might be relevant for certain scenarios in which the robot pro-actively explores its environment, but in other scenarios it may be feasible to employ passive observation to distinguish for example background from foreground structures.

Traditional background subtraction algorithms apply the assumption that the static background does not change over time to identify moving objects by detecting changes in the scene. For example Sheikh et al. (2009) de-

scribe a sophisticated background subtraction algorithm that can be applied on freely moving systems like robots, which analyses trajectories of salient features over time. However, these approaches only detect moving objects, whereas for many scenarios movable objects as described by Sanders et al. (2002) are of greater interest.

3.2. Detecting Roles in Articulated Scenes

Knowing the geometric structure of the environment is not enough for interaction with and communication about objects in the scene. For human agents and embodied artificial agents alike, the segmentation of the scenery into meaningful parts is crucial for dealing with unknown environments. The system described in the following applies dynamic properties to partial structures of the scene just by observation. No world knowledge or explicit teaching is required. For typical tasks in unknown environments a mobile robot needs to detect and track other agents as possible interaction partners or for the awareness that these do not represent permanent insurmountable obstacles. Further, the robot needs to know the parts of the scene that are relocatable because these are typical subjects of conversation and manipulation. In order to extract these information from the scene, the system uses the *Articulated Scene Model (ASM)* first proposed by Swadzba et al. (2010). Instead of building a complex ontology of specific items in the environment and equipping the robot with strong detectors to apply sets of attributes to these items, the model learns a pixel-wise labeling of the observed structure (see Figure 3.2). Representations of instances can be inferred from these structures in a post-processing step. The *ASM* enables the system to gain spatial awareness in a bottom-up manner through observation. It builds up a three-layered scene representation:

Definition 1 (Articulated Scene Model)
A separation of an observed scene into three distinct layers:

Static Scene *Structures of the scene that ultimately limit the view as static scene parts and will not move and therefore not allow a farther perception. Geometric background structures like walls and large furniture.*

Movable Parts *Parts of objects that can be relocated through manipulation by other agents and allow a farther perception when moved. This also includes smaller furniture and doors.*

Dynamic Parts *Agents, regardless whether human or artificial that can move by themselves.*

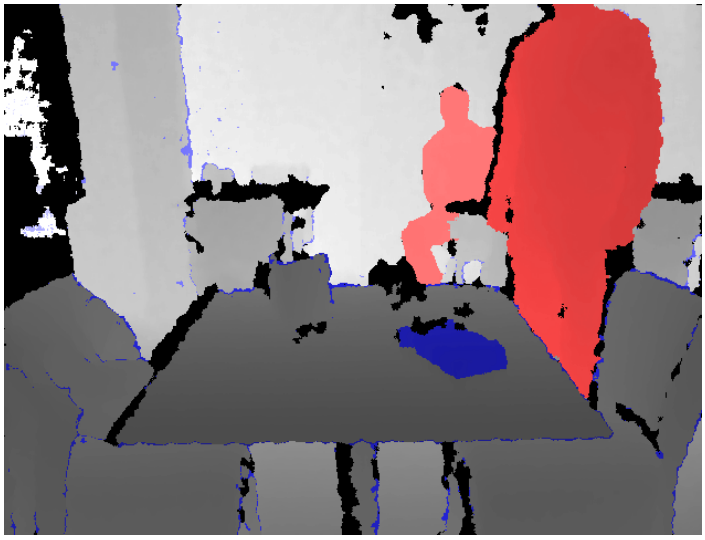


Figure 3.2.: Articulated Scene Model. **Red**: Dynamic Entities. **Blue**: Movable Objects. **Gray**: Static Scene.

3.2.1. Benefits of Observed Functional Roles

In an unknown situation a mobile service robot needs at least some a-priori knowledge about certain structures in order to find its way around the apartment. Since the purpose for service robots is to assist humans in their households, the unknown scene presumably contains a human who has to be detected in order to be able to communicate. Further, one typical task in an unknown environment might be the *home tour* scenario in which the human tells the robot labels for certain objects in the apartment. But there are also a lot of other tasks one can think of that involve reference of objects

in the environment that the robot does not know yet. In these situations the system can use the observed functional roles of the movable and dynamic parts to infer the position of the human interlocutor and the referenced objects.

Also pro-active behaviors of the robot can benefit from these functional roles. These enable the system for example to actively ask about certain structures and thereby enlarge its knowledge base (e.g. "What is the name of the object you just put on the table?"). In exploration situations it can put objects aside or open doors because of the known movable nature of things. Object or category recognition tasks also become easier if a pre-segmentation can be applied.

If several egocentric models of geometric structures with dynamic properties are combined the field of applications for inferring new knowledge becomes even broader. Imagine a robot detecting movable objects and agents by observing changes in the scene from a first egocentric viewpoint. After moving to a subsequent location the robot is able to infer a new egocentric representation of the same scene which already includes information of the movable parts of the scene previously observed. This is because it can access a more precise static background model which may also include knowledge of areas that have not yet been perceived from the current viewpoint, e.g. because of shadows or occlusion by movable objects. So the system can gain much more information about a scene by observing it from different views.

Furthermore, this allows the robot to use the knowledge gathered at one viewpoint also in the more distant future when things at this location may have changed. The system is able to extract objects in a re-visited scene without observing any bottom-up cues in the robot's environment from the current viewpoint. Even objects that have never been seen before can be detected, as long as their former absence can be proved by previous scene models. Additionally, the system can infer additional information in the past by back projecting the current scene model to previous viewpoints. This allows to retrospectively detect objects that were previously assumed to be static background. This may be useful for long-term analysis of action and behavior patterns of agents in their home environment or for searching and cataloging tasks.

3.2.2. The Challenge in Segmenting a Scene

Segmentation of 3D scenes is usually solved by highly task-specific approaches using *point clouds* or depth images. Figure 3.3 gives an overview of a typical processing pipeline allowing different pathways for segmenting a 3D scene. The visualization is by far not complete but aims to pin down where the focus of this work and related approaches lies.

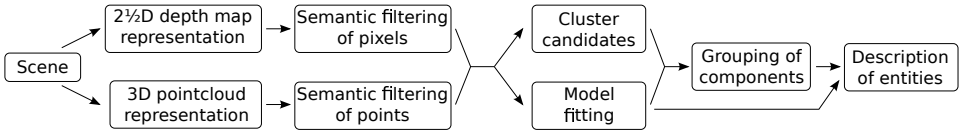


Figure 3.3.: A pipeline of typical steps in the process of segmenting a 3D scene. Most approaches either choose a depth map or pointcloud representation for their algorithms. A semantic filtering step applies world knowledge to mask the relevant parts of the scene. These are clustered or fitted to a model and optionally grouped to describe the entity.

A large field of application for this topic is mobile manipulation. So, many publications for segmenting tabletop scenarios can be found that generate simplified representations of the segmented objects in order to support grasping strategies. Fitting geometric primitives to point clusters on top of a supporting plane is a widespread strategy (Rusu et al., 2009c; Marton et al., 2010; Goron et al., 2012). Other approaches renounce the use of predefined templates, but instead incorporate self-motion in order to separate individual objects visually over time in an fixation process while moving a camera around objects (Björkman and Kragic, 2010; Nalpantidis et al., 2012) or manipulate object hypotheses in order to verify connected parts (Kuzmic and Ude, 2010).

Recent model-free approaches consider multiple cues from basic surface properties in order to generate sensible object hypotheses. Uckermann et al. (2013) focus on the grouping aspect and combine identification of smooth object surfaces and the composition of these surfaces to unified objects by using a probabilistic similarity graph considering adjacency, curvature and co-planarity. This enables them to reliably segment highly cluttered tabletop scenarios of stacked objects.

Another graph-based segmentation that is not limited to tabletop scenarios and also includes color cues is described in Strom et al. (2010). When moving from close-range tabletop scenarios to mid-range vista space scenarios the requirements and challenges change significantly. Single objects are usually not as nicely spatially separated and their appearance is not as compact as it is the case for many on-table objects. These are usually quite convex while larger objects like furniture often appear in a concave shape. Further, the background of vista space scenarios is much more complex and the scenes contain more variety in the level of detail that must be applied in the segmentation.

Nan et al. (2012) try to deal with these problems by fitting templates of three-dimensional structures into the scene. Their goal is to detect known primitives using model fitting and to gain directly an entity description. In contrast to the fitting approaches we have seen in tabletop scenarios, these templates are much more complex. They use a set of specific models of objects that may be found in indoor scenarios – such as chairs, tables and other furniture – instead of geometric primitives, which has the disadvantage that this approach is limited to segmenting the fixed set of predefined objects. Another example of very goal directed handling of the segmentation problem is presented by Rusu et al. (2009a). They generate a dense semantic 3D object map by systematically decomposing the complete room structure into meaningful parts while making extensive use of world knowledge about typical indoor environments. So this approach handles basically all aspects of the processing pipeline in a very manually crafted way. The presented implementation is highly optimized for the kitchen environments described in the paper, but does not scale to different scenarios because other room types require different world knowledge and also vary much more than kitchens usually do.

Apart from these geometric approaches utilizing *point clouds* for segmentation, there is another type of algorithms that primarily rely on the depth image. This is a two-dimensional map which codes the measured depth for each pixel of the sensor. Silberman et al. (2014) try to realize a pixel-wise full image labeling of semantic regions based on color and depth information. They use hierarchical segmentation trees in combination with convolutional network features for generating an optimal set of classified *superpixels* for grouping the scene into meaningful regions.

The latter approach is the most similar one compared to the segmentation

strategy I will describe in the following, although the methodology differs strongly. The presented strategy focuses on labeling scene parts with dynamic properties rather than on distinguishing between individual uniform instances. Assembling continuous structures or building entities out of this information is not within scope of this work. But a combination with an approach which exploits geometric features for identification of reasonable entities like presented in Uckermann et al. (2013) would be imaginable.

The goal of this chapter is to generate an egocentric scene model that supports the interaction between an domestic service robot and the human. The basic assumption for this is that the robot can learn many relevant structures of the environment without prior world knowledge just by observation of the interlocutor's (or collaborative) actions and descriptions which are linked to changes of the environment. Hence, a full segmentation of all visible entities as seen in the tabletop approaches — especially by fitting known templates — or even a pixel-wise full scene labeling as seen in Silberman's work is not required here. Whereas the depth image based strategy seems suitable for an egocentric model dealing with scene changes. The segmentation approach I will be using for realization of the egocentric scene model is purely based on observation of change of the background and foreground structures.

3.2.3. The Articulated Scene Model

A first approach to segmenting the currently observable scene into functional parts was introduced by Swadzba et al. (2010). Their *Articulated Scene Model* (*ASM*) is the basis for the further considerations that lead to the results described in this thesis. The model is motivated by the definition of *motion* and *change* proposed by Rensink (2002). *Motion* is defined as variation referenced to location and *change* as variation referenced to structure. This has consequences on the perceptual processes involved. For motion only local derivatives are needed so that motion detectors can be located at the initial stages of visual processing where spatial representations have minimal complexity. In contrast, change is referenced to a particular structure that must maintain spatio-temporal continuity and needs therefore more sophisticated processing. In the model, the assumption of a separated processing of change and motion is realized by two layers, one responsible for handling the articulated scene parts and one for handling moving entities.

Concretely, the model focuses on detection of completed changes which involves a comparison of currently visible structures with a representation in memory. Hence, it detects movable parts and adapts the static background model of the scene simultaneously. The detection of dynamic object parts (like moving humans) is modeled in a separate layer and requires a tracking mechanism. In the original implementation by Swadzba et al. (2010) this was realized with particle filters using clusters from optical flow. The refined implementation used in the research for this thesis utilizes the body tracking algorithms in the NiTE Middleware in combination with the OpenNI SDK[1]. It should also be said, though, that the algorithm only labels the pixels of a 2.5D representation of the scene with the respective dynamic properties, it does not track or distinguish instance-level entities. This needs to be done in a post-processing step.

The algorithm introduced by Swadzba et al. (2010) (see Algorithm 1) works on depth images and assumes that the view direction remains still while the processing of the model is active. At each time step t the algorithm receives the depth image of the current frame

$$\mathrm{F}_t = \left\{ f_t^i \right\}_{i=1\ldots n}, \quad f \in \mathbb{R}_{\geq 0},\ t > 0 \tag{3.1}$$

(where f_t^i is a depth measurement for an individual pixel) and the current dynamic regions $\mathrm{D}_t \subset \mathrm{F}_t$ provided by the tracking module. In the default version of the algorithm the dynamic regions are excluded from the current frame so that $\mathrm{F'}_t = \mathrm{F}_t \backslash \mathrm{D}_t$. Now the algorithm compares the input frame $\mathrm{F'}_t$ with the currently known static background S_{t-1}, where

$$\mathrm{S}_t = \left\{ s_t^i \right\}_{i=1\ldots n}, \quad s \in \mathbb{R}_{\geq 0},\ t \geq 0,\ s_{t=0}^i = 0 \ \forall\ i \tag{3.2}$$

The algorithm does a pixel-wise comparison in order to determine the dynamic property of the structure at this location. For each pixel $f_t'^i$ a decision is made whether it

[1] OpenNI and NiTE are software components by the PrimeSense Company providing APIs for using RGB-D sensors like the ASUS Xtion Pro Live. After the acquisition of PrimeSense by Apple, it was announced that they would no longer support the software. However, Occipital and other former partners of PrimeSense are still keeping a forked version of OpenNI 2 (OpenNI version 2) active as an open source software. `http://structure.io/openni` (visited: March 1, 2015)

- supports the background model S_{t-1}
- defines a new background observation $s_t \neq s_{t-1}$
- or represents a movable part $o_t \in O_t$ of the scene

These decisions incorporate an adaptive noise model which defines the noise level θ_d^i for a pixel linearly to its depth value

$$\theta(d) = \beta d, \quad \beta, d \in \mathbb{R}_{\geq 0} \tag{3.3}$$

where β is the sensor specific noise factor (e.g. for the ASUS Xtion Pro we choose $\beta = 0.03$ which gives an expected noise range of 3cm at 1m distance). Additionally, for every pixel a weight value w^i is saved which represents the reliability of the background model. In detail, the decisions are made as follows (also see Algorithm 1):

- If the distance of the observed depth $f_t'^i$ to the corresponding background depth s_{t-1}^i is in the range of $\theta(s_{t-1}^i)$ then it is accumulated to an updated static background point s_t^i with improved reliability (line 4).
- Otherwise, if the input $f_t'^i$ is farther than the known background s_{t-1}^i, the background point is reset to the new measurement and the reliability is reset to 1 (line 8).
- In the opposite case, when the input is nearer to the camera (distance is smaller) than the known background, it is assumed to be part of a foreground object and therefore added to the set of pixels representing movable objects (line 11).

In contrast to the original implementation, in our case the static scene model is not directly fed back into the tracking module providing D_t, because the proprietary tracking module used here works completely independently (see above). However, a slight alteration of the original algorithm allows a combination of both detection mechanisms that relies more strongly on the movable parts detection instead of the black box tracking module. Instead of subtracting the dynamic parts from the input, it may also be subtracted from the detected movable parts O_t. In this case, the pre-processed input

Algorithm 1 Background adaptation and detecting movable objects

Require: $\mathrm{F}_t = \{f_t^i\}$ (current frame)
Require: $\mathrm{D}_t \subset \mathrm{F}_t$ (current dynamic clusters)
Require: $\mathrm{S}_{t-1} = \{s_{t-1}^i\}$ (current background model)
Require: $\mathrm{W} = \{w^i\}\ , \quad w^i = 1 \quad \forall i$ (reliability model)

$\mathrm{O}_t \leftarrow \varnothing$
for $i = 1$ to n **do**
  if $|s_{t-1}^i - f_t^i| < \theta_d^i$ **then**
    $s_t^i \leftarrow s_{t-1}^i + \frac{1}{w^i}(f_t^i - s_{t-1}^i)$
    $w^i \leftarrow w^i + 1$
  else
    if $f_t^i > s_{t-1}^i$ **then**
      $s_t^i \leftarrow f_t^i$
      $w^i \leftarrow 1$
    else
      $s_t^i \leftarrow s_{t-1}^i$
      $\mathrm{O}_t \leftarrow \mathrm{O}_t \cup f_t^i$
    end if
  end if
end for
$\mathrm{O}'_t \leftarrow \mathrm{O}_t \setminus \mathrm{D}_t$
$\mathrm{D}'_t \leftarrow \mathrm{O}_t \cap \mathrm{D}_t$
return $\mathrm{S}_t = \{s_t^i\}$ (new background)
return O'_t (movable parts)
return D'_t (dynamic parts)

F'_t would not be necessary anymore, instead F_t would be used as input for the algorithm. As a consequence this gives the new movable parts model

$$\mathrm{O}'_t \leftarrow \mathrm{O}_t \setminus \mathrm{D}_t \tag{3.4}$$

and the new dynamic parts model

$$\mathrm{D}'_t \leftarrow \mathrm{O}_t \cap \mathrm{D}_t \tag{3.5}$$

This altered algorithm confides more strongly in the movable object detection, because only non-static parts of the scene can be dynamic objects. This should eliminate mis-detections of the person tracking module, given the falsely detected structure is assumed to be static by the *ASM*.

3.3. Anchoring and Integrating Egocentric Models

Now, egocentric models do not exploit their full potential if they do not persist for later reference. But most systems in recent publications build up egocentric representations only for short term use, for generating symbolic entities that are transferred to the allocentric representation so that the egocentric model can be dismissed (e.g. Zender et al., 2008; Vasudevan et al., 2007; Chitta et al., 2012; Nieuwenhuisen et al., 2013). This means that information gathered egocentrically at one location might either not be accessible when the same location is re-visited at a later occasion, or a potentially large number of complex information must be maintained in a global representation. Both options do not seem convenient since reuse of previous information is essential for many tasks and the maintenance of a global environment representation incorporating the appropriate level of detail is impractical. A more detailed analysis of the functional requirements of an enclosing spatial representation has been discussed in Section 2.1.

Especially when considering long term operation of service robots in domestic environments detailed persistent knowledge about certain areas is vital. This can be seen in the applications described in this thesis (e.g. the *lost key scenario*), but also in other interaction and learning tasks going beyond applying dynamic properties to segments of the scene. One could argue that the algorithms discovering the relevant visual properties just have to be good enough to generate the same information again when re-approaching a previously seen area. But this is not true for "invisible" information. For example when one of several identical objects was referenced by another agent, this information must be conserved and cannot be regained later. Or when going back to the work of Kuzmic and Ude (2010), the information about connected parts of the scene discovered by manipulation would require a lot of effort in order to regain. Here, a persistent representation is desirable. The *ASM* algorithm relies on observation of changes in the scene. Since the robot cannot observe the whole apartment at once it must be able to detect changes that happened while it was not observing them. For this it needs a detailed representation of the background structures when returning to a location where such a change occurred. Also in order to infer which objects have been added, removed or relocated in a scene, it must be able to compare segments from the movable layer of the *ASM*.

All these operations would require a very complex model if they were

done allocentrically. It would have to have a very detailed structural representation at least at those locations where the changes are expected and it would require a sophisticated temporal management mechanism that is able to track changes in the currently visible parts of the scene, but also infer temporal correlation in (temporally) invisible areas at the same time in the same representation. This leads to a high computational overhead just to maintain the overall allocentric representation of the environment.

3.3.1. A Twofold Spatial Representation

In this thesis I argue for a closer integration between allocentric and egocentric representations, and also for persistent egocentric representations which interact among themselves as well. The system described in the following sections is supposed to take a step in this direction by exploring a way of anchoring multiple short-range, egocentric representations in a global map structure. The representation follows the idea described in the analysis Chapter 2 of using three different types of representations in order to provide a unified way of representing spatial knowledge (see Sections 2.3). The developed distinction suggests to use an instance based and a structural allocentric representation in combination with a structural egocentric representation.

Following these deliberations, I propose a twofold representation inspired by the findings of Mou et al. (2006) that incorporates allocentric information as a map structure which relates objects to landmarks, as well as egocentric subsystems anchored in the global coordinate system.

Those egocentric models are semantically annotated 3D models of the near field of view of the robot. They are anchored through the camera position from where the scene was captured in global coordinates. A proposal for the decision making process triggering this can be found in Section 3.3.4; a schematic visualization of the representation is depicted in Figure 3.4. This solution does not require a sophisticated management strategy for maintaining a complex overall model. Through the topological anchoring of these models it is possible to update local spatial structures through locomotion which allows simple transfer of knowledge. The available information can easily be filtered on a spatial basis, while the spatial relations of the individual egocentric models are accessible at any time which allows the correlation of the represented data. This way, the results from egocentric calculations

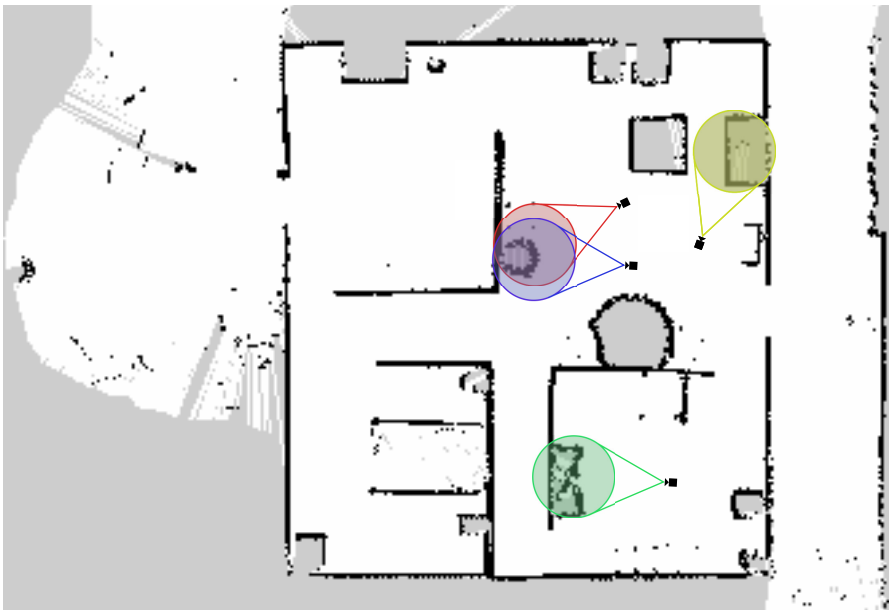


Figure 3.4.: Schematic visualization of the twofold spatial representation. Depicted is the allocentric map containing probabilities for obstacles in the environment and egocentric models anchored in the global representation.

can be preserved and reused when approaching the same location again. But this also allows for extraction of new information from previously visited locations. Either because of newly gathered facts that were not available at the time when the location was originally visited, or just because the information was not needed but has become relevant in the meantime.

The allocentric part of the representation does not only hold anchors for the egocentric models, but also anchors for semantic entities like objects, agents or locations. A two-dimensional map structure holds a rough representation for the spatial layout of the environment which allows to relate locations to landmarks. This serves as the allocentric basis for the system and is realized through a grid-cell based obstacle map from a *SLAM* system. This map can be overlayed with additional regional semantic information, for example for a probabilistic belief about the existence of certain properties in different regions. An application using this feature is described at the end of this chapter in Section 3.4.2.

3.3.2. Registering a Scene Model with the Current View

In order to utilize previously gathered information on a mobile robot, the system must be able to integrate multiple scene models from different viewpoints. In the following I will present a method to reliably fuse multiple scene models from different locations into a new model representing the current view. The method only transfers previously generated background models to the new scene, because non-static parts can be calculated if the background is known. Since the method is designed for a mobile robot, I assume that a position estimate of the camera in global coordinates is available, which drastically reduces the search space for the registration process. The *Iterative Closest Point (ICP)* method is used in combination with the localization information for registration of the corresponding *point clouds*. One of the reasons for registration is to compensate inaccuracies in the localization, which also argues for matching only the static parts to the currently visible scene. Because of the movable or even dynamic nature of the remaining parts it is likely that their location has changed since the last observation. This would make the correct registration of *point clouds* very hard, because registration algorithms usually try to find the best matching of the complete scene. When trying to register two *point clouds* that originate from two very different scene configurations, the probability for making mistakes in the matching is very high. It is much saver to try to match a reduced representation that only contains the static parts of a scene to a full scene with additional movable objects than trying to match two very distinct scenes (see Figure 3.5). Of cause, the higher the confidence that the static labeled structures are in fact static, the better are the chances to correctly register scenes. Newly generated scene models with only few observations involve the risk of containing actually movable structures in the static layer and therefore are endangered to suffer from major scene changes while registering scenes.

Further, especially if the angle on the same scene is particularly different, it is possible that previously discovered fronts of smaller objects are not fully visible anymore and therefore occluded by the object's back. When trying to register those scenes risks exist that the object's back and front are matched onto each other, which would result in an inaccurate model. Chances for this to happen with static parts of the scene is significantly lower, because usually static scene parts are represented by larger structures like walls,

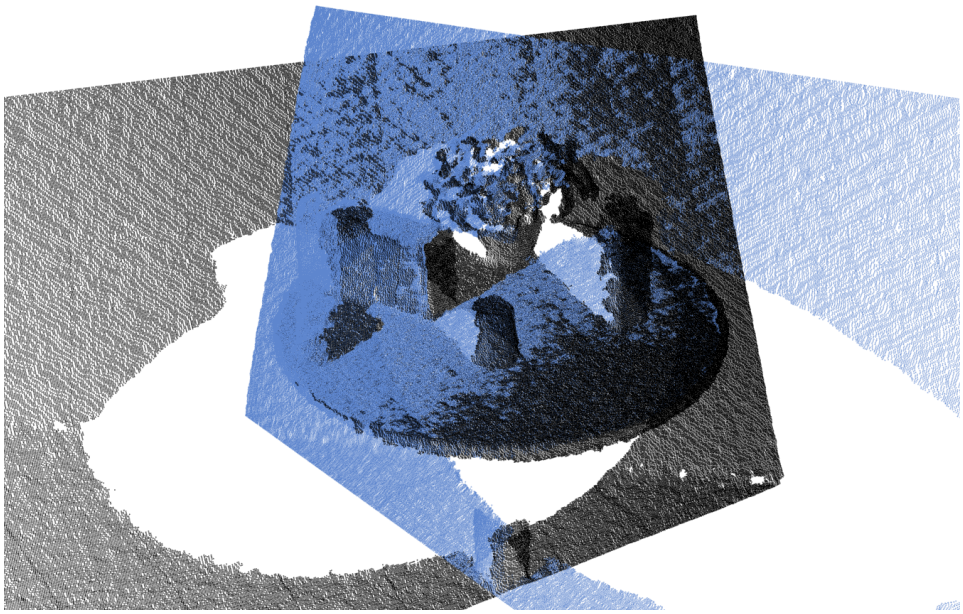


Figure 3.5.: Registering a previous scene model to the current view.

floor or furniture which are rarely seen from front and back. And if so, they are sufficiently far apart from each other for not being mismatched. The distinction of movable and dynamic parts of the scene must therefore be done after merging the background models using the methods explained in Section 3.2.3.

3.3.3. Generating a Valid Model for the Current View

When initializing a scene model at a new location, a reasonable subset of the previously generated static background models S_u^j are transformed into the current position of the camera. The subset can be retrieved through filtering on a spatial and temporal basis. Scene models that represent a completely different location without overlap with the current field of view are skipped (a strategy for limiting the number of scene models for a certain location, so that the number of egocentric models does not outgrow over time, is explained in Section 3.3.4). This way, previous information from other viewpoints is transferred to the current situation. The transformation

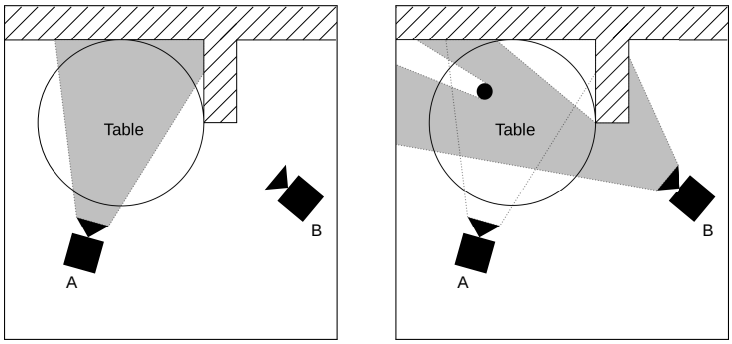
for each merged scene model is calculated from the memorized self-motion from the models' positions to the current one provided by the allocentric representation which is maintained by the navigation component.

For every scene model the static background information is transferred to a *point cloud* representation. The corresponding transformation is applied to the cloud in order to represent it in the coordinate system of the current view. Because of the possible inaccuracy of the position estimate for the current or previous locations, the resulting *point clouds* are aligned to the current view by using the ICP method (Chen and Medioni, 1991; Besl and McKay, 1992). Afterwards, they are again rasterized as a depth image. Thereby the smallest distance is chosen if two or more points fall in the same cell. As a post-processing step a simple closing operation is applied to the resulting depth image in order to close small holes in the surface resulting from uneven distribution of the 3D points on the raster.

Beuter et al. (2011) suggest to use only the ICP registration method to accurately reconstruct the current view from the previous model. However, they assume a subsequent combination of scene models and only small changes in the camera position. This is problematic when the robot travels greater distances. Also this prevents the system from considering knowledge from older scene models. But most importantly, the background model cannot be transformed and used unchanged from a different viewpoint, because at the new location different environmental structures may provide the actual static background. The effects of this are analyzed in the evaluation Section 3.5.1 and are depicted in Figure 3.19. Two examples illustrate the problems with this naive approach (see Figure 3.6):

Example 1 (False movable parts) If the current view contains background structures that were not visible from the previous location, but do now occlude the previously assumed background, the naive method would mark these structures as movable.

Example 2 (False static parts) Also, the new view may contain objects that would have been foreground in the previous location because the background was known. At the current location the object has different background structures which previously may not have been visible. So the naive method would mark this object as background although all necessary information is available to label it correctly.



(a) Previous viewpoint (b) Current viewpoint

Figure 3.6.: Errors in matching a scene using the naive matching strategy.

As part of this work a new merging algorithm for fusing the models with the current camera frame has been developed which is similar to the basic *ASM* algorithm but utilizes the spatial relations of the merged models. The merging algorithm (see Algorithm 2) subsequently merges the transformed static background models S_u^j into the new accumulated background model S_v which initially contains the currently perceived depth camera frame F_t. Note that this algorithm only merges the scene models, but does not distinguish movable or dynamic parts. This is done in a subsequent step using Algorithm 1.

For each pixel the same tests as for the basic *ASM* algorithm are performed. If the incoming transformed pixel $s_u^{j,i}$ is in range of the current accumulated background value s_v^i the model is updated with increased reliability (lines 4). If it is farther than the currently believed background or unknown, a refinement of the accumulated model may be necessary (line 6-10). As described before, a few special cases must be ruled out in order to ensure a correct model. If the transformed pixel does not meet the corresponding premises, it is ignored. The premises are:

Premise 1 (Field of View)

The corresponding 3D point of the pixel s_v^i representing the accumulated background must have been in the field of view at the time when the incoming model S_u^j was generated.

Otherwise it is not safe to assume that the object was moved, it just may have not been visible. This addresses the problem stated in Example 1 on page 46. This premise applies to the blue areas in Figure 3.7.

Premise 2 (Occlusion)

The currently assumed background point s_v^i must not have been occluded by any other point for the original location of the camera of the incoming model S_u^j.

Otherwise the object may again have been invisible because it was hidden. This is a variant of the situation described in Premise 1 and addresses a similar problem. This premise applies to the green areas in Figure 3.7.

Premise 3 (Neighborhood)

The candidate point s_v^i of the accumulated model must not have neighboring points in the incoming transformed model S_u^j.

Because of the transformation of the scene models it is possible that small

Algorithm 2 Merging multiple scene models from different viewpoints

Require: $\mathrm{F}_t = \{f_t^i\}$ (current frame)
Require: $\mathrm{S}_u^j = \{s_u^{j,i}\}$ (transformed background models)
Require: $\mathrm{S}_v = \{s_v^i\}$ (accumulated background model)
Require: $\mathrm{W} = \{w^i\}\,, \quad w^i = 1 \quad \forall i$ (reliability model)
Require: V^j (view frustums of the models)
Require: $P(x, \mathrm{V})$ (projection of a measurement x to the rear boundaries of a view frustum V)

```
1:  S_v ← F_t
2:  for j = 1 to m; i = 1 to n do
3:      if |s_u^{j,i} − s_v^i| < θ_d^i then
4:          w^i ← w^i + 1
5:      else
6:          if s_u^{j,i} > s_v^i and all premises apply for s_v^i then
7:              s_v^i ← s_u^{j,i}
8:              w^i ← 1
9:          else if s_u^{j,i} is unknown and all premises apply for s_v^i then
10:             s_v^i ← P(s_v^i, V^j)
11:             w^i ← 1
12:         end if
13:     end if
14: end for
15: return S_v = {s_v^i} (accumulated background)
```

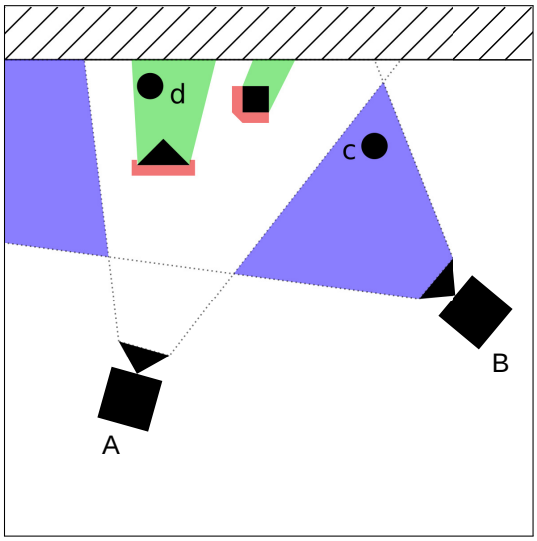


Figure 3.7.: Visualization of the area where the premises apply. Camera position A is the former viewpoint, B is the current view. Notice that the background of objects c and d in viewpoint B are known from the former viewpoint. Object c will not be marked as movable because Premise 1 applies (**blue area**). Object d will not be marked as movable because Premise 2 applies (**green area**). Any measurements in the **red area** will not be marked as movable because of Premise 3.

holes or inaccurate borders of the objects appear when re-rasterizing the image. In order to not propagate these errors into the accumulated model, the neighborhood in the transformed model must be checked. This premise applies to the red areas in Figure 3.7. All premises are calculated using the *point clouds* of the involved scenes and the corresponding camera information including 6D pose and angles of view.

If the premises apply for s_v^i and the candidate $s_u^{j,i}$ is farther than the currently believed background, the accumulated background model set to $s_u^{j,i}$ and the reliability are reset (line 8). If the transformed measurement is nearer than s_v^i it is ignored, because it is already known that the static background is behind this pixel. However, if $s_u^{j,i}$ is unknown (has no measurement) the model is set to the rear projection $P(s_v^i, \mathrm{V}^j)$ of s_v^i on the

view frustum of the camera corresponding to S_u^j (line 10). This accounts for the problem stated in Example 2 on page 46. In this situation it is known that s_v^i is not a static structure because it was not there in the model S_u^j at moment j. The premises ensure this fact. However, the correct static background at this pixel is not known. It is only known that it is not inside the field of view of the camera at moment j (because otherwise there would be a measurement in S_u^j). Accordingly, the accumulated background at this pixel is set to the border of the viewport corresponding to S_u^j in order to enable the subsequent regular *ASM* algorithm to detect this structure as movable (c.f. Figure 3.8).

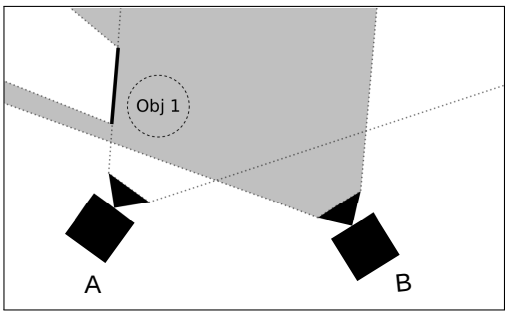
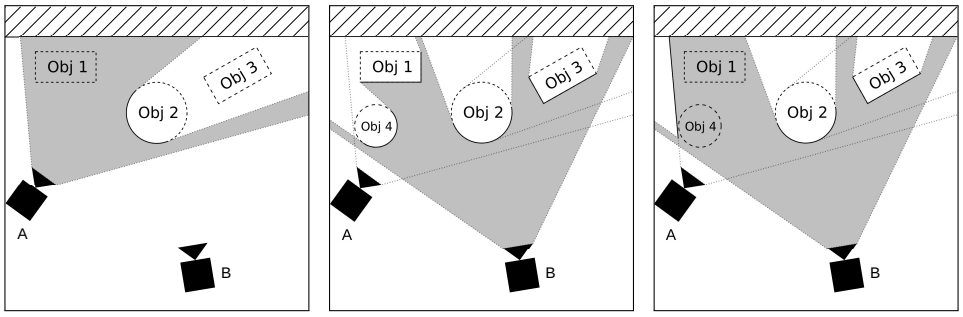


Figure 3.8.: Rear projection to view frustum of former viewpoint. *Object 1* is added in between proceeding from viewpoint A to B and therefore is known to be movable, but the background is unknown. For this reason, the measurements are projected to the rear boundaries of A's view frustum for the static model, because this is the farthest known area which provably does not contain static structures.

Figuratively speaking, the algorithm refines the currently perceived static background using evidence from other viewpoints and thereby fills areas that have not yet been measured, e.g. because of shadows or reflecting surfaces. From the knowledge of the static parts of the scene at different times in the past, the algorithm can implicitly detect if an object was manipulated. Certain parts of the current scene will be marked as movable if one of the merged models provides evidence that the corresponding object was not present at the time the model was built up or it was already known to be movable. The premises prevent that an object is falsely marked as movable because it



(a) Previous scene model (b) Current viewpoint (c) Merged scene

Figure 3.9.: Scene from different viewpoints. *Object 1* is known to be movable, *Object 4* is added to the scene while changing perspective.

was simply not visible from older views, but appeared in a subsequent view. So this algorithm allows the system to gain a much more informative *ASM* without observing any change in the scene from the current viewpoint.

Figure 3.9 summarizes a few of the properties just described. In viewpoint *A* the static background for *object 1* is known (solid black), so it is marked as movable (dashed) (3.9a). In viewpoint *B* before applying the merging algorithm everything is assumed to be background (solid black) (3.9b). This includes the new *object 4* which was added in between observing the scene from the two viewpoints. After merging (3.9c) *object 1* can be marked as movable because the transformed model provides the required background data and all premises apply. However, although the background is known for *object 2* it cannot be marked as movable because the previous front is known to be static and the back is ignored because it does not fulfill the occlusion premise. *Object 3* is assumed to be background as well because of the premises and the missing background. *Object 4* can be marked as movable although the correct background is not known, according to Premise 1 (Field of View). Instead of the correct background structure the border of the viewport belonging to viewpoint *A* is marked as static.

3.3.4. Applications Exploiting the Model's Potential

The developed scene model is not very complex. It does not feature a sophisticated segmentation or tracking strategy and does not require elab-

orate event history management. Yet, it is possible to implement powerful applications that allow substantial statements about the scene.

Keyframes as Memory References

One of the most powerful tools for making statements about the scene changes is the comparison of snapshots of the scene models at different points in time. In order to detect changes that have a semantic relevance it is important to compare the correct pairs of situations. For example, if the system has the goal to find the car keys which have been last seen by the home owner the day before, it needs to compare memory references from the previous day with more recent data – either from the recent past or from the present through renewed inspection. For this it is crucial to store reasonable snapshots of the scene and scene model as persistent references.

This is realized through the concept of keyframes. A keyframe is described through the raw depth image snapshot at a specific point in time and the corresponding static background model. My suggestion is to keep keyframes at the beginning and end of an observation (here observation means the duration of recording a *ASM* at one specific location). A more sophisticated strategy would be to detect additional keyframes in the observation marking a completed movement sequence in the scene. This could be easily implemented by using the optical flow on the movable objects layer of the *ASM*.

By saving these keyframes the system has a reference to the scene layout at this point in time. It can even apply a more recent version of the scene model to the raw depth image to find movable objects that were not detected at the time the keyframe was generated.

Certainly this strategy needs some kind of forgetting mechanism so that the number of saved keyframes and models does not outgrow over time. For the scene models this can be done quite intuitively. When a new scene model is established from a very similar viewpoint as an older model, this older model can be forgotten, because its information should be merged into the new one. Models from a significantly different view on the same scene, however, should be maintained in order to facilitate occlusion issues in future analyses. Selecting candidates from the keyframes mentioned above for forgetting is not as intuitive. The goal is to only forget those keyframes that are not relevant for future reference. A naive approach to this problem

would be to apply an age filter, but the better solution would be to somehow extract the semantic relevance of these specific configurations through a high-level component.

Layered Action Models

When continuing the idea of taking snapshots for completed movements in order to reference reasonable states of the scene model, one may also establish a layered action model. This means that every detected keyframe at the end of a movement is interpreted as a completed action. The detection of these keyframes must be done by another high-level component on top of *ASM* (e.g. using optical flow). A completed action triggers the generation of a new action layer for representing the next manipulation of the scene. This new layer takes the keyframe as a basis for creating a new scene model. This way, the new action layer contains only changes that have been made since the last keyframe while the older layers still track the accumulated changes. Together with the persistence strategy from the previous section this can even be done retrospectively.

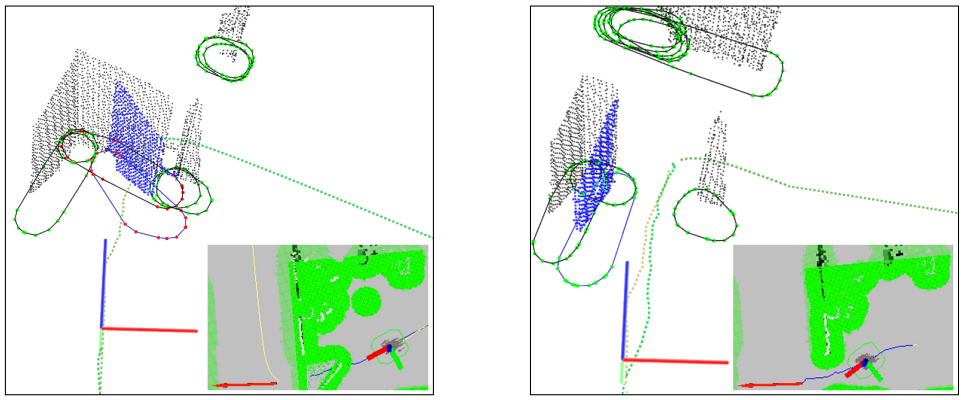
A layered action model like this allows to overcome the shortcomings of the raw *ASM* in terms of segmentation (see Sections 3.2.2, 3.2.3). As described above, the naive *ASM* algorithm cannot distinguish two movable objects that are located closely together. By using this layered action model a distinction of the objects would be possible as long as they were manipulated independently.

Further, the extracted actions can be used to analyze scene changes on a trajectory level. Since these actions should optimally represent the change to only one object, the start and end configuration of the now unique movable object allow a rough approximation of the trajectory without using a classic visual tracking algorithm. Among other applications, this is useful for learning trajectories of objects that have a defined but limited movement space like doors or drawers. This ability is used in the case example described in the section below.

Case Example: Movement Strategies for a Mobile Robot

The navigation system described in Meyer Zu Borgsen et al. (2014) utilizes the *ASM* component developed for this thesis. The goal of the system is

to clear obstacles in navigation tasks through cooperation with a human. For this it uses the functional roles provided by the *ASM* component. In a first phase the robot observes the human manipulating objects in the environment, specifically opening or closing doors. From the observations the robot infers the movement space of the doors.



(a) Movable object detected blocking the path.

(b) Object was moved out of the way.

Figure 3.10.: Example visualizations of the navigation system incorporating *ASM* for movement strategies on a mobile robot. The images illustrate the perceived *point clouds* including the movable parts (blue), the planned path (green), and expected collisions (red). The lower right corners illustrate the robot's position on the occupancy grid.

In a second phase the robot can utilize this information in a navigation scenario in which the planned path is blocked by one of the doors. Through the knowledge of their movable nature the robot triggers a behavior that tries to clear the path (Figure 3.10a). The robot asks a nearby person to open the door. Using the knowledge of the door's trajectory it positions itself in a way that the door can be opened by the human. As soon as the analysis component utilizing *ASM* provides the information that the movable object has moved out of the way the navigational task is continued (Figure 3.10b).

Case Example: Lost Key Scenario

For the evaluation of the work described in this thesis and other research projects a scenario has been developed bringing together a few results from research on *situation model* acquisition. The scenario involves teaching the locations of several objects to the robot, including the homeowner's keys. The goal for the system is to detect additions of new objects to a scene and also the movement of objects from one location to the other while matching verbally assigned labels to the placed objects. *ASM* is used to detect events describing the addition or removal of objects to or from the scene over the observation period which is triggered by the human teacher through a speech command. A controlling application infers that the removal of a labeled object followed by an addition event without re-labeling means a relocation of the same object. This way the location of several objects in the apartment can be tracked over time.

Further, the robot is able to verbalize the currently believed locations of the tracked objects. Since keys are a typical candidate for getting lost in the household, a plausible query for the robot would be "Do you know where I recently put my keys?". This application of the *ASM* contributes to the causality dimension of the theoretic definition of a *situation model*. It exploits the assumed causal link between two observed events to make statements about the movement of objects.

3.4. Focusing the Robot's Attention

In order to realize a robotic system that is able to learn the *ASM* just from observation in *HRI* it is crucial that the robot is located at a position where it can perceive the scene changes when they occur. Hence, it is necessary to design a behavior that incorporates a sophisticated attention and positioning strategy that is able to detect situations in which a relevant event might occur and perform a reasonable repositioning for closer observation. The system described in this thesis contains an attention mechanism that consist of two stages. First, a multi-modal person tracking system is used to keep the robot's interlocutor in visual focus and to evaluate her viewing direction. Further, a visual attentive region mapping system uses the allocentric map representation of the *situation model* to provide information about potential interaction spaces in the environment, namely horizontal

surfaces. The repositioning behavior evaluates the positions of persons and surfaces, as well as the viewing direction in order to detect whether the interlocutor approaches a possible interaction space. In this case it repositions the robot so that it can look over the interlocutor's shoulder.

3.4.1. The Interlocutor's Viewing Direction

For person tracking on the mobile research platform *BIRON* the multi-modal anchoring system by Fritsch et al. (2003) has proven to work well in the past across multiple scenarios and research projects at Bielefeld University. However, the component provides only locations of person hypotheses, not their viewing direction. Within the context of this thesis I have extended the system in order to being able to receive viewing directions of the tracked persons as well.

System overview

The approach of Fritsch et al. (2003) applies perceptual anchoring of symbols from multiple modalities to coherent person hypotheses. Anchoring is defined as the process of linking abstract representations of objects in the world (symbolic level) to physical observations of these objects (sensory level). These links (anchors) are dynamically updated every time a new observation of the object is perceived. The symbolic description of a complex object contains multiple anchors to different types of percepts, originating from different perceptual systems. Different anchors cope with different spatio-temporal properties of the individual percepts, because each anchor defines its own *component anchoring process*. A signature list provides an estimate for the values of the respective observable properties of the object. In the implementation for this thesis torso and leg percepts have been used to compose the complex person symbol. However, in principle the system also supports face, sound source and shirt texture percepts. Each percept is generated from the sensory input by individual detection components.

The additional *composite anchoring process* manages the composition and connection of individual percepts to the symbolic descriptions (see Figure 3.11). This process requires three models describing the complex symbol:

- The **composition model** describes the spatial layout of the individual components within the composite symbol.

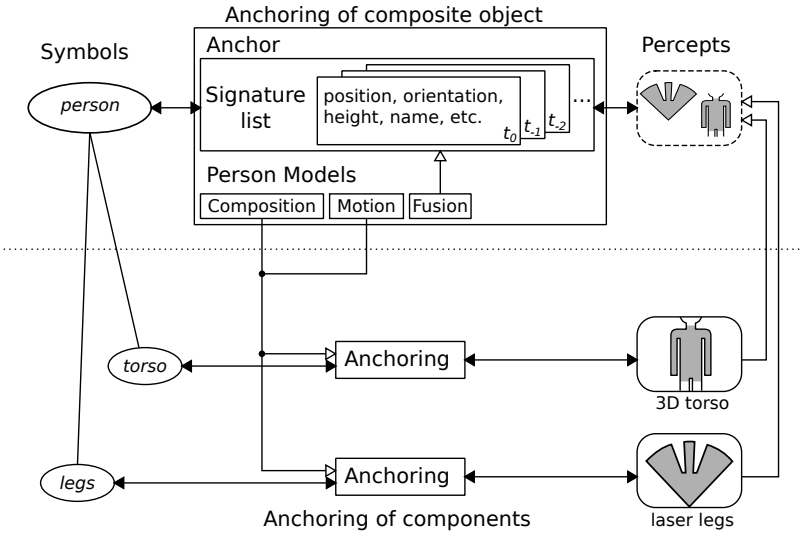


Figure 3.11.: Multi-modal anchoring of persons

- The **motion model** describes how the complex symbol may move. This allows a prediction of the composite symbol's position as well as those of the components by using the spatial relations from the composition model.
- The **fusion model** defines how the percepts from the individual anchors combine to the composite symbol.

Since every percept is assumed to originate from exactly one object in the real world, a *supervising process* is required that controls the selection of percepts by anchors. Instead of statically selecting matching percepts, every component anchoring process assigns scores to all percepts rating the proximity to the predicted state of the corresponding composite object. The supervising process then manages the optimal assignment of percepts to anchors and decides on the establishment of new anchors or removal of not updated anchors.

Percept Detectors

For the system presented in this thesis two types of components are used to compose a person symbol: legs and torso. The detector for pairs of legs uses the radial distance scan from the laser sensors of the robot at leg-height. This gives 720 distance measurements at $\sim$ 35Hz within a complete 360° field of view. The detector groups neighboring distance points to segments. These segments are classified for being legs based on the number of reading points, mean distance, standard deviation, width in world coordinates and distance to the adjacent segments. Single leg hypotheses are ultimately grouped to pairs of legs (see Fritsch et al., 2003).

The torso detector uses the depth image from a ASUS Xtion Pro mounted on top of the robot. The detector can run in two different modes: The *static mode* (the robot stands still) and the *moving mode* (the robot is in motion). In the static mode the person detector from the OpenNI NiTE framework can be used (Shotton et al., 2011). Their algorithm performs a dense probabilistic body part labeling by classifying every pixel of the depth image using a decision forest based on specialized depth image features. The resulting body parts are used as approximation for skeletal joints. Their 3D position is refined by back projection to the person's point cloud which can be retrieved from depth image. The output of this algorithm is either a complete or partial body skeleton or just the person's center of mass, depending on the currently observable body parts and the algorithm's performance which is limited by outer disturbances like sun light, reflections, occlusion, etc. However, this algorithm is designed for stationary views on the scene and does not work very well when the sensor is in motion. This is why the moving mode of the torso detector was implemented. In this mode the 3D *point cloud* of the scene is analyzed for clusters of certain size as rough person hypotheses. Their centers of mass are returned as a result.

Extension to the existing System

In order to enable the multi-modal anchoring to track the person's orientation, a few changes have been applied. The composite symbol and it's anchors now host a new signature *orientation* with its corresponding merging instructions in the *fusion model*. The implementation of the torso percept now supports a function to receive an orientation property on condition

that the detector provided a skeleton (as mentioned above, this is not always the case). The orientation is calculated from the waist joint's orientation, whereby we assume at this point that the person is always oriented towards the robot, because the detector is not able to distinguish between a person's front and back. The score calculation scheme of the torso anchor and the *fusion model* consider both the assumed orientation and the mirrored counterpart as matching candidates. This enables the system to track arbitrary orientations – especially when the person turns away from the robot – although only orientations towards the robot can be observed.

Additionally, the system assumes that persons in motion are always oriented towards their moving direction. This can be exploited to refine the tracked orientation through close cooperation between *motion model* and *fusion model*. If the motion model detects a linear velocity above a certain threshold, the direction of the corresponding vector is used by the fusion model to update the orientation. Since this angle is unambiguous in contrast to the torso percept's information, the corresponding fusion scheme allows to switch the previously assumed orientation of the composite symbol to the opposite direction.

3.4.2. Detecting Interaction Spaces for Manipulation

For the observing service robot it is inappropriate and also impractical to closely follow the observed person around the apartment to never miss any important event. Instead, an appropriate strategy must be applied that reduces the search space and enables the robot to approach the human only in reasonable situations. A valuable subset of the locations inside an apartment that are worth being observed for changes in order to segment unique objects and their functional role are horizontal surfaces. It is very likely that those surfaces – typically provided by tables, shelfs or cupboards – support objects that are frequently manipulated by the homeowner. The *Semantic Annotation Mapping (SeAM)* system has been developed for mapping these kinds of task-specific semantically important areas in the robot's environment (see Siepmann et al. (2014) for details and application in a different scenario).

The goal of the *SeAM* system is to enrich the allocentric, purely spatial map representation (Section 3.1) with low-level visual information. The potentially relevant areas are detected within the robot's visual field of view

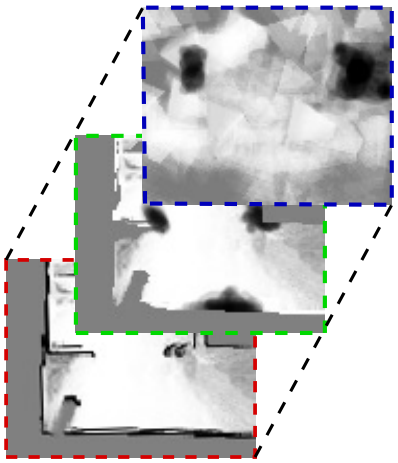


Figure 3.12.: Grid map layers of the *SeAM* system.

by using simple and computationally efficient visual features. I refer to this kind of data as peripheral visual cues. In this case only horizontal surfaces serve as input for the system. Those are detected through evaluation of *point clouds* from the ASUS Xtion Pro depth sensor using the *Point Cloud Library (PCL)* (Rusu and Cousins, 2011). The *Random Sample Consensus (RANSAC)* method (Fischler and Bolles, 1981) is used to fit plane models into the 3D data considering the surface normals. The post-processing involves clustering of the resulting points to extract planar patches and calculation of concave hulls for a geometrical description of the patches as polygons. Other peripheral visual cues are described in Siepmann et al. (2014).

The data representation in *SeAM* is a hierarchically layered grid map structure, based on the *SLAM* obstacle map which contains only physical obstacles that can be detected by the laser range finder, such as walls and furniture. Additional grid map layers containing a spatial representation of the low-level visual cues are stacked on top of the base map (see Figure 3.12). Each of these semantic information layers cover the same space as the base map in the real world. Similarly to the obstacle map, the cells of the peripheral information layers represent a probability value for containing the corresponding visual cue. This way, the information is directly coupled with the spatial representation of the world surrounding the robot. Data

from different layers can directly be combined, not only visual cues but also their relation to the outline of the physical environment.

The actual mapping of the visual cues is done by raising or lowering the cell's probability values of the corresponding layer in the *SeAM* map. If a cell is covered by the area containing the visual stimulus, its probability value is raised. For cells that are in the robot's field of view but are not activated by the visual stimuli, the probability values are lowered. This encoding is similar to the representation of the *SLAM* obstacle map.

Because of the layer structure of the grid maps representing the same spatial area, information from multiple layers can be fused to generate more sophisticated data. When registering visual cues in the grid map that do not have a three-dimensional representation, the algorithm assumes that the corresponding source of the stimulus is not behind a wall or another tall obstacle. So, cells that correspond to obstacle cells in the *SLAM* layer, or are positioned behind those cells in respect to the robot's viewpoint, are not altered and remain unchanged. Further, it is possible to introduce additional grid map layers that fuse information from different sources by applying logical operators on the detection results. Semantically these maps could represent areas where one of the visual stimuli was detected exclusively, or could map only areas which contain several specific stimuli at once, etc.

The grid map layer representing the horizontal surfaces is used in the system described above to detect valuable observation targets in the environment. By applying a threshold filter and a region growing mechanism, a representation in global coordinates can be compiled that allows a comparison with the tracked person's viewing direction.

3.4.3. Repositioning for Observation

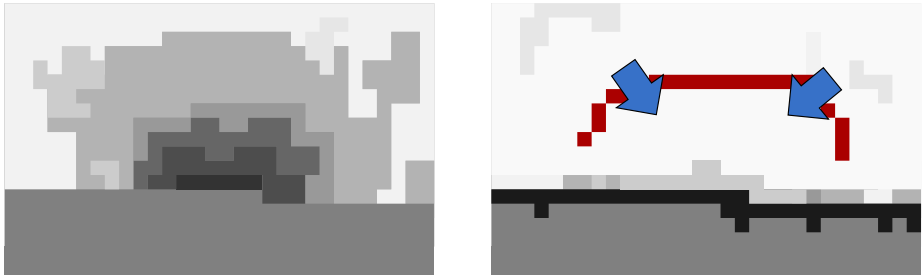
The implementation of the repositioning behavior for the robot was realized using the system abstraction framework *BonSAI* (Lohse et al., 2013). It makes use of the person's locations, their viewing directions and the horizontal surface information retrieved from the allocentric representation in the *situation model*. The system tries to keep the present persons in focus by tracking and turning towards them. A monitoring component analyzes changes to the *situation model*. Whenever a person is standing next to a surface and the viewing direction points towards this surface, the repositioning is triggered. The assumption behind this is that it is quite likely

that the person approaches the surface in order to manipulate something. Observing this manipulation may lead to new insights into the dynamic properties of the structures on top of the observed surface and therefore to an enlargement of the knowledge base.

In order to find a reasonable position for observing the activity, a few requirements must be considered. The observation spot should be near to the surface (not farther than 1 or 2 meters) but should not disturb the person. It should be reachable from the robot's starting position in terms of potential obstacles blocking the path and there should not be any obstacles blocking the view on the surface. Also, the viewing angle onto the scene should be aligned with the person's angle as close as possible in order to minimize the chance of some other structures on the surface occluding the manipulated object.

The implemented strategy begins with extracting possible viewpoints around the surface similar to the strategy described in Siepmann et al. (2014). The probability map from the horizontal surface layer of the *SeAM* representation is received via a *sensor* interface (Figure 3.13a). The system binarizes the map and applies a dilation operation with a structuring element of roughly one meter radius. A Sobel filter operation reduces the regions to only their boundary cells B which are assumed to have a reasonable distance to the actual surface. Since not all of the real world locations corresponding to the boundary cells are appropriate navigation goals, the obstacle map is used to delete all cells which are not reachable, do not have a minimum distance to obstacles, or are located in an unknown area. The remaining cells are clustered using the k-means algorithm with $k \sim |B|$. The centroids of the resulting clusters are treated as viewpoints (Figure 3.13b).

For every viewpoint candidate a rating is calculated. The rating involves the distance to travel and the position of the viewpoint relative to the person. A position right next to the person is rated best, while positions in the person's back are rated worst. Finally, the algorithm confirms whether a viewpoint is reachable by consulting the navigation module and then navigates to the best rated reachable target for observing.



(a) Probability map from horizontal surface layer

(b) Remaining boundary cells with viewpoints

Figure 3.13.: Conceptual visualization describing the calculation of viewpoints.

3.5. Evaluation

The evaluation of the systems described in this chapter is split into three parts. First, a quantitative evaluation of the merging methods for *ASM* processing will be performed. Here I will test on a pixel-basis how precise the merging works in different situations, including corner cases. The ability to detect change events combined in a high-level interpretation component using the *ASM* system will be tested in a qualitative evaluation. For this a teaching scenario has been conducted in a real life environment. The pro-active robot behavior for observing relevant scene changes described in Section 3.4 is tested in a qualitative case study as well. Here the robot's ability to focus its attention to relevant changes made to the environment by a human is examined.

3.5.1. Quantitative Evaluation

For the evaluation of the performance of the merging algorithm described in Section 3.3 a large set of constructed cases in multiple settings was defined. A quantitative performance analysis using a measure for pixel-wise accuracy tests will be done and used for comparison with another naive matching approach. Notice that a quantitative evaluation of the original *ASM* algorithm is described in Swadzba (2011).

Goals

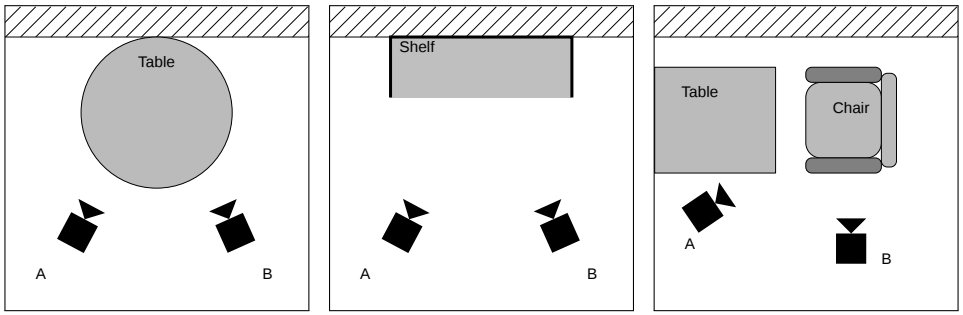
The goal of the evaluation is to analyze the accuracy of the simple *ASM* approach and the merging algorithm in various situations. Its performance will be compared to the naive merging strategy described in the beginning of Section 3.3.3 — with and without registering of the 3D scenes. First, the original *ASM* algorithm will be analyzed on a per-pixel basis in order to get a baseline for the expected accuracy and a signal to noise ratio. Then, the naive merging approach will be deployed to a small set tests that are known to be problematic in order to show the limitations of this approach. Finally, the accuracy of the developed merging algorithm will be analyzed and compared with the previous tests. This allows to make statements about the quality of the merging results.

Method

In order to regard a wide range of situations that might occur in a real-world scenario a set of settings for the tests have been defined. The settings differ in the types of furniture that support the manipulated objects. The chosen settings were designed to cover the most common places in a realistic home environment where relevant object manipulations typically take place. For every setting two camera positions were defined that were used for recording the scenes. Most of the actual test cases were carried out in three of the defined settings. These are a tabletop situation (TA), a shelf with multiple boards being used (SE) and an armchair with a small table representing a living room situation (CA). See Figure 3.14 for a visualization of these settings.

An additional set of another four settings (DO, SO, KI1, KI2) has been designed to test a small subset of test cases in situations that do not conform to the classic object manipulation scheme tested in the other settings (see Figure 3.15). However, these settings highlight a few very interesting fields of application for the *ASM* that go beyond the detection of addition or removal of small objects.

Settings DO and KI2 target the manipulation of entities that have a limited range in which they can be moved. But still doors and drawers are movable objects and their function relies heavily on this fact. Since their range for manipulation is limited an analysis of the possible trajectories is



(a) Setting TA (b) Setting SE (c) Setting CA

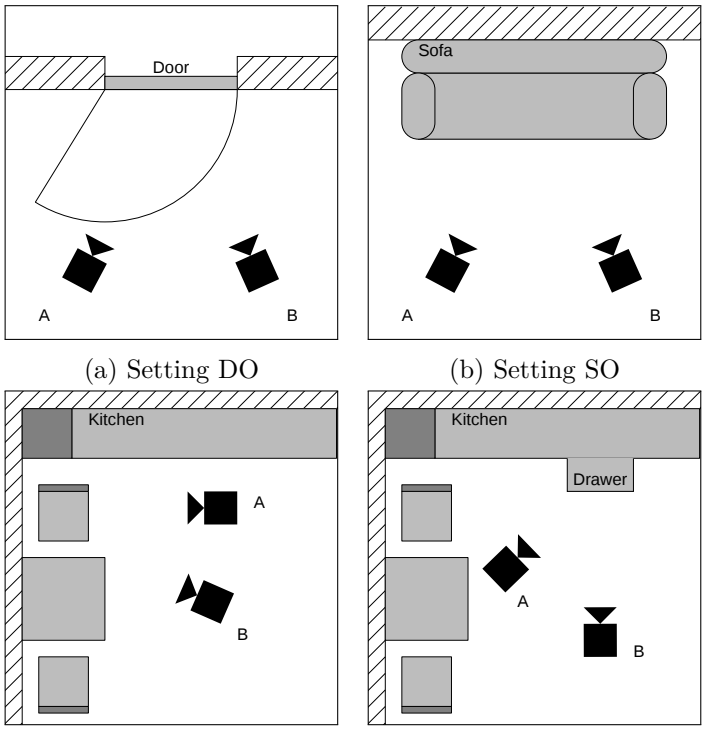
Figure 3.14.: Schematic visualization of settings TA, SE and CA

an interesting information to be exploited in various scenarios. As seen in Section 3.3.4, the system has already been deployed for this reason in a navigation scenario that contained a door as an obstacle (Meyer Zu Borgsen et al., 2014).

Settings SO and KI1 are designed to show that the addition/removal detection events generated by the system are not limited to table-top scenarios. They can be applied valuably in situations where larger objects like furniture are displaced (KI1), or possibly dynamic entities are observed in a non-dynamic situation — in this, case a person taking a nap on a couch (SO).

All of the settings described above have not been cleaned from possible distractions, on the contrary, all settings have been deliberately equipped with additional objects that needed to be ignored by the system. The test cases that have been used for the evaluation were defined in advance to the execution of the tests and are similar for all settings. The definitions contain instructions for how the objects in the scene need to be repositioned and which manipulation should be performed in which viewpoint.

However, the specific objects used for the tests and their positions in the scenes varied across the settings. As mentioned above, not all cases have been tested in every setting. The test cases were designed in order to enable an analysis of the performance of the simple *ASM* algorithm, the naive merging of the multiple models and the performance of the refined merging algorithm. A complete overview of the test cases used can be found in Appendix A (page 205).

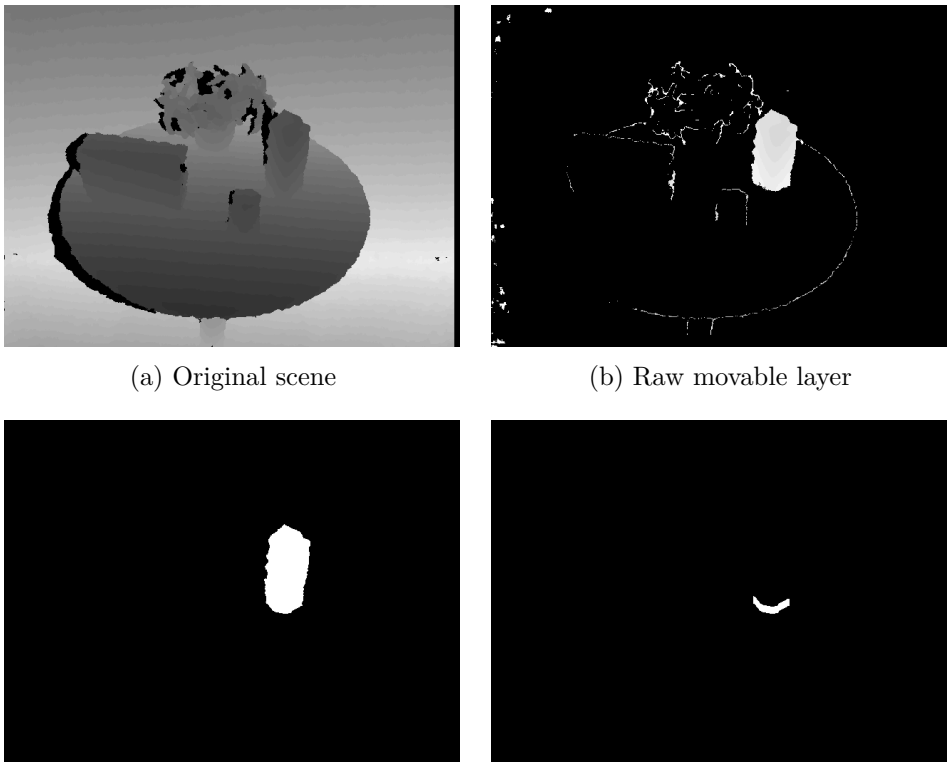


(a) Setting DO (b) Setting SO

(c) Setting KI1 (d) Setting KI2

Figure 3.15.: Schematic visualization of settings DO, SO, KI1 and KI2

For all test cases snapshots of the raw depth image and the various layers of the scene model have been taken at the end of the performed test (Figure 3.16a, 3.16b). The depth images were used to label the movable parts of the scene as ground truth (Figure 3.16c). The resulting movable layer masks from the algorithm contain a significant amount of noise, due to the camera's inaccurate measurements. A simple heuristic for removing those noise pixels from the masks has been implemented. This is a reasonable means for enhancing the results for analysis, because any application defined on the results of the *ASM* algorithm must task similar steps for generating usable data. The noise canceling implementation removes all pixels that have not at least 25% movable neighbors in a 15×15 neighborhood. Since the transition from the supporting structure to the actual object, or from one object



(a) Original scene (b) Raw movable layer

(c) Ground truth (d) Transition zone

Figure 3.16.: Required information for quantitative analysis. The depicted example is test case A0 in setting TA.

to the other is usually blurred in the depth image, an additional "transition zone" was defined (Figure 3.16d). Another justification for this zone is the fact that due to the sensor's noise the algorithm applies a threshold for relabeling static pixels to being movable, which leads to systematic errors at the transition between static and movable structures. Now the results from the algorithms can be compared with the ground truth by counting pixels. Each pixel is classified as being part of one of the following classes:

	inside GT	**outside GT**	**inside TZ**
movable	TP	FP	*ignore*
static	FN	TN	

Table 3.1.: Categories for pixel-wise evaluation. The rows distinguish whether the algorithm marks the pixel as movable or not. The columns represent whether the pixel falls into the labeled ground truth area or the transition zone.

Procedure

The robot platform *BIRON II* has been used for the execution of the tests. While following the instructions from the respective test cases, the robot recorded the depth image stream from the depth sensor mounted on its top. Additionally, the positions of the robot's base provided by the *SLAM* module has been recorded at viewpoints that have been taken during the run-through of the tests. In a subsequent step the recorded image streams with the respective robot positions have been applied to the raw *ASM* and merging algorithms.

Analysis of Original Algorithm

The baseline evaluation of the pure *ASM* algorithm gives an impression of how accurate the algorithm works without artifacts from the merging process. It also provides a measure for the signal-to-noise ratio which originates from the measurement inaccuracy of the sensor used. For test cases A0 (see Appendix A, p. 205) the raw algorithm output gives an average $\boldsymbol{F_1}$ **score of 0.808** (recall: 0.896, precision: 0.735). Figure 3.17a depicts an example of a scene with color-coded results.

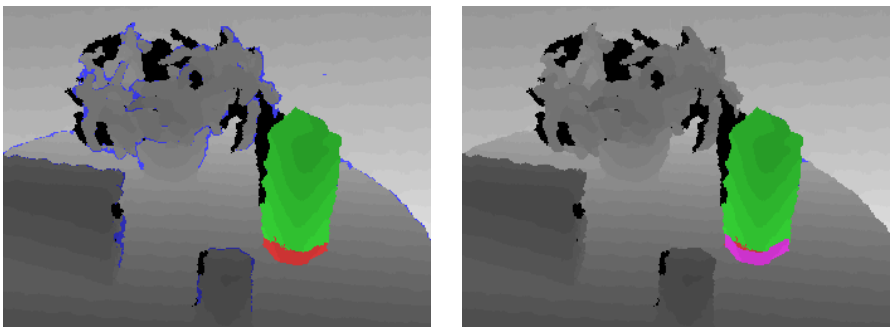
When the noise canceling heuristic is applied, the precision raises to 0.975 while the recall values stays exactly the same. This indicates that the heuristic works reliably, because it deletes most of the noise (precision is close to 1.0) but does not delete any relevant signal (recall value stays the same). Assuming the heuristic models the noise correctly, the **signal-to-noise ratio** can be calculated to **10.3:1**.

When the transition zone is applied additionally to the analysis of the

output, the recall value increases as well and the resulting **F_1 score reaches 0.973** (recall: 0.972, precision: 0.974). See Figure 3.17b for a visualization of the clean results. Since both methods for generating more meaningful results seem to actually enhance the evaluated statements' quality, they will be deployed to the remaining analyses as well.

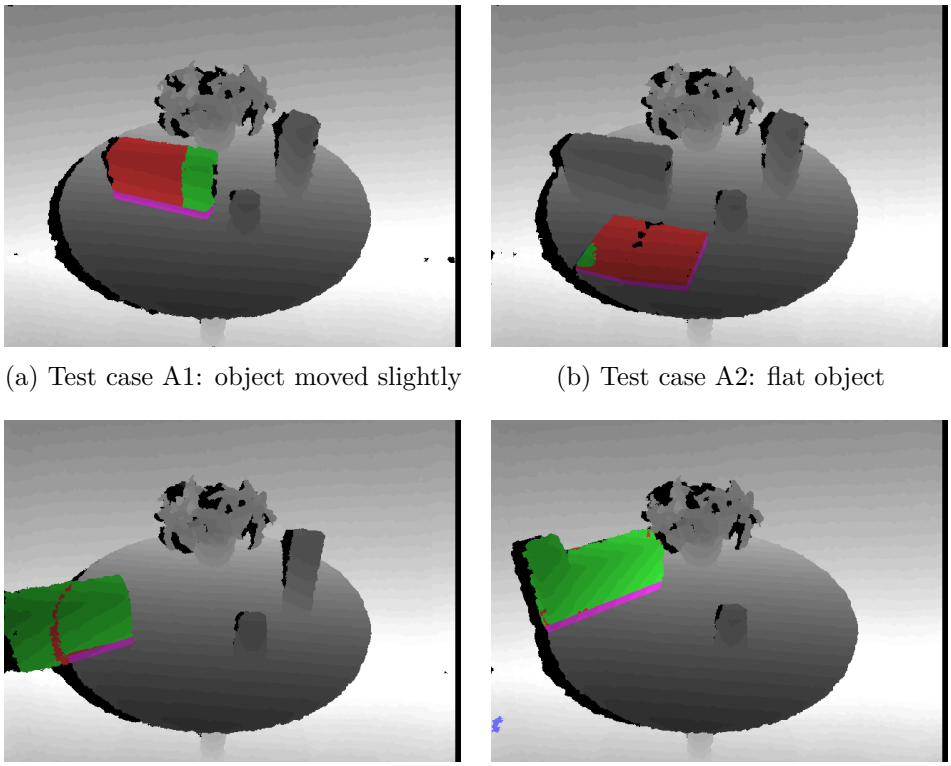
Additionally to test case A0 which represents the default case for the deployment of the *ASM* algorithm, a set of corner cases (A1–A4) has been tested in order to indicate the limitations of the approach. Figure 3.18a depicts a situation in which an elongated object was moved only a few centimeters along its main axis. Since the majority of the object's volume overlaps with the former positions of different parts of the same object, the algorithm can not perceive anything behind the object at these locations. This leads to the visualized situation in which only the part which was moved outside the original volume is marked as movable (F_1 score: 0.576).

The fact that the *ASM* utilizes a noise threshold in order to compensate for the inaccuracies of the measurements from the depth sensor has the effect that thin objects which do not stretch above the noise threshold will not be detected as movable (see Figure 3.18b). This is demonstrated with test case A2 (F_1 score: 0.092). As mentioned in Section 1, the noise threshold is defined through a linear model depending on the depth of the evaluated pixel and must be chosen in a way that planar surfaces do not show noise artifacts



(a) raw results (b) clean results

Figure 3.17.: Close-up on color coded results with and without enhancement methods. Color code: Green: TP – Blue: FP – Red: FN – Magenta: ignored. More results can be found in Appendix B.



(a) Test case A1: object moved slightly

(b) Test case A2: flat object

(c) Test case A3: Object in front unknown background

(d) Test case A4: two close objects

Figure 3.18.: Examples of quantitative results from additional test cases. Color code: Green: TP – Blue: FP – Red: FN – Magenta: ignored. More results can be found in Appendix B.

in the movable layer. However, it should be chosen as low as possible, so that the demonstrated effect of canceling thin objects (and the lower parts of taller objects) has minimal impact.

Another shortcoming of the *ASM* algorithm is visible in Figure 3.18c. If an object is placed in front of an area in which no measurement could be observed so far (in this case because of the projection shadows from the structured-light depth sensor behind the table) it cannot be marked as movable (F_1 score: 0.917).

The last demonstrated problem is not accountable for wrong labeling of pixels but has consequences for interpretation of the data (Figure 3.18d). As already mentioned in Sections 3.2.3 and 3.2.2, the *ASM* algorithm does not distinguish between entities on object-level. The algorithm itself has no means for separating the two close objects in test case A4. A solution for this has already been discussed in Section 3.3.4.

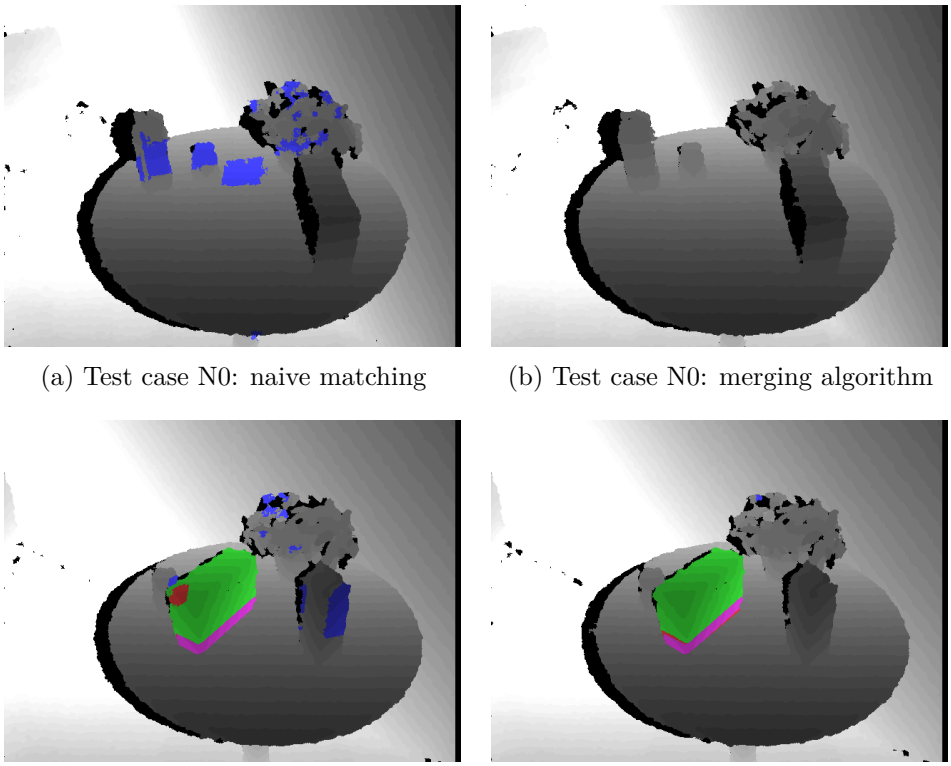
Actually, this solution can contribute to rectifying most of the presented shortcomings. A combination with an approach exploiting geometric features for identification of reasonable entities like presented in Uckermann et al. (2013) would enable object detection and overcome inaccuracies in the labeling process. Especially cases A1, A3, and A4 could be rectified, resulting in correctly segmented objects. However, for case A2 this solution would not yield any positive result. Here, a tracking approach applied to the mechanism for distinguishing movable parts from static ones would be imaginable.

Analysis of Merging Algorithm

For the analysis of the merging algorithm the test cases M0–M9 will be analyzed. But in order to show the relevance of a sophisticated merging scheme when combining two scene models from different viewpoints, the naive approach which just matches the *point clouds* and continues with the default *ASM* algorithm will be evaluated.

From Figure 3.19 one can read that the naive matching generates the same errors as predicted in Section 3.3.3. In test case N0 the naive version marks many areas of the scene as movable although nothing changed in the scenes, neither while observing nor when transitioning from *VP A* to *VP B* (Figure 3.14). This conforms to Example 1 (p. 46). In test case N1 one can observe false negatives on the moved object when using the naive method. These errors occur after transitioning to the new viewpoint although the relevant information from the previous viewpoint is accessible. In both cases the sophisticated merging algorithm makes almost no errors.

Expressed in numbers this means a large difference in the average F_1 score for test case N1. The **naive method** produces a $\boldsymbol{F_1}$ **score of 0.787** while the **merging algorithm reaches 0.965**. Since test case N0 does not contain any change, no true positives or false negatives can be observed. But the "fall-out" or *False Positive Rate (FPR)* can be compared. The results



(a) Test case N0: naive matching

(b) Test case N0: merging algorithm

(c) Test case N1: naive matching

(d) Test case N1: merging algorithm

Figure 3.19.: Examples of quantitative results from comparison of naive matching and merging algorithm. More results can be found in Appendix B.

from the **naive matching** have a ***FPR* of 0.0265** compared to **0.0013 for the merging algorithm**. When also the registration step is skipped for the naive matching scheme, the F_1 score even decreases to 0.264 and the *FPR* rises to 0.0982. All numbers are calculated on the cleaned results.

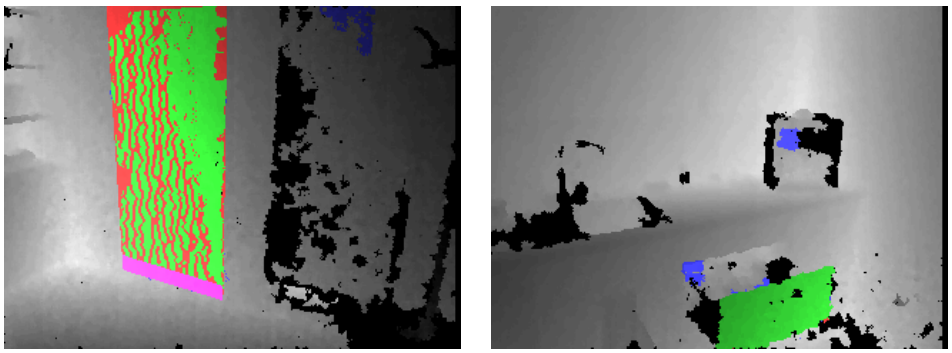
For a more detailed analysis of the performance of the merging algorithm I will first look at settings SE, TA and CA. Here an average $\boldsymbol{F_1}$ **score** for test cases M0–M9 of **0.920** is measured (precision 0.897, recall: 0.945). For comparison, the baseline value was 0.973 (precision 0.972, recall: 0.974) in the simple case without the merging of two models from two viewpoints. It

is striking that the F_1 scores are relatively close but the precision in the merging case is much lower than the baseline value. This difference is also visible from the *FPR* values for those test cases that do not contain any change. The tests with the merging algorithm give a *FPR* of 0.0013 (baseline: 0.0005). This reveals that through the merging process few additional areas are being falsely marked as movable but most actually movable areas are being identified correctly. From qualitative observation while performing the tests I can report that this can mostly be attributed to an inaccurate registration of the *point clouds*. The localization of the robot was not always optimal in the iteration of the tests and the *ICP* registration algorithm was not always able to fully compensate for this location inaccuracy. This is especially true when large objects were added to the scene compared to the static background models that are used as registration counterparts in the merging process.

However, the results show that the goals for the merging algorithm are reached. See Appendix A (p. 205) for descriptions of the test cases.

- Test cases M0, M2, M3, M5 and M6 demonstrate that the knowledge about the static background can successfully be transferred from one viewpoint to the other (F_1: 0.935, precision: 0.926, recall: 0.945).
- Test cases M5–M9 show that changes which were performed in between two observations can be detected (F_1: 0.925, precision: 0.893, recall: 0.959).
- Test cases M1, M4 and M7 show that occlusion is handled correctly (*FPR*: 0.0011).
- Test case M8 demonstrates that movable objects in front of unknown background can be detected (F_1: 0.947, precision: 0.934, recall: 0.960).
- Test case M9 shows that more than one viewpoint change can be handled correctly (F_1: 0.925, precision: 0.965, recall: 0.888).
- All test cases prove that the rear part of movable objects can be detected correctly from a subsequent viewpoint although previously it was not visible.

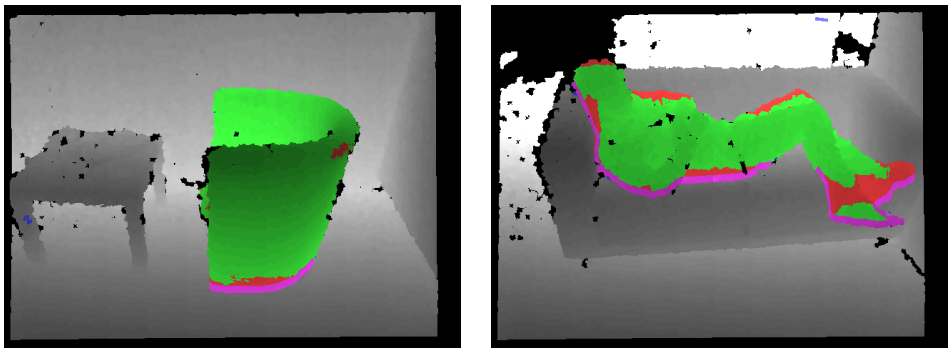
The additional settings DO, SO, KI1 and KI2 demonstrate that the general approach for a multi-viewpoint *ASM* is beneficial not only in table-top scenarios but also in different fields of application (see Figures 3.20, 3.21). It can be used to detect movable parts of the articulated scene that have a limited movement space. This is true for detecting doors (setting DO, F_1 score: 0.780) and drawers (setting KI2, F_1 score: 0.933). Also large objects that are moved quite frequently can be detected (setting KI1, F_1



(a) Setting DO: door was previously open

(b) Setting KI2: drawer was previously closed

Figure 3.20.: Evaluation results from additional settings.



(a) Setting KI1: armchair has been moved previously

(b) Setting SO: previously, the person was not there

Figure 3.21.: Evaluation results from additional settings.

score: 0.986), as well as people who are observed in situations when they are clearly not dynamic (setting SO, F_1 score: 0.911).

Summarizing this quantitative analysis, one can say that the merging algorithm works fine in various — including challenging — situations with a reasonable amount of accuracy. Information about dynamic properties of perceived structures can be transferred successfully from one viewpoint to the other. Also it has become clear that this merging scheme is necessary, since the results from the competing naive strategy indicate the need for a more sophisticated approach.

3.5.2. Qualitative Evaluation: Event Detection with ASM

The proposals from Section 3.3.4 for applications that make use of the *ASM* algorithms for detecting changes in a scene will be tested with the qualitative evaluation described in this section. It employs a human-robot interaction scenario in which certain addition and removal event must be detected. The application being used is a simple implementation of the keyframe analysis strategy proposed earlier. It saves keyframes of the raw depth image stream at the beginning and the end of each observation and applies them to the *ASM* algorithms for the detection of movable parts. By using a standard region growing algorithm the large connected parts within the movable parts layer are clustered and compared across the keyframes. This way, additions and removals of objects of a size above a certain threshold can be detected either meanwhile the observation or in between two observations of the same scene.

Goals

The evaluation aims at analyzing whether the *ASM* system can be used in a relatively simple application in order to reliably detect additions and removals of objects to or from a scene.

Method

In the scenario for this evaluation a human had to show two different locations in an apartment to the robot. In the process the human added and removed objects to and from the scenes. The exact sequence consisted of

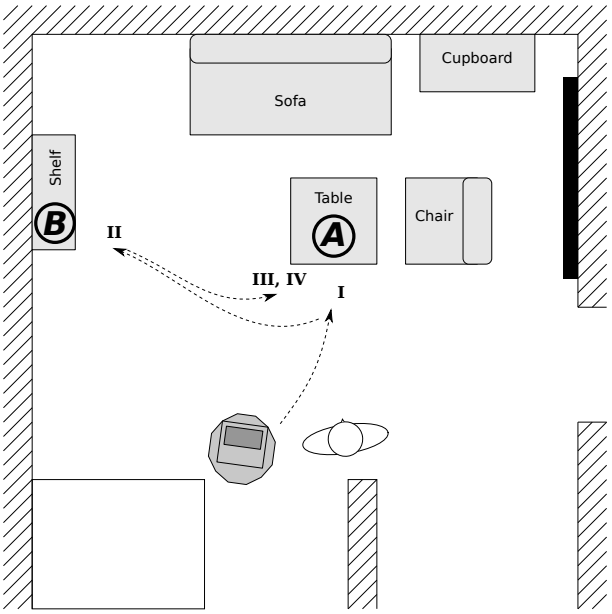


Figure 3.22.: Depiction of the various steps of the scenario for the qualitative evaluation including the observable events.

the steps described below (see Figure 3.22 for a visualization of the various steps):

1. Human brings *object 1* to *location A*
2. Human goes to *location B* and picks up *object 2*
3. (In the meantime the operator places *object 3* at *location A*)
4. Human brings *object 2* to *location A*

For the evaluated system this means that four events can be detected during one iteration of the test:

I. One object is added to *location A*

II. One object is removed from *location B*

III. A new object was added in the meantime to *location A*

IV. One object added to *location A*

In order to enable the robot to perform the test, a robot behavior has been implemented using the high-level abstraction interface *BonSAI*. For approaching the locations *A* and *B* the robot has been programmed to navigate to pre-defined navigation goals. Since the focus of this evaluation lies on the detection of additions and removal events, no real following behavior has been deployed. Through verbal commands the robot could be instructed to proceed to the next location. The viewpoint for the second observation of *location A* has been programmed to be 40 to 50 cm and 30 to 40 degrees displaced from the initial viewpoint onto the scene. The observed events have been announced verbally by the robot and were logged to a text file. These have been used in the analysis for comparison with the true number of observable events. Like in the quantitative analysis (Section 3.5.1) observable and missing events are classified as being true or false:

	expected	**not expected**
detected	TP	FP
not detected	FN	TN

Table 3.2.: Categories for event evaluation. The rows distinguish whether the system detected an event or not. The columns represent whether the event was indeed performed or not.

Procedure

The participant for the tests has been instructed which objects are to be added and removed from the scenes and how to instruct the robot to proceed to the next location. The exact positions where the added objects had to be placed in the scenes were not fixed and varied throughout the tests. The operator placed randomly one or two objects to varying positions at *location A* while the participant showed the removal at *location B* to the robot. Overall, 17 runs of this test have been performed.

Analysis

The system proves that it is able to detect most of the changes to the scene and thereby only rarely mis-detects additional events. Overall, the analysis of the accuracy based on the numbers of correctly and falsely detected events gives a recall value of 0.923 with a precision of 0.882. The errors in the detection come from inaccuracies in the clustering of movable parts. Artifacts from the matching create false positives and correctly detected objects are split into several clusters. The implementation used does a simple region growing on the movable layer's mask. A euclidean clustering on the corresponding *point cloud* which incorporates the real 3D positions of the measurements would probably work better.

It is striking that the detection of changes that are not directly observable makes significantly more errors than the detection while observing. On average, the error rate when detecting meanwhile changes is 23.5% while it is only 4.9% for observed changes. This is not surprising because the detection of changes in the meantime requires to merge two *ASMs* of the same scene but from different points of view. The quantitative evaluation has already revealed that the accuracy of the results when merging two scenes is not as good as when only analyzing the current observation. The inaccuracies in the registration of the *point clouds* and the resulting errors in the merging process result in an even worse performance of the clustering. The detection of changes during the observation phase however just requires to compare two states of the same *ASM*.

Again, when using a more sophisticated euclidean clustering scheme on the *point cloud* there would probably be less false detection events. Especially, when considering that the inaccuracies in the matching were not severe and the resulting artifacts are clearly evident. A smarter heuristic could exclude the obvious merging artifacts and prevent splitting of one object into several clusters. Hence, the observed inaccuracies are not critical for applications using the *ASM* algorithm.

In summary it can be said that the multi-viewpoint *ASM* can be deployed in a scenario for detecting changes in a scene while identifying the moved instances on object-level using a simple extension on top the original *ASM* algorithm.

3.5.3. Qualitative Case Study: Robot Behavior Performance

In order to demonstrate that the system can also be deployed to a more complex robotic behavior than described in the previous evaluation, the behavior described in Section 3.4 for building an attentive robot that constantly learns new facts about the geometric structures in the environment has been realized. The behavior involves a person tracking approach which is used to trigger a sub-behavior that repositions the robot so that it can observe the working space of the tracked person. A simple dialog implementation has been used in order to enable the robot to understand utterances of the person describing his or her actions. The understood labels of the manipulated objects are mapped to the observed addition or removal events. For reasoning about the high-level trajectories which the manipulated objects have taken — in terms of bringing an object from one observation location to the other — a very simple logic has been implemented.

The *lost key scenario* (see Section 1.4) has been performed several times for demonstration of the overall system. The robot has been able to follow the people in the room and often detect when a person stopped in front of a planar surface. If this has been detected, it repositioned itself next to the person facing in the same direction. After visiting a few observation spots, the robot has been able to verbally describe where the objects were last seen and how they were moved from one location to the other. So one can say that the *ASM* method can be used to realize an attentive robot that is able to observe actions and track objects for later reference.

3.6. Summary

In this chapter an approach to building a comprehensive scene model was taken. The focus of considerations was the spatial and functional analysis of the robot's environment. Before proceeding to a semantic view on the situation, I want to sum up the outcomes of the deliberations on spatial analysis described here.

I argue for a twofold spatial representation for encoding the structural environment of a mobile robot. The representation developed contains both an allocentric view on the surrounding which allows to relate objects and landmarks in a global manner, as well as an egocentric perspective on subareas which supports tasks that require to put the self in relation to objects

in the extended field of view. It is shown that there is evidence in literature that a similar approach can be observed in the spatial processing of humans. The allocentric representation is realized with an occupancy grid map approach as known from *SLAM* implementations. The implementation used is a multi-layered extension to this representation, first deployed in the *SeAM* system as a spadework for this thesis. The *ASM* system first described by Swadzba et al. (2010) is used as the egocentric counterpart to the global map and represents dynamic properties of visible structures from a self-centered perspective.

The main contribution described here is a method for maintaining multiple individual egocentric models which allows to transfer knowledge spatially and temporally through interplay with the enclosing allocentric representation. The challenge is here to merge multiple scene models from different views in a semantically correct way while maintaining correct semantic roles despite occlusion and perspective issues. Additionally, a strategy for referencing specific model configurations in the past is described. Together with the chosen scene model for representing dynamics of structures this allows to build powerful application for analyzing scenes only through observation. It is shown that the chosen representation is able to approximate segmentation and tracking results that require sophisticated visual processing in classical systems.

In order to realize a scene model that actually contains the relevant information for future reference, it is crucial to deploy a robotic system that pro-actively observes the important actions. For this I described the implementation of a robotic behavior using the *BonSAI* framework. The behavior utilizes a person tracking system for detecting possibly relevant situations, as well as the *SeAM* framework for mapping locations that presumably serve as a stage for relevant object manipulation actions by the human. Through combination of this information the behavior triggers a repositioning strategy that considers viewing direction of the human and the spatial layout at the target location.

Chapter 4

Applying Semantics - Grounding through Visual Perception

So far the geometric properties of the robot's environment have been investigated. But the segmentation of relevant structures in the surrounding does not suffice to establish a *situation awareness* that is capable of supporting a complex conversation. In referential communication a key requirement is to reliably ground verbal references to objects in the perception. Segmentation alone leaves too much complexity for successful grounding, especially when the reference is not supported by an observable action as in the scenarios discussed previously. Furthermore, apart from the ability to ground utterances the robot needs to be able to reference objects in its vicinity that were not explicitly introduced to it by a human. Accordingly, the system requires a way for applying *a priori* semantic knowledge to objects in its *situation model*. It should be able to categorize objects, furniture and also whole rooms or functional areas within an apartment in order to being able to reference those in an interaction. Moreover, in human cognition the visual system was found to make extensive use of the fact that in real-world situations a strong relationship exists between the environment and the objects within it (Palmer, 1975; Biederman et al., 1982; De Graef et al., 1990). In humans the visual context is processed first in order to index object properties which facilitates the detection and recognition of instances. In the context of visual recognition of objects Torralba (2003) argues that

> "The structure of many real-world scenes is governed by strong configurational rules akin to those that apply to a single ob-

> ject. In some situations, contextual information can provide more relevant information for the recognition of an object than the intrinsic object information."

Further he states that the mentioned contextual priming is also beneficial for object recognition in artificial robotic systems. This, however, requires that the robot is aware of the current situation — in terms of domestic service robots of the type and function of the enclosing room. The algorithms described in this chapter should be seen as a prerequisite for the establishment of a coherent *situation model* and the applications making use of this. The focus of this thesis is the investigation of ways to realize a general *situation awareness* for a mobile service robot. Visual recognition approaches are required, because a large part of the model relies on visually perceived information, but the accuracy is not critical here. Applications making use of the results use probability distributions of the possibly ambiguous outcomes and try to rectify them using other modalities (c.f. Chapter 5).

Related Work

The visual categorization of segmented objects is a fundamental problem in robotics and has been approached in many different ways so far. In recent years, especially classification of 3D data has gained attention in the computer vision community. Huber et al. (2004) present early work on a parts-based object representation for 3D object classification. Their work is based on the idea that specific parts of an object are unique for their category. Three dimensional scans of objects are divided into a fixed number of parts which are grouped into part classes using a hierarchical clustering algorithm and described using spin-images (Johnson and Hebert, 1999). The classification of objects is realized through part-to-object mapping. More sophisticated methods using 3D and 2D features for various recognition tasks are presented in the following.

Furniture Recognition A particularly difficult field of 3D classification is the recognition of furniture because of the high in-class variation. Somanath and Kambhamettu (2011) describe an approach to this problem that makes use of a parts-based model as well. Their approach learns a canonical model for each class using *Gaussian Mixture Models*. The 3D

training samples are represented by aligned spherical functions from which typical parts for each furniture class are derived for generating the models. Random Forrest classifiers are used for recognition.

More powerful features for description of local regions are presented by Rusu et al. (2009b, 2010) and others (e.g. Tombari et al., 2010). The *Fast Point Feature Histograms* are based on surface characteristics derived from surface normals in the neighborhood of a target point. They use them for a global description of objects in order to learn and categorize them using probabilistic graphical methods (Conditional Random Fields).

The use of a vocabulary or codebook of local object parts is a widely used technique for learning a sparse representation of objects based on the work of Agarwal and Roth (2006). For categorization purposes this method is often extended to a *Bag of Words (BoW)* approach in which a histogram over the generated vocabulary is used to describe a category (Csurka et al., 2004). Extensions to this method are described for example in Grauman and Darrell (2005) and Lazebnik et al. (2006). The main disadvantage of the *BoW* approach is that only the distribution of features is considered, not the spatial relations between them. Therefore, this method is mainly used in tasks in which the shape is not particularly crucial, like in the classification of manipulable household objects. However, there are publications that describe approaches for spatially sensitive *BoW* implementations, for example for shape retrieval applications of nonrigid objects using so called "geometric word" (Lazebnik et al., 2006; Bronstein et al., 2011).

Using artificial 3D models from web databases to train a system for recognizing furniture in indoor room scenes has been addressed by Mozos et al. (2011). They use the data to build *Shape Models* of furniture categories by learning the geometric relationship of object parts. The parts are identified by a prior segmentation step and described by a set of geometrical features. To find representatives of typical furniture parts, a clustering is performed to finally get a codebook of object parts. This codebook is then used to build the *Shape Models* of the different categories. For testing they use a probabilistic *Hough Space Voting* to find hypotheses for a location of an object instance of the learned category.

Scene Recognition As stated above, object recognition can benefit from knowledge about the context the target object is situated in. Fisher and

Hanrahan (2010) enhance the shape retrieval by considering context information extracted from 3D scene graphs including object shape, semantic labels, and spatial relations between pairs of objects. Their goal is to predict the strength of a relationship between a candidate model and its existence in the scene to perform context-based queries. Here the context information is not automatically perceived, but other approaches try to visually categorize scenes. One application for this is to distinguish between indoor and outdoor scenes (Serrano et al., 2002). Payne and Singh (2005) present a classification algorithm based on edge detection using a two-stage classification scheme, while Szummer and Picard (1998) use color, texture and frequency information from a *Discrete Fourier Transformation* and a *Discrete Cosine Transform* of the complete image to distinguish between indoor and outdoor scenes.

As in object recognition, the usage of local features for categorization of scenes has been shown to be a powerful strategy as well (e.g. Vogel and Schiele, 2004). Here again, local features are more robust to occlusion and spatial variation than global ones. Fei-Fei and Perona (2005) present an approach for learning natural scene categories by a collection of local regions denoted as codewords. Using *SIFT* features (Lowe, 2004) they show that local features hold a strong descriptive power for describing scenes by using them in a *Bayesian Hierarchical Model* for classification.

Instead of using holistic approaches as described above, other publications present systems that establish intermediate representations using detected objects. This additionally allows a spatial representation of the scene, apart from the currently visually perceived field of view. The advantage of these approaches is that models are easily generalizable to newly perceived scenes. This is an important aspect for indoor scene categorization because of the high in-class variability. This is also discussed by Quattoni and Torralba (2009). They present an approach for labeling scenes based on a combination of objects detected within them and a global description of the scene using *gist* (Oliva and Torralba, 2001). An alternative approach using mixtures of multiscale deformable part models (DPM) to detect objects in the environment for probabilistically inferring the corresponding place type is presented by Viswanathan et al. (2010). Similarly, Espinace et al. (2013) describe a probabilistic hierarchical model using objects like doors or furniture as intermediate semantic representation of the room. *Histogram of Oriented Gradients*, *gabor* and *gray scale features* in combination with *Ran-*

dom Forest classifiers used by an *AdaBoost* implementation are employed. They apply the system to a mobile robot which enables them to increase the detection accuracy by using 3D data for a focus of attention mechanism based on geometrical and structural information.

The work presented in this chapter is inspired by the work of Swadzba and Wachsmuth (2011) who argue that the semantic approaches suffer from very constrained settings and the required extensive modeling efforts (Swadzba and Wachsmuth, 2008). They aim at a more holistic approach to scene classification in the spirit of the work of Torralba (2003) and Murphy et al. (2003). Both try to improve object detection through contextual priming and generate a model for jointly solving the detection and scene categorization tasks using 2D imagery. Whereas Swadzba and Wachsmuth (2011) rely on a combination of 3D surface features which reliably model the furniture's shape independent of the texture and 2D *gist* features which model the texture through wavelet image decomposition and have proven to work well for scene classification (Torralba et al., 2003). This approach of combining different types of features seems promising because it covers the different properties of a room structure that are particularly descriptive for distinguishing room types. This was also exploited by Martinez Mozos et al. (2012) in an approach using depth and gray scale images for creating histograms of *Local Binary Patterns* in a *Support Vector Machine* and *Random Forest* classification scheme.

Combination of Features and Classifiers However, one of the approaches presented in this chapter has the goal of combining different features *and* different classifiers which can be dynamically adapted to the current training set in order to find the best descriptive combination for categorization. Hence, a more general approach for the combination method needs to be found. In literature there is a large number of solutions for strategies of combining a variety of features. One quite widespread approach is to combine features at decision level using *Hidden Markov Models* (Oliver et al., 2004) or *Mixture of Gaussian Models* (Kapoor and Picard, 2005) and other strategies based on expert mixtures.

Another widespread approach is supervised ensemble learning which tries to combine a set of *weak learners* in order to build a single *strong learner*. One of the most prominent members of this family of machine learning al-

gorithms is *AdaBoost*, first introduced by Freund and Schapire (1997). It is an adaptive implementation of the classical *Boosting* definition in that sense that it adapts subsequent weak learners in favor of those instances mis-classified by previous classifiers. The famous face detection framework by Viola and Jones (2001) uses *AdaBoost* with a wide range of very simple *weak learners* based on *Haar Features*. The work mentioned before on scene recognition by Espinace et al. (2013) makes use of *AdaBoost* with several features types as well. The boosting step can easily be adapted to perform a decision making on classifier/feature combinations with respect to generalization qualities. Treptow and Zell (2004) present a strategy for choosing between different features for face detection. They use an evolutionary algorithm in the training step for *AdaBoost* to search for a new feature that results in a better classifier. This is particularly relevant if a huge amount of different classifiers is available, which is not true for the framework described in the following sections of this chapter. Another approach for selection of features for boosting was introduced by Opelt et al. (2006). They have developed a *Weak Hypotheses Finder* that uses a distance matrix of all features to find the most descriptive one. The computation of the distance matrix takes most of the time but can be done prior to boosting. This leads to a linear computation time to the number of training samples for finding the optimal weak hypothesis.

In this chapter I will present several approaches for visual interpretation of objects and areas. First, a system for categorization of furniture will be presented. Since only few pieces of furniture are manipulated on a regular basis the *ASM* cannot be used for segmentation of the candidates, so the presented approach includes a segmentation step. The categorization approach utilizes three-dimensional properties of the furniture for classification. Subsequently, a more general approach for visual recognition is presented which applies an ensemble classification scheme utilizing both 2D and 3D data. This method is used to realize a holistic approach for identification of rooms and functional areas of an apartment incorporating a spatial anchoring of features for the inclusion of peripheral information into the classification of the perceived scene.

4.1. Furniture Categorization

Classification of objects in real-world situation data is still a challenging task for computer vision. Especially categorization of furniture is difficult because the intra-class variety of each class is as big as only in few other categories. Using 2D data for this task is not very promising because the textures vary strongly from instance to instance, but likewise often resemble across different classes (e.g. wooden decor, matching finish of sitting suite). Likewise the shape of the furniture is not perceived very well using 2D data. But still, even though in 3D the shape seems to be a good hint for distinguishing different furniture categories, classical approaches often fail because the perceived *point clouds* mostly do not contain many distinguishable features in terms of corners, edges, curvature, or other geometric properties that are typically exploited by computer vision features. Furniture objects mostly consist of many planar or almost planar regions and hence do not differentiate much in comparison to other structures inside an apartment like walls, floors, doors, or windows. Additionally, also in shape some furniture categories have a high intra-class variability (e.g. chairs) and a low inter-class variability (e.g. couches and armchairs).

Considering these difficulties, the classification approach presented in this section makes use of the spatial layout of the different regions of a furniture object. It incorporates the *Implicit Shape Model (ISM)* method for learning the three-dimensional spatial relation between typical object regions, which allows a certain level of occlusion in the target scenes. For categorizing the appearance of a candidate, a probabilistic *Hough Voting* is performed that matches the perceived relations to the learned ones which allows to simultaneously recognize and localize objects of the learned categories in the scene. Therefore, a pre-segmentation of candidates is not necessary – concentrations of evidences in the whole scene give good hypotheses for the locations of objects. However, statements about the orientation of the target objects are not possible.

For the application described in Chapter 5 the localization capabilities are not required. Here, a pre-segmentation is applied, and the *Hough Voting* mechanism is slightly altered in order to only verify the category.

The work on furniture recognition presented in the following has been done in collaboration with Jens Wittrowski (see Wittrowski et al. (2013)).

4.1.1. Implicit Shape Model

The implemented *ISM* approach is based on a method first proposed by Salti et al. (2010). They suggest to use the original 2D *ISM* method (Leibe et al., 2004) in an extended form for 3D object categorization. The main benefit of *ISM* classification compared to other statistically based classifiers is that it considers the spatial relations of the object parts found in the models. They define the *Implicit Shape Model* for a category C as

$$ISM(C) = (I_C, P_{I,C}) \tag{4.1}$$

where I_C is an alphabet of typical local appearances of the selected object category (termed "codebook") and $P_{I,C}$ is a spatial probability distribution which specifies where each codebook entry may be found on an object. So the learned models contain the frequencies and relative positions of typical regions of the objects within the corresponding class. Specifically, the approach learns the possible geometric relations between the features and a reference point – preferably the object's center. This is realized by assigning a vector to the learned features which points to the reference point. If the same features and relations to the reference point of one model are found on a candidate object, this object can be classified as the corresponding class.

For the furniture recognition system presented here, the original 3D *ISM* algorithm of Salti et al. (2010) is adapted in two ways. First, the feature calculation step was adapted to the special requirements in the domain of furniture recognition. Typical indoor room scenes usually contain — because they are man-made — many planar surface structures. This includes general structures of the room itself like walls, floors, and doors, but this is also true for the furniture within the room, especially shelves, cupboards, and tables. Using 3D shape descriptors should focus on non-planar features of the furniture, because the surfaces do not contain sufficient descriptive power to distinguish furniture from the background and to describe the furniture's properties that enable categorization.

Secondly, an alternative approach to the original *Hough Space Voting* is presented which is used for feature position aware detection and classification of the shape models. The new approach allows to use an unlimited amount of training data while keeping the upper bound of computational effort at a constant level. Further it eliminates the need to use models of a correct real-world scale for training and classification of shape models.

Training Procedure

For the training of the classifier a web database of artificial 3D models of furniture is used. A successful learning scheme for furniture categorization using this database was demonstrated by Martinez Mozos et al. (2012). Since the features for the *Shape Models* are calculated from *point clouds*, the artificial meshes from the database need to be preprocessed. In order to receive realistic data that is similar to the expected real-world data in the prediction process, virtual 3D scans of the models are created (see Figure 4.1). These emulate the use of a depth sensor for creating realistic *point clouds* from 12 virtual positions around the target object. Visualizations of the models used can be found in Appendix C.

Figure 4.1.: Left: Furniture meshes from the database. Right: virtual scans.

As stated above, it is important to focus on non-planar regions of the objects in order to find descriptive features. This is realized by performing a boundary estimation on the given *point clouds* in order to receive a set of keypoints for feature calculation. The boundary estimation is based on angle differences of normals in the neighborhood of a target point.

For describing the keypoints found on the object's boundaries the *Signature of Histograms of Orientations (SHOT)* descriptor is used (Tombari et al., 2010). This local descriptor aims at characterizing a keypoint by generating a description of the neighborhood (support) of a target point.

One of the reasons for the choice of this descriptor is the fact that it is able to define an unique local reference frame to the target point. The detected characteristics of the surface in the surrounding are stored using the local coordinates which makes it rotation- and viewpoint-invariant. The descriptor vector is calculated by assigning the neighboring points to spatial bins which are defined by performing 8 azimuth, 2 elevation, and 2 radial divisions of a virtual sphere around the target point. For each bin a histogram over the cosines of the angles between normals corresponding to the points within the bin and the keypoint's normal is calculated. The use of cosines of the angles has the effect that — when using equally spaced bins — the histogram is more coarse for the angles parallel to the keypoint's normal and more fine grained for angles orthogonal to the keypoint's normal. As the points with normals that have a large angular distance to the keypoint's normal are the most informative ones, a finer binning supports the descriptive power of the descriptor. The 32 spatial bins around the keypoint containing 11-dimensional histograms of directions result in a 352-dimensional descriptor vector which is ultimately normalized so that it is independent of the number of points in the neighborhood.

The local reference frame of the feature needs to be repeatable and unambiguous in order to be able to generate the same descriptor independent of the viewpoint. It therefore uses an adapted Principal Component Analysis of the neighboring data. The data for the calculation of the covariance matrix is weighted by the distance to the target point:

$$C = \frac{1}{\sum\limits_{i:d_i \leq R} (R - d_i)} \sum_{i:d_i \leq R} (R - d_i)(p_i - p)(p_i - p)^T \tag{4.2}$$

where R is the the radius of the sphere and d_i is the euclidean distance between p_i and the keypoint p. This increases the repeatability of the local reference frame in presence of clutter. In order to disambiguate the axes of the found principal components the algorithm ensures that the sign of the eigenvectors is coherent with the majority of points it represents (see Tombari et al., 2010, for more details).

These local descriptors are now used to generate a codebook describing typical parts of furniture. It is generated by clustering *SHOT* descriptors from all training samples in the feature space using the *K-Means* clustering scheme. The centroids of the corresponding clusters define the words for

the codebook of the size K.

In order to generate a *Shape Model* for each class the calculated *SHOT* descriptors on the training samples of that class activate codewords using the Nearest-Neighbor search. Additionally, each codeword builds up a histogram of voting directions. This strategy is different from the original 3D *ISM* approach. Every keypoint that actives a corresponding codeword votes for a direction that points to the object's center. The voting direction is represented in the descriptor's local reference frame. Hereby the codeword is applied with a probabilistic belief about the direction in which the object's center is located. This information is later used for the *Hough Space Voting* mechanism. So the *Shape Model* for a category of furniture consists of a set of activated codewords with an individual vote direction histogram assigned to it. The frequencies of codewords are implicitly represented in the vote direction histograms.

4.1.2. Ray-Based Hough Space Voting

The concept of *Hough Space Voting* is based on the *Generalized Hough Transform* approach for detecting shapes (Ballard, 1981). It is based on the idea that feature points vote for the specific geometric pose of a shaped object that is searched in the scene. Hypotheses for instances of the searched objects are generated from concentrations of votes in the *Hough Space*. This method was used for example by Tombari and Di Stefano (2010) in order to detect object instances in 3D scenes containing occlusion and clutter. They use vectors of specific length for voting for discrete bins in 3D *Hough Space*.

This method has two main disadvantages when applied to a categorization task using *ISMs*. First, the models that are used for training need to be scaled in the same way as the structures that are to be analyzed, because the vectors for voting represent the true distance between the feature points and the object's center. The databases used for training do not always fulfill this requirement, because they often use individual scales and units. An even worse problem is that furniture exists in many different sizes. A voting vector calculated for a small coffee table does not vote for the correct center of a larger dinner table.

The second disadvantage relates to the computational effort that is required for the proposed voting scheme. In the approach of Tombari and Di Stefano (2010) the number of voting vectors for every codeword can

potentially become very high since all feature points contribute an individual vector. Accordingly, also in the prediction phase this potentially large number of votes must be cast into the *Hough Space*. The mean number of votes v that are cast for each keypoint found in the scene can formally be estimated as $v = c \cdot \overline{v_c}$. Whereas c denotes the number of codewords that are activated by each feature and $\overline{v_c}$ is the mean number of votes that are assigned to each codeword. The latter depends on the number of training samples n, the mean number of keypoints $\overline{k}$ extracted from one sample and the codebook size C:

$$\overline{v_c} = \frac{n \cdot \overline{k}}{C} \tag{4.3}$$

Apparently the number of votes that need to be cast in the Hough Space increases linearly with the number of training samples and decreases with the size of the codebook. This is contradictory to the goal of achieving a good generalization through a wide range of training samples and a compact codebook.

The ray-based voting scheme described in the following, however, has a constant upper bound of votes per codeword and is independent of the number of training samples and the size of the codebook. Further, the previously described shortcomings regarding scale are overcome through scale-invariant voting. This is realized by ignoring the individual lengths of the voting vectors and instead regarding the votes as being rays with infinite length. Additionally, the voting rays of one codeword are clustered based on the direction they point to. To account for the frequencies of the directions clustered into a single ray and therefore also the frequency of the corresponding codeword, each ray is applied a voting weight. The clustering of votes is realized by creating a virtual unit sphere around the corresponding keypoint, aligned to the local reference frame provided by the *SHOT* descriptor (see Figure 4.2). Each voting ray is transformed into the local spherical coordinate system, represented by azimuthal angle θ and polar angle φ. The sphere is divided into $u \cdot q$ cones, where u and q denote the resolutions of bins on the azimuthal and polar axis respectively. Each cone is interpreted as a histogram bin collecting corresponding votes.

In the prediction phase for each codeword only one ray per bin in the histogram must be cast into the *Hough Space* instead of processing every voting vector individually. Furthermore, the rays corresponding to empty

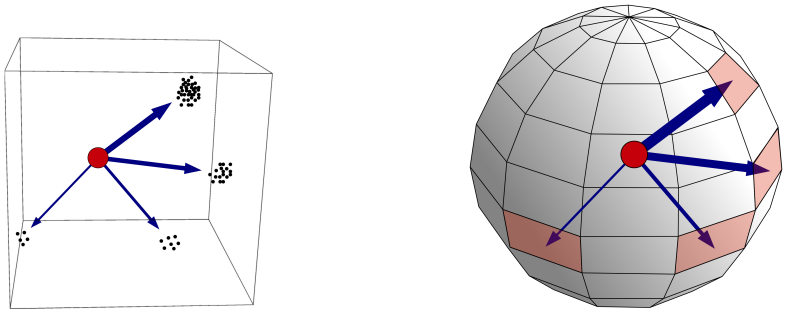


Figure 4.2.: Clustering of single vote vectors into rays by using a spherical histogram. The blue arrows represent vote rays for the red marked histogram bins, the thickness represents the vote weight.

bins can be skipped. Here, the middle azimuthal and polar angle of each bin in combination with the corresponding weight is used. In order to find concentrations of rays in the *Hough Space* a slightly different formalization is used. Instead of using a fixed three-dimensional grid, the ray-based *Hough Space* consists of overlapping spheres represented by a 3D point and radius. The voting is realized through counting intersection between rays and spheres (see Figure 4.3).

Assuming each ray r is represented by $r = p + \lambda v$ (where p is the point from where the vote is cast and v is the voting direction) the smallest distance between sphere center s_i and ray r_j can be calculated by

$$D_{i,j} = \frac{|(s_i - p) \times v_j|}{|v_j|} \tag{4.4}$$

Since the vote direction is normalized, the division by $|v_j|$ can be skipped. The dot product $(s_i - p) \cdot v_j$ can be used to determine whether the ray intersects with the sphere on the positive side of the feature point regarding the vote direction. If the distance is smaller than the sphere's radius and the dot product is positive, a vote is registered respecting the vote's weight.

The vote weight depends on the one hand on the probability of activating codeword c given a feature vector f which accounts for the uniqueness of the assignment of the current feature to the codeword. On the other hand

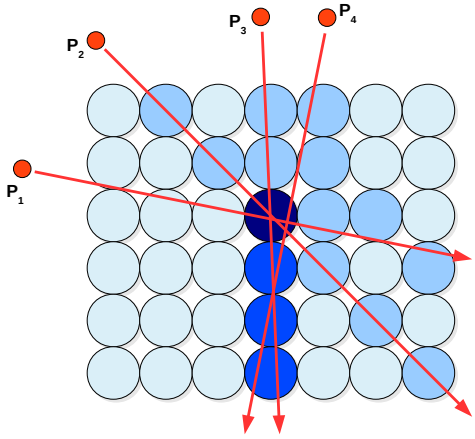


Figure 4.3.: Intersection of Hough Space spheres and vote rays. P_1 to P_4 are keypoints in the scene. For demonstration every keypoint only casts one ray and the spheres do not overlap.

it depends on the probability of a specific vote v given codeword c which accounts for the codeword's frequency in general and the direction's frequency in particular. The first part is calculated by

$$p(c_i|f) = \frac{d(c_i, f)}{\sum_j d(c_j, f)} \tag{4.5}$$

Here, $d(c_i, f)$ is a distance function proportional to the inverse of the euclidean distance between the vectors. The second part can be obtained from the direction histograms:

$$p(v|c_i) = \frac{h(v)}{\sum_j h(j)} \tag{4.6}$$

Here, $h(j)$ is the histogram value for direction j. So the final vote weight w is given by

$$w = \frac{d(c_i, f)}{\sum_j d(c_j, f)} \frac{h(v)}{\sum_j h(j)} \tag{4.7}$$

For every sphere in the Hough Space the vote weight of the intersecting rays are summed up. Maxima in the Hough Space give hypotheses for centers of objects in the corresponding category. The keypoints that vote for a specific maximum sphere can then be used for segmentation. A combination of outlier removal and clustering strategies allows to define a geometric hull of the detected furniture.

As mentioned above, this voting strategy reduces the number of votes that need to be cast in the prediction phase. For comparison, in the original *Hough Space Voting* scheme the average number of cast votes per codeword was specified through Equation 4.3 (page 92). In the presented approach the mean number of cast votes corresponds simply to the mean number of non-empty direction histogram bins. In the worst case every bin contains votes and thus the number of cast votes per codeword is $u \cdot q$. But neither the number of training samples, nor the size of the codebook has influence of the number of votes which allows to use large training sets and compact codebooks. Moreover, the presented changes make the voting scheme scale-independent.

4.1.3. Evaluation

The evaluation of this approach to furniture recognition is split into two parts. First, the ray-based *Hough Space Voting* mechanism is tested and compared with the work of Salti et al. (2010). For this, a furniture recognition task was performed on a set of artificial 3D models to show the general applicability of the voting scheme to furniture-related tasks. Subsequently, the performance of the complete *ISM* approach is tested on real-world scenes. Thereby the detection and categorization capabilities could be tested.

Results for Voting Scheme

For comparability the voting scheme was tested on the Aim@Shape Watertight (ASW) dataset[1] which contains 19 categories with 20 artificial 3D models each. The models were normalized by scaling them to a unit cube to provide equal conditions for comparison with Salti et al. (2010). For each category a *Shape Model* was learned with a codebook size of 1000. One half of the models of each category was used for training and the other for testing. For all tests a fixed number of 200 randomly chosen keypoints per model were used for training and 1000 for testing. The *SHOT* descriptor with a radius of 25cm was used for describing the keypoints. The *Hough Space* contained a single sphere with a radius of 10cm in the center of the models for measuring the sum of vote weights of the rays intersecting the sphere. The recognition result was given by the *Shape Model* which provided the highest measurement.

The results of this test are shown in the confusion matrix presented in table 4.1. The true categories of the tested models are listed row-wise, the recognition results are represented by columns. The average rate of correct classifications is **82%**, while Salti et al. (2010) achieved a not significantly different value of **81%**. This shows that the altered voting scheme is evenly well suited for this kind of recognition of furniture. However, the strength of the new approach — namely the scale-invariance and constant complexity while voting — is not tested with this experiment. Therefore a second test using real-world scenes was performed.

[1] `http://shapes.aim-at-shape.net/` (visited: February 17, 2015)

	Human	Cup	Glasses	Airplane	Ant	Chair	Octopus	Table	Teddy	Hand	Plier	Fish	Bird	Armadillo	Bust	Mech	Bearing	Vase	Fourleg
Human	0.6				0.2									0.2					
Cup		1.0																	
Glasses			0.9	0.1															
Airplane				1.0															
Ant					0.9					0.1									
Chair						0.8		0.2											
Octopus							0.7			0.2				0.1					
Table		0.1						0.9											
Teddy									1.0										
Hand	0.1						0.1			0.7								0.1	
Plier											1.0								
Fish							0.1					0.9							
Bird												0.2	0.8						
Armadillo														1.0					
Bust															0.4			0.6	
Mech		0.1														0.9			
Bearing										0.2		0.2					0.6		
Vase		0.2																0.7	0.1
Fourleg												0.1							0.9

Table 4.1.: Confusion matrix for the Aim@Shape Watertight dataset. The rows show the true categories of the tested models and the columns represent the recognition results. Results are expressed as percentages.

Results of 3D ISM on Real-World Scenes

In order to test the detection and categorization capabilities of the custom 3D *ISM* approach, scenes form the *New York depth dataset (V2)* were used (Silberman et al., 2012). This dataset contains indoor scenes captured with a Microsoft Kinect depth sensor. A large amount of the data is labeled with furniture categories which gives a well suited ground truth for testing detection and categorization. The used scenes are depicted in appendix C.

For the tests the *Shape Models* were trained using the *SketchUp 3D Warehouse*[2] database which contains many artificial 3D models of furniture. 50 models for each of the 6 trained categories (chair, table, couch, bed, shelf, cupboard) were used. Since 12 views for the virtual scanning of the meshes

[2] `http://3dwarehouse.sketchup.com` (visited: February 17, 2015)

	Chair	Couch	Bed	Cupboard	Shelf	Table
Chair	0.45	0.08	0.10	0.10	0.04	0.23
Couch	0.15	0.33	0.24	0.14	0.05	0.10
Bed	0.09	0.08	0.60	0.10	0.02	0.11
Cupboard	0.04	0.11	0.16	0.53	0.10	0.06
Shelf	0.17	0.08	0.15	0.25	0.25	0.11
Table	0.08	0.05	0.03	0.05	0.02	0.78

Table 4.2.: Results of the furniture categorization. The rows show the categories that were tested and the columns represent the categorization results as percentages.

were used (see Section 4.1.1) this results in 600 samples per category. As in the previous test a codebook size of 1000 words was used. The *SHOT* descriptors were calculated on a 10cm radius. The voting direction histograms were learned with an azimuthal resolution of 90 and a polar resolution of 45 bins which leads to a maximum angle error of 2° when clustering the actual vote vectors.

For a preliminary baseline test of the categorization capabilities of the *ISM* approach the dataset was split into 30 models of each category for training and 20 for testing. Like in the ray-base Hough Space voting evaluation (see Section 4.1.3) a single voting sphere was defined in the center of the models accumulating the voting weights of the intersecting rays. Table 4.2 shows the confusion matrix of correct categorization percentages. On average **49%** of all models were categorized correctly.

For the main test 30 scenes of the *New York depth dataset* were chosen that contain at least one object of the trained categories. The *Shape Models* were trained as described in Section 4.1.1 with all 50 models for each category. The *Hough Space* was filled with overlapping spheres of 20cm radius. As a measure for accepting detections, all spheres across all categories with accumulated weight of at least 75% of the global maximum sphere were chosen.

The detected hypotheses have been compared with the ground truth from the labeled database. A hypothesis is considered correct if the corresponding

sphere's center point has an euclidean distance to the ground truth location of less than half of the mean training model's width. Hypotheses with a larger distance are considered false positives. Neighboring spheres (based on the same distance criterion) with identical categorization result are merged.

Category	TP	FP	GT	Precision	Recall
Chair	37	42	86	0.47	0.43
Couch	29	23	50	0.56	0.58
Bed	23	34	32	0.4	0.72
Cupboard	18	31	38	0.37	0.47
Shelf	22	21	34	0.51	0.65
Table	31	22	37	0.58	0.84

Table 4.3.: Recognition results on real-world indoor scenes. For each category the amount of true positive detections (TP), false positives (FP), and the number of objects in the ground truth (GT) are presented.

The results of this evaluation are reported in table 4.3. Objects with a certain conspicuousness in the scene are found correctly in most cases. However, as expected, heavily occluded objects or those that are located in the background are often not found correctly (see Figure 4.4). Both phenomenons can be explained by the lack of descriptive data. In the occlusion case only few keypoints are found so that few rays vote for the corresponding category which leads to low accumulated weights in the spheres. The detection in the distant location case suffers from the increasingly low resolution of the *point cloud* the farther it is from the camera. This results in low quality feature descriptors in addition to few voting rays.

4.1.4. Summary & Discussion

The previous sections have shown that the altered version of 3D *ISM* works well for detecting and categorizing furniture in real-world scenes. The main advantage of the ray-based Hough Space Voting is that it allows an unlimited number of training data to be used while keeping the upper bound of computational effort constant. Beyond this, it is able to deal with arbitrary sizes, both in training and prediction. Since the calculation of rays intersecting spheres is computationally expensive compared to sorting points into 3D bins, the advantages of this method will outweigh when using a large

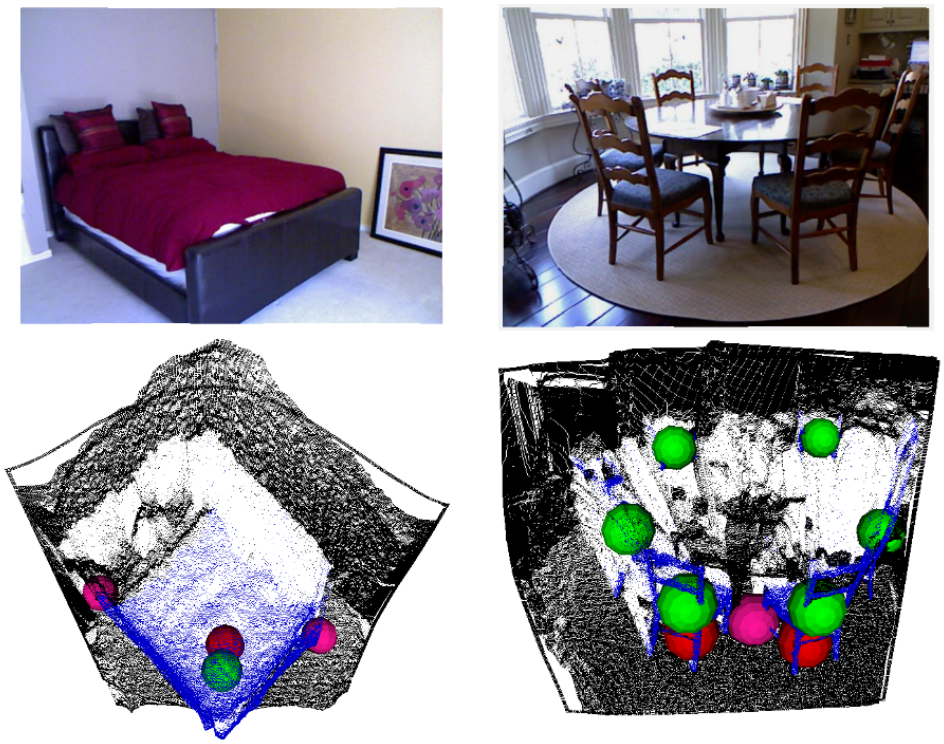


Figure 4.4.: Images and *point clouds* of sample test scenes. The green spheres represent ground truth center locations. The red and pink spheres represent true and false positive hypotheses respectively.

number of training data. Further speed up might be achievable by using a classical 3D grid and finding intersections using the Bresenham line algorithm, though this would reject the kind of linear interpolation that is implicitly performed by using overlapping spheres. Furthermore, the performance in detecting furniture in real-world scenes could possibly enhanced through further shape verifications steps to reduce false positives. A more sophisticated ray concentration analysis strategy might improve the number of true positive. The evaluation of these adaptations is left for future work.

Nevertheless, *ISMs* are apparently an effective means for detecting static

instances like furniture and applying semantic annotations to these objects in a mobile robot's environment. They exploit the shape of the geometric structures which is suspected to be the most descriptive feature for distinguishing various furniture categories (see Section 4.1). This allows a robot companion to enrich its *situation model* with knowledge about the identities of the furniture in its surrounding. This is particularly important for verbal referencing. Thus this approach is also used in Chapter 5 which describes a system for grounding verbal descriptions of furniture. The accuracy of the recognition is not crucial here, because ambiguities in the recognition should be rectified using the auditory modality.

However, there are different kinds of semantic analysis mentioned in the introduction to this chapter that require the exploitation of other features. A more general approach to utilizing different kinds of features for recognizing manipulable objects and whole rooms is described in the next section.

4.2. Classification of Household Objects

In typical real-world situations future robot companions will face a wide variety of object types that need to be recognized. Accordingly, they need to consider many different features in order to distinguish between those arbitrary objects. The shape is certainly one of the features that need to be taken into account, but texture or color might also be informative for the recognition process. Many objects like books and different kinds of boxes have a canonical shape and are only distinguishable via their texture. Their shape can here only serve as a preselection for certain categories, but does not allow identification. However, in other cases the shape might be the key feature for distinguishing objects. The example of furniture having uniform surfaces (e.g. wooden decor, matching finish of sitting suite) but representing different types of objects has already been mention in the introduction. Another example from the domain of manipulable objects is tableware. Only relying on texture and color is not sufficient to distinguish textureless plates from cups. Swadzba and Wachsmuth (2011) already pointed out that the classification of scenes also benefits from both shape and texture.

The idea for the classification approach described in the following sections is the combination of different features and classifiers in order to obtain a mechanism that automatically detects an appropriate combination of these for the current classification problem. The described system implements a boosting technique for combination of several *weak classifiers*. This approach will be used for classification of manipulable objects as well as for a room categorization scheme. Boosting is also chosen with respect to the high variance within the room categories because of its ability to improve training and generalization results.

The main idea of the boosting approach is to convert a set of so called *weak* or *base classifiers* into one strong classifier. The term "weak" implies that the individual classifiers used for boosting only have a weak descriptive power when used in isolation. This is not true for all of the used classifiers, which is why I will prefer the term *base classifier* in the remainder of this chapter. One of the most commonly used *base classifiers* for boosting are *Decision Trees(DTrees)*. They are also used by Freund and Schapire (1997) in combination with classical *AdaBoost*. Taking the results from Schwenk and Bengio (1997) into account, different configurations of *Multilayer Perceptrons(MLPs)* are also promising as they achieved good results

for boosting. To have a comparison to the behavior of a so-called strong classifier, *Support Vector Machines(SVMs)* with an RBF kernel will also be taken into account for boosting. It has been shown that strong classifiers usually decrease the accuracy of boosting. However,Li et al. (2008) propose that *SVMs* with configurations that lead to at least acceptable classification results are, in combination with boosting, able to perform better than boosted classical *base classifiers* like *MLP* and *DTree*. Additionally to these classifiers the implemented system uses a simple *k-Nearest Neighbor* classifier and a custom color distribution detector (described in Siepmann et al. (2014)).

In order to account for the different features that make an object distinguishable from others, a variety of different types of feature descriptors are chosen for this work. Local features for 2D and 3D data are used to describe local regions, this increases the classifiers' robustness against occlusion. As a powerful descriptor for texture in 2D data the well known *Scale-Invariant Feature Transform (SIFT)* (Lowe, 2004) and *Speeded Up Robust Features (SURF)* (Bay et al., 2006) are used. Moreover, the quite recently introduced but promising feature descriptors *Oriented FAST and Rotated BRIEF (ORB)* (Rublee et al., 2011), *Binary Robust Invariant Scalable Keypoints (BRISK)* (Leutenegger et al., 2011), and *Fast Retina Keypoint (FREAK)* (Alahi et al., 2012) are also used.

Local 3D features like *Fast Point Feature Histogram (FPFH)* (Rusu et al., 2009b) and *SHOT* (Tombari et al., 2010) as well as its extension to colored *point clouds COLORSHOT* (Tombari et al., 2011) are used for this approach as they reliably describe different kinds of 3D shape. Their applicability to furniture recognition has been shown in the previous sections (4.1.1), while Swadzba and Wachsmuth (2011) used 3D features successfully for scene categorization. They define a global 3D feature based on properties of planes taken from a 3D scene in combination with the *gist* feature introduced by Oliva and Torralba (2001). Both features are incorporated into this approach when applied to the room categorization task, whereas for recognition of manipulable objects they seem not very suitable. In order to describe detected objects in a way that is comparable to other detected objects the *BoW* approach is used. A codebook is generated for each local 2D and 3D feature type to estimate global histogram features of a complete object or scene. This approach does not provide any segmentation. It requires the training samples and candidates for prediction to be pre-segmented.

4.2.1. Boosted Classification

As already mentioned, one of the most effective and often used boosting approaches is *AdaBoost* by Freund and Schapire (1997). The main difference between *AdaBoost* and most other boosting algorithms is the strategy of applying adjustable weights ω_i to each sample in the training set $(\vec{x}_i, y_i, \omega_i)$ with $i = 1...n$. Where $\vec{x}_i$ belongs to some instance space X and y_i denotes some label from the label space Y of size K. The algorithm adapts the weights in each boosting step according to the results of the currently trained *base classifier*. The original *AdaBoost* approach was developed as a classifier for binary problems. Zhu et al. (2009) present a new multi-class boosting algorithm called *Stagewise Additive Modeling using a Multi-Class Exponential Loss Function (SAMME)* which extends the classical *AdaBoost* to be used for multi-class problems.

Both boosting algorithms call a given *base classifier* $h : X \to Y$ repeatedly in a series of rounds $t = 1, \ldots, T$. Initially all weights are set equally, but on each round the weights of incorrectly classified examples are increased so that the *base classifier* has to focus on the previously misclassified examples in the training set. The goodness of a *base classifier*'s instance h_t in round t is measured by its error:

$$\epsilon_t = \frac{\sum\limits_{i:h_t(x_i)\neq y_i} \omega_{i,t}}{\sum\limits_{i:1..n} \omega_{i,t}} \tag{4.8}$$

Notice that the error depends on the sample weights on which the *base classifier* was trained. This leads to a particularly low error and thereby to a high rating if the previously misclassified samples are now correct. If in practice the *base classifier* does not support weighted samples (which is the case for the chosen implementations), a subset of the training examples can be sampled according to the weights which then will be used for training.

The trained set of T *base classifier* will ultimately yield the additive stagewise model $C(\vec{x})$ for prediction. Each classifier in this model has a parameter α_t which measures the importance that is assigned to h_t.

$$\alpha_t = \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \underbrace{+\log\left(K-1\right)}_{\text{new term}}. \tag{4.9}$$

Note that K denotes the total number of classes and that α_t gets larger as ϵ_t gets smaller. A classifier is accepted only if $\alpha_t > 0$. If this assumption is not true the classifier performs worse than random guessing and is discarded. This is because the original theory of boosting assumes that each used classifier performs better than random guessing. In a two-class problem this corresponds to an error of less than 0.5 which is equivalent to $\alpha_t > 0$. The additional term in equation 4.9 is the main difference between classical *AdaBoost* and *SAMME*. It enables *SAMME* to be used for multi-class problems which requires to accept *base classifier* with an error greater than 0.5. More precisely: a classifier is accepted if $\epsilon_t < (K-1)/K$ (or $(1-\epsilon_t) > 1/K$). This ensures that a classifier performs better than random guessing depending on the number of used classes. In the case of $K = 2$ the algorithm reduces to classical *AdaBoost* with a probability of 0.5 for each class. For more details and theoretical justification see Zhu et al. (2009). Figure 4.5 illustrates the correlation of the classifier weight for the additive model and the classification error for different numbers of classes. For $K = 2$ the α-curve is identical with classical boosting whereas for more than two classes α is still positive for $\epsilon_t > 0.5$ (and $\epsilon_t < (K-1)/K$).

Subsequently, the weights w_i of the samples are adjusted according to the classification error in the current round:

$$\omega_{i,t+1} \leftarrow \omega_{i,t} \cdot \begin{cases} e^{-\alpha_t} & : h_t(\vec{x_i}) = y_i \\ e^{\alpha_t} & : h_t(\vec{x_i}) \neq y_i \end{cases}, \quad i = 1, 2, \ldots, n. \tag{4.10}$$

This equation increases the weight of examples misclassified by h_t and decreases the weight of correctly classified samples. Thus, the weight tends to concentrate on "hard" examples. Finally, after passing all rounds, the additive stagewise model can be created which consists of T pairs of *base classifier* instances h_t and classifier weights α_t.

For prediction one simply has to find the class k for which the additive weight of the stagewise model is the highest. In other words, the classification result of the resulting *strong classifier* H is

$$H(\vec{x}) = \underset{k}{\operatorname{argmax}} \sum_{t=1}^{T} \alpha_t \cdot \begin{cases} 1 & : h_t(\vec{x}) = k \\ 0 & : h_t(\vec{x}) \neq k \end{cases} \tag{4.11}$$

The complete procedure is shown as pseudo code in Algorithm 3.

Algorithm 3 SAMME pseudo code

Require: training set $(\vec{x}_i, y_i, \omega_i), \quad i = 1, \ldots, n$

Require: base classifier h

1: Initialize weights for all samples: $w_i \leftarrow \frac{1}{n}, \quad i = 1, \ldots, n$

2: **for all** $t = 1, \ldots, T$ **do**

3: Fit a base classifier $h_t(\vec{x})$ to the training data using weights w_i

4: Computer weighted error for h_t:

$$\epsilon_t = \frac{\sum\limits_{i:h_t(x_i)\neq y_i} \omega_{i,t}}{\sum\limits_{i:1..n} \omega_{i,t}}$$

5: Computer classifier weight:

$$\alpha_t = \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right) + \log(K-1)$$

6: Set new sample weights:

$$\omega_{i,t+1} \leftarrow \omega_{i,t} \cdot \begin{cases} e^{-\alpha_t} & : h_t(\vec{x_i}) = y_i \\ e^{\alpha_t} & : h_t(\vec{x_i}) \neq y_i \end{cases}, \quad i = 1, 2, \ldots, n.$$

7: Re-normalize w_i

8: **end for**

9: **return** $H(\vec{x}) = \underset{k}{\operatorname{argmax}} \sum\limits_{t=1}^{T} \alpha_t \cdot \begin{cases} 1 & : h_t(\vec{x}) = k \\ 0 & : h_t(\vec{x}) \neq k \end{cases}$

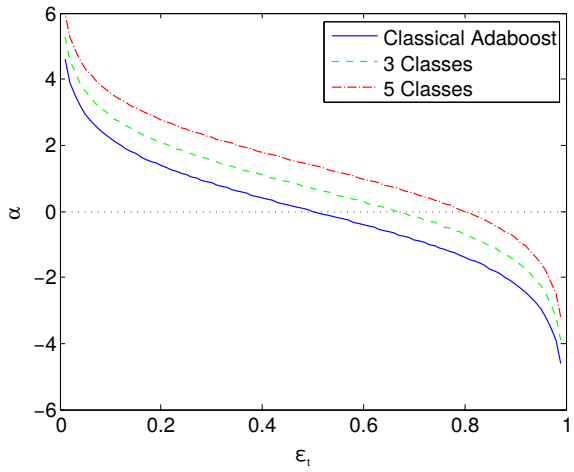


Figure 4.5.: The plot shows the coherence between error and classifier weight for 2, 3 and 5 classes. For two classes, the graph shows the same behavior as for classical *AdaBoost*. For a higher number of classes, the base classifiers are allowed to have a higher error on the training samples.

Multiple Classifiers

The *SAMME* approach does support multi-class data but is not able to process more than one classifier type. In order to achieve this the *SAMME* algorithm is modified to perform an exhaustive search during steps 3 and 4 (see Algorithm 3). It therefore trains each available classifier-feature combination from a set of classifier settings and features. The combination that reaches the smallest error ϵ_t on the weighted training set is added to the additive model in Equation 4.10. This algorithm will be denoted as *Exhausive SAMME (E-SAMME)* (see Algorithm 4).

All permutations of *base classifier* and feature are taken into account for each training step. Thus, the optimal classification regarding the available set of classifier settings and features can be found for the weighted training set in each step.

However not every type of *base classifier* supports training with weighted samples. As already mentioned above, the weights are used to resample the training set. For the described system two alternatives were implemented:

Algorithm 4 E-SAMME pseudo code

Require: training set $(\vec{x}_i, y_i, \omega_i), \quad i = 1, \ldots, n$

Require: set of feature-classifier combinations $h_c, \quad c = 1, \ldots, C$

1: Initialize weights for all samples: $w_i \leftarrow \frac{1}{n}, \quad i = 1, \ldots, n$

2: **for all** $t = 1, \ldots, T$ **do**

3: Initialize $\epsilon_{\min,t} = 1$

4: **for all** $c = 1, \ldots, C$ **do**

5: Fit a base classifier $h_{c,t}(\vec{x})$ to the training data using weights w_i

6: Computer weighted error for $h_{c,t}$:

$$\epsilon_{c,t} = \frac{\sum\limits_{i:h_{c,t}(x_i) \neq y_i} \omega_{i,t}}{\sum\limits_{i:1..n} \omega_{i,t}}$$

7: Computer classifier with minimum error:

$$\begin{aligned} \epsilon_{\min,t} &\leftarrow \min\left(\epsilon_{c,t}, \epsilon_{\min,t}\right) \\ h'_t &\leftarrow \underset{h_{c,t}}{\operatorname{argmin}}\left(\epsilon_{c,t}, \epsilon_{\min,t}\right) \end{aligned}$$

8: **end for**

9: Computer classifier weight:

$$\alpha_t = \log\left(\frac{1-\epsilon_{\min,t}}{\epsilon_{\min,t}}\right) + \log(K-1)$$

10: Set new sample weights:

$$\omega_{i,t+1} \leftarrow \omega_{i,t} \cdot \begin{cases} e^{-\alpha_t} & : h'_t(\vec{x_i}) = y_i \\ e^{\alpha_t} & : h'_t(\vec{x_i}) \neq y_i \end{cases}, \quad i = 1, 2, \ldots, n.$$

11: Re-normalize w_i

12: **end for**

13: **return** $H(\vec{x}) = \underset{k}{\operatorname{argmax}} \sum\limits_{t=1}^{T} \alpha_t \cdot \begin{cases} 1 & : h'_t(\vec{x}) = k \\ 0 & : h'_t(\vec{x}) \neq k \end{cases}$

Roulette Wheel Selection This method selects a subset randomly from a set of weighted samples by randomly choosing a number n_r between 0 and 1. The weights of the samples are summed up until the sum exceeds the random number n_r. The corresponding sample will be added to the subset until a maximal number of samples Z is reached. The samples with high weights are more likely to be chosen which has the effect that the classifier is trained more often on samples that are badly classified.

Maximum Selection This method chooses the first Z samples with the highest weights as the subset for training. This has the effect that no sample is chosen more than once. So the *base classifier* is trained on more different samples which leads to better generalization. The number Z is computed via a sample factor z_f: $Z = z_f \cdot n$.

Since the *base classifiers* are trained only on the weighted subsets (step 5), but the training error is calculated on the complete training set (step 6) this error also serves as generalization measure for classifier $h_{c,t}$.

The prediction in *E-SAMME* works analogous to the common *AdaBoost* method. Each base classifier in the additive model of *E-SAMME* predicts the identity of the given candidate. For each class k the classifier weights α_t of the *base classifiers* that predict class k are summed up. Finally, the class with highest accumulative classifier weight is chosen as final prediction result (see step 13).

In the actual implementation which is used on the robot (e.g. in RoboCup@Home) the set of classes include an additional *unknown* category so there are $K + 1$ classes instead of K. Depending on the *base classifier* and the specific task the classification is used for, the *unknown* class implicitly emerges through thresholding the prediction results of the *base classifiers* or by explicit demonstration of negative samples.

4.2.2. Evaluation

For evaluation of the general performance of the presented *E-SAMME* classification approach, a typical instance recognition task was prepared. Note that the algorithm's applicability to categorization tasks is described in Section 4.3 and evaluated in Section 4.3.3. For the execution of the tests a dataset of 15 typical manipulable household objects containing on average

Figure 4.6.: Example images of each object in the test dataset for evaluation of *E-SAMME*.

$\sim$ 53 samples for each class was assembled. The objects were selected considering a wide variety in texture and shape. Deliberately, the set contains several box-shaped and cup-shaped objects, as well as texture-poor objects in order to challenge the exploitation of different object features. All objects have been pre-segmented in 3D using a simple plane extraction and a clustering algorithm based on *point clouds*. Each sample contains a pre-segmented 2D image of the object and a 3D *point cloud*. The objects were recorded using an ASUS Xtion Pro Live RGB-D sensor and a regular consumer camera. Thereby roughly half of the samples contain a high-resolution, high-quality 2D image from the consumer camera and the other half a comparably low resolution image from the RGB-D sensor. This increases the variety within the dataset. The collocation of the set and choice of sensors is inspired by the requirements in RoboCup@Home. Figure 4.6 depicts sample images of each object class.

All tests presented in the subsequent sections were performed using the following configurations: The number of boosting steps was set to 20. To avoid that outlier results of the classification are analyzed, a $k = 5$ fold

cross-validation was applied and the error for each step was averaged over the k folds. The set of classifiers was composed of different settings of *SVMs*, *DTrees*, *MLPs*, *k-Nearest Neighbor* classifiers, and custom color distribution classifiers. As local visual features the deployed system used *SIFT*, *SURF*, *ORB*, *FREAK*, *FPFH*, and *SHOT* – likewise with varying parameters. As sampling method for weighting the training data, the *maximum selection* strategy with a sample factor of $z_f = 0.6$ was used. The classification results have been recorded in confusion matrices. Since not all explicitly model a rejection of candidates, an "unknown" class is not considered in this evaluation. However, in principle *E-SAMME* is capable of handling rejection.

For comparison with other state-of-the-art algorithms for instance recognition the dataset was evaluated with selected classifiers typically used for this task. For all selected classifiers parameters were optimized in order to create equal conditions for all cases. Local features have been combined by the implementation to global descriptors using the *Bag of Words (BoW)* method. Figure 4.7 shows average results over three trials of a $k = 5$ cross-validation in form of the *Correctly Classification Rate* (CCR). The abbreviation "COLD" means the color detector described in Siepmann et al. (2014). "E-SAMME-ALL" means the developed boosting algorithm using all classifiers and features mentioned above. However, "E-SAMME-2D" and "E-SAMME-3D" mean a restriction of the previous configuration to 2D and 3D features respectively. The other abbreviations are self-explanatory.

The *E-SAMME* algorithm reaches the highest scores. The difference between E-SAMME-ALL (CCR: 0.854) and the *Support Vector Machine* using *SURF* features (CCR: 0.822) has a statistical significance according to an independent two-sample t-test ($p < 0.05$). The *E-SAMME* configuration using only 2D features (CCR: 0.818) performs equally well as the SVM-SURF configuration. Using only 3D features (CCR: 0.629), however, works significantly worse than the combination of 2D and 3D features. Additionally, the results of the E-SAMME-ALL condition compared to the results of the E-SAMME-3D are shown in tables 4.4 and 4.5 respectively (more results can be found in Appendix D). In the former case the only slight systematic confusion can be observed for class "paper cup". The confusion with classes "cup1" and "cup2" is not surprising because of the similar shapes. However, in the E-SAMME-3D case many confusions can be observed, mainly for objects that have similar visual properties.

	tea	paper cup	cup1	duster	wall mount	tape	milk	coke	cup2	book	stapler	sponge	cup0	hair gel	joy pad
tea	0.911		0.022				0.044			0.022					
paper cup		0.686	0.086		0.029		0.029	0.029	0.086		0.029	0.029			
cup1	0.017	0.033	0.867				0.017		0.050		0.017				
duster				0.986							0.014				
wall mount	0.060		0.020		0.900					0.020					
tape			0.018			0.782					0.036	0.055	0.055	0.036	0.018
milk					0.020		0.940	0.020		0.020					
coke	0.025		0.025				0.025	0.825		0.050			0.025		0.025
cup2	0.017		0.033						0.917		0.033				
book		0.044	0.022	0.022	0.022		0.044			0.778	0.022			0.044	
stapler			0.043	0.014	0.014				0.014	0.014	0.886				
sponge			0.018			0.073		0.018			0.018	0.855			0.018
cup0			0.022		0.022	0.022				0.022			0.911		
hair gel				0.020	0.040	0.020	0.040	0.040		0.040	0.020		0.020	0.700	0.060
joy pad				0.067			0.022		0.022					0.022	0.867

Table 4.4.: Confusion matrix of the E-SAMME-ALL condition. Rows: categories tested. Columns: classification results.

The good results of the E-SAMME-2D case are not surprising when considering the average amount of used features in the boosting process. Figure 4.8 shows the average feature type distributions for three different cases. It is obvious that the *SURF* feature describes the given dataset best, since it

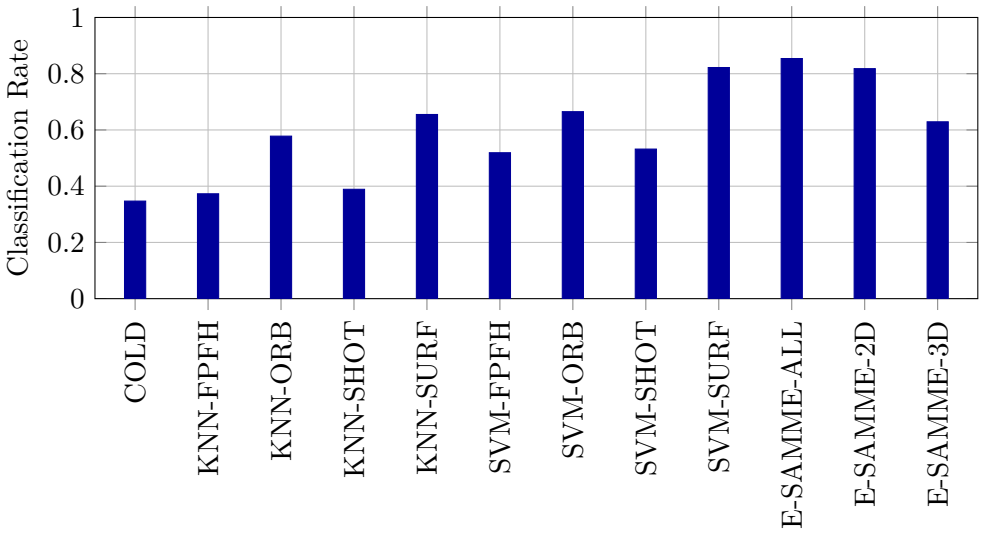


Figure 4.7.: Recognition results on household objects over $k = 5$ folds.

	tea	paper cup	cup1	duster	wall mount	tape	milk	coke	cup2	book	stapler	sponge	cup0	hair gel	joy pad
tea	0.600				0.200		0.022	0.022		0.022		0.133			
paper cup	0.029	0.314	0.143		0.029	0.057		0.029	0.257		0.029		0.086		0.029
cup1		0.017	0.617					0.017	0.167				0.167		0.017
duster			0.014	0.900			0.014					0.014		0.014	0.043
wall mount	0.120	0.040		0.020	0.480		0.060		0.020	0.040	0.040	0.140			0.040
tape	0.018	0.055	0.018	0.018	0.055	0.600		0.018			0.109	0.018	0.018	0.055	0.018
milk	0.020			0.040	0.080		0.700	0.020			0.060	0.040		0.040	
coke			0.022		0.022	0.022		0.622	0.044		0.133	0.022		0.089	0.022
cup2		0.033	0.167			0.017			0.617		0.033		0.133		
book	0.044									0.956					
stapler		0.029			0.029	0.086	0.029	0.043	0.014		0.729		0.014	0.014	0.014
sponge	0.091	0.018			0.182		0.036	0.018		0.018	0.018	0.564		0.036	0.018
cup0		0.044	0.244				0.022	0.044	0.244				0.378	0.022	
hair gel				0.060	0.040		0.080	0.060			0.020	0.040	0.020	0.620	0.060
joy pad	0.022			0.222	0.022	0.022	0.067	0.089			0.022		0.022	0.089	0.422

Table 4.5.: Confusion matrix of the E-SAMME-3D condition. Rows: categories tested. Columns: classification results.

is used in 0.663 of all boosting steps. However, the *SHOT* descriptor is used in 0.195 and the *ORB* descriptor in 0.122 of all cases. It seems that the high recognition rate of the *E-SAMME* algorithm profits from the availability of different types of features.

In order to analyze this in more details, the dataset was split into those objects which are believed to mainly differentiate in shape (OBJ-S) and those expected to mainly differentiate in texture (OBJ-T). The cross-validation

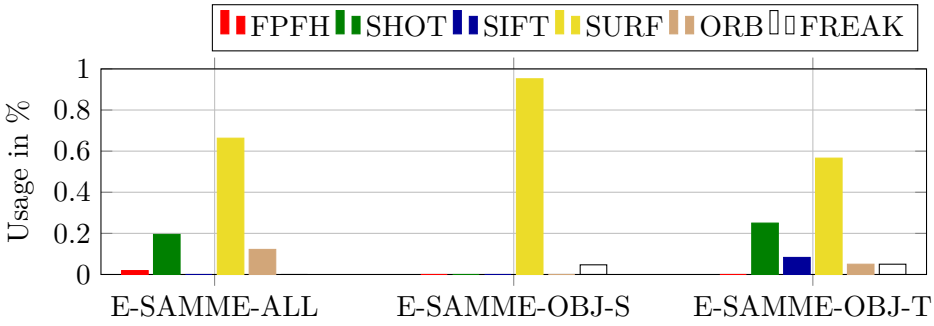


Figure 4.8.: Average amount of features used in a trained E-SAMME classifier. The default case using the complete dataset (ALL) is compared to cases using subsets mainly differentiating in shape (OBJ-S) and texture (OBJ-T) respectively

was performed on both subsets (see Figure 4.8). Interestingly, in the OBJ-T case the boosting chose a comparably high amount of 3D features, although this subset was believed to be distinguishable mainly by 2D texture features. Obviously, the assumptions about the most discriminating features of the two groups have not been correct. However, this analysis shows that the developed algorithm not generally prefers the *SURF* feature – there are conditions in which other features dominate.

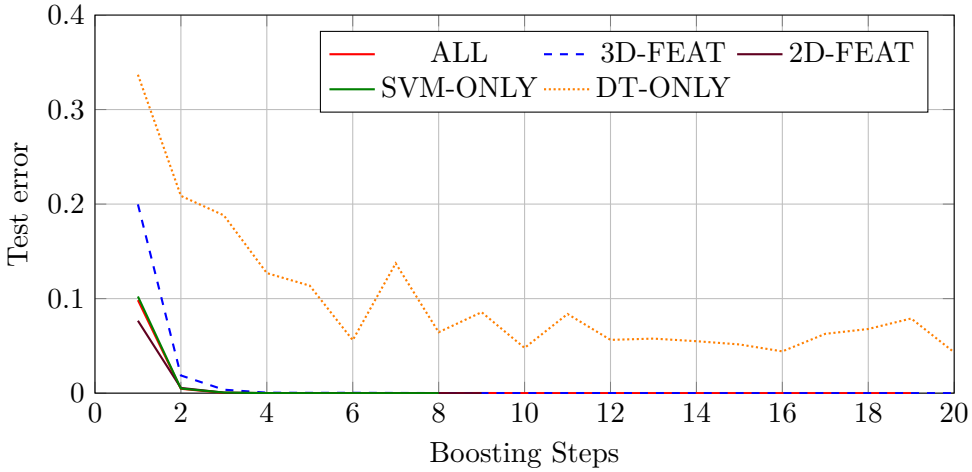


Figure 4.9.: Mean test error over 5 training runs using the k fold cross validation. The test error progress over 100 boosting steps is plotted for four combinations of feature types and extraction strategies.

Figure 4.9 reveals that the strong discriminating power of the *SVM* classifiers lead to a quick convergence of the test error to zero. The cases "ALL", "3D-FEAT", and "2D-FEAT" represent configurations containing all available classifiers. Cases "SVM-ONLY" and "DT-ONLY" mean configurations using only *SVM* and *DTree* classifiers respectively. It can be seen that the test error converges much slower when a *weak* classifier (meaning low descriptive power) is used. This also means that only few *base classifiers* contribute to the overall classification results of the configurations using *SVMs*. However, since the final classification rates are high, this seems not to be disadvantageous.

Summarizing the results of this evaluation, the presented approach can be

considered successful in classifying the assembled dataset. It works better than the compared approaches and the positive effect of combining 2D and 3D features can be shown. Possibly, this becomes important when applied to even larger datasets. However, it is shown that the algorithm automatically chooses the approximately best combination of classifiers and features for describing the classification problem. For the purpose of enriching the knowledge base represented in the robot's *situation model* the developed approach seems feasible, especially, since the disambiguation system described in Chapter 5 only requires a rough visual interpretation of the perceived scene.

4.3. Room Categorization

Following the arguments of Torralba (2003), it is important for a mobile robot companion to consider the context in order to ground specific objects in a scene. One of the aspects of a situation's context is certainly the type of room that encloses the current situation. By considering this in the grounding process, certain object categories can be expected to likely appear in the situation, others can be ignored. Apart from that, this knowledge is beneficial for referential communication. Not only furniture and other objects are referenced in typical domestic interaction, also areas and rooms play a central role when aiming at a general purpose domestic service robot that is able to naturally interact with humans.

Swadzba and Wachsmuth (2011) approach the classification of room types by analyzing the three-dimensional structure of a room using a custom *Scene Plane Descriptor (SPD)* in conjunction with the 2D *gist* feature (Oliva and Torralba, 2001). They thereby analyze the scene visible in the current field of view and apply a subsequent voting scheme for merging the results from multiple frame-by-frame analyses for receiving a coherent complete room categorization.

The approach presented in the following sections employs a more holistic strategy for classification. It builds up on the idea of storing egocentrically perceived data in an allocentric representation in order to being able to incorporate the information on a spatial basis in the decision making processes. Before using a classifier for determining the type of a room, the robot gathers the relevant information in terms of visual features. This clearly demonstrates the progression of this thesis in comparison to the work of Swadzba (2011) from a constrained consideration of the *vista space* to a more comprehensive view on the *environmental space* (see Section 1.2). Once the robot has gathered information about the whole room, it is able to incorporate not only the currently visible features in the classification, but also peripheral information that were perceived from a previous view.

Since the goal of the holistic classification is the general categorization of whole unseen rooms, a large amount of data is needed in order to train a suitable classifier. The assembling of a database containing a reasonable amount of training data of six types of rooms will be described in the following section. In the subsequent section, the realization of an allocentric representation for the egocentrically gathered features will be presented.

Therefore the feature descriptors will be anchored spatially in a global coordinate system. Finally, the approach will be evaluated in comparison to the voting based approach from Swadzba and Wachsmuth (2011). The creation of the database, the implementation of the holistic classification approach and the evaluation was performed by Tobias Röhlig in a supervision relationship as part of his master's thesis (Röhlig, 2013).

4.3.1. Generation of Training Data

In order to train a classifier that is able to cope with whatever room or apartment a robot enters, the classification system needs a representative dataset that includes many samples of all types of rooms that should be recognized. Since no adequate database was found in literature for public access, and it is quite time consuming to record suitable data, the number of categories was limited to six: bedroom, dining room, living room, bathroom, office, and kitchen. The data was recorded in the homes of colleagues, friends and other volunteers using a ASUS Xtion Pro sensor and a laptop.

As mentioned above, the holistic classification relies on a spatial anchoring of the used features. On a mobile robot this is not problematic, because the localization ability of the robotic system can be used to transform the egocentric information from the visually perceived data to a global coordinate frame. When recording the room type dataset no localization of the sensor is available and a *point cloud* from a 3D sensor usually comes in camera coordinates. Hence, after having received the raw data in form of correlated frames of depth and RGB images, the next step must be to generate a 3D reconstruction of the scenes. For this the real-time *point cloud* reconstruction approach called *Kinect Fusion (KinFu)* developed by Izadi et al. (2011) was used. They developed a GPU pipeline for processing the depth images that consists of four main stages:

1. Convert depth image to 3D *point clouds* and normals.
2. Camera Tracking - Using a GPU implementation of the *ICP* algorithm, a 6-DOF transformation of the camera position is computed.
3. Integration of the new data into the scene using a volumetric surface representation (Curless and Levoy, 1996). The transformed and oriented points are used to update a single 3D voxelgrid.

4. The volume is ray-cast to extract a view of the implicit surface representation for the user.

Izadi et al. (2011) do not provide a solution to the loop-closure problem. This issue appears when having a full 360° scan. Since the reconstruction is not perfect, there are still variations from the correct transformations. This leads to displacements when reaching the starting point again after a 360° scan. In practice, this problem can be overcome by simply not recording full 360° scans. For classification the scans do not need to be complete but should capture the essence of a room's structure.

Due to the limited processing power of the used laptop the *KinFu* algorithm is not applied online to the recorded data, but in a subsequent post-processing step in order to make use of the full 30 fps scan frequency of the sensor. This way, the algorithm works more stably and produces better results in terms of completeness and accuracy. The intermediate camera positions received from the tracking step are used for transformation of the egocentrically perceived visual features to a global coordinate system. Some of the features for the categorization process are calculated from the complete *point cloud* of the current scene. In the frame-by-frame case this is just the complete scanned cloud. In the holistic approach it would be ideal to use a *point cloud* of the whole room. However, *KinFu* only provides a smoothed mesh as representation of the fused scene. So the 6D camera positions are used again to merge the *point clouds* from each frame to a coherent global non-smoothed *point cloud* of the complete room. Finally, a uniform sampling is applied to the resulting cloud in order to remove redundant data. An example is depicted in Figure 4.10.

The database contains 114 scanned rooms in total. Each sample is represented by the raw sequence of RGB and depth images, the 6D camera transformation for every frame, and the reconstructed complete *point cloud* of the whole room.

4.3.2. Anchoring of Features

The spatial analysis of the scene is not only relevant for the reconstruction in order to provide continuous data for global feature calculation. As stated already, all egocentrically perceived visual features — local and global — are anchored in an allocentric representation of the robot's surrounding. The

Figure 4.10.: A reconstruction of a living room scan using KinFu.

anchoring is relevant for eliminating redundant information which emerges through overlap of the egocentrically perceived data from which the visual features are extracted.

In the prediction phase the anchoring of features becomes particularly relevant. When using the categorization approach to the classification of a room in a real-world scenario, the robot needs to make sure that only the features received in the room in question are fed to the algorithm. Features from neighboring rooms need to be ignored. The anchoring approach also allows other applications to analyze the features with respect to their position, e.g. for the detection of objects or functional sub-areas of a room.

As discussed in the previous sections, the goal is to use various types of features which incorporate different ways of extraction. The 2D features need a gray scale or color image. 3D local features from *point clouds* can also be computed frame-by-frame or on the reconstructed cloud. When using the software on a mobile robot to perform online computation of features, the per-frame option prevents from having to cope with updates to the global reconstructed *point cloud*. It is usually also more accurate because small inaccuracies in the reconstruction of the scene result in possibly severely

altered descriptions of keypoints. Global features on the other hand, like the custom *SPD* descriptor, need the reconstructed scene as foundation for computation.

The local features are not computed for each frame because this would produce a large amount of redundant data and unnecessary computational load both in the detection and the training/prediction phase. In practice, using each 50th frame turned out to be a good compromise between reducing redundancy and capturing enough descriptive evidence. Each local feature contains its anchor position in the allocentric 3D scene representation. In the case of 3D features the corresponding 3D keypoint is used as an anchor. In the case of 2D features, the keypoint is projected to the corresponding depth image, which provides the 3D anchoring position. These pairs of descriptors and anchors are stored in the sparse allocentric representation. In order to again reduce redundancy in the data, each feature is assigned its corresponding codeword from the global codebook. This is used to eliminate duplicate data which is detected by small distances in the feature space and the geometric space. To integrate the global *SPD* descriptor into the same representation the individual detected planes from which the feature is calculated are anchored in the allocentric feature-anchor space via their centroid.

Eventually, this approach generates a large database of visual words representing the analyzed scene. For each type of feature the classification system creates a *BoW* which contains the distribution of visual words corresponding to the processed feature type for the complete room. In the case of the evaluation described in the next section all gathered features can be used for this because only enclosed rooms were scanned. In a real-world situation a robot would need to establish a new feature database whenever it enters a new room in order to replicate this strategy. Alternatively, it can apply spatial reasoning using the anchors in order to generate a reasonable subset of features in its surrounding for classification. This is further discussed in Section 4.4. Finally these *BoWs* are used for categorizing the represented room using the *E-SAMME* approach described in Section 4.2.

4.3.3. Evaluation

In this section the applicability of the *E-SAMME* algorithm to room categorization tasks will be evaluated. The general capabilities of the boosting approach will be further investigated in the context of a different task. While the previous evaluation in Section 4.2.2 focused on instance recognition, here the approach is deployed to a category recognition task. In this context, the usability of the frame-by-frame computation of local and 2D features is evaluated, as well. Furthermore, *E-SAMME* will be trained and evaluated on the IKEA dataset established by Swadzba and Wachsmuth (2011) to be able to compare this new approach for room type classification to their approach.

Like in the previous evaluation, the test error was recorded during each boosting step. The number of boosting steps was set to 100 to show that the test error converges after an appropriate amount of base classifiers in the additive model and a $k = 5$ fold cross-validation was applied. The set of classifiers was composed of different settings of *SVMs* ($\times$200), *DTrees* ($\times$200) and *MLPs* ($\times$10). A sampling method for weighting the training data, the *maximum selection* with a sample factor of $z_f = 0.6$ was used. The 114 samples from the training set have been distributed across the six room categories as follows: living room ($\times$23), office ($\times$13), bedroom ($\times$19), kitchen ($\times$25), dining room ($\times$9), bathroom ($\times$25).

Generalization Capabilities for Category Recognition

The most salient measurement for a classifier is the generalization error. For categorization tasks this is particularly crucial, because the classifier needs to deal with high in-class variation in order to correctly categorize previously unseen rooms. This experiment aims to show that the use of different features and classifiers for boosting leads to good generalization abilities on the generated dataset. Simultaneously, the descriptiveness of the local 3D features when derived from a frame-by-frame analysis compared to the extraction from the reconstructed *point cloud* are tested.

First, the overall performance of different combinations of feature extraction strategies and feature types are tested. Therefore four different cases are defined:

KF-3D
: Only 3D features extracted from the *KinFu* reconstructed *point cloud*.

FBF-3D
Only 3D features extracted from frame-by-frame *point clouds*.

FBF-ALL
All feature types; local features being extracted frame-by-frame.

FBF-GS
SPD features from reconstructed cloud; *gist* feature frame-by-frame.

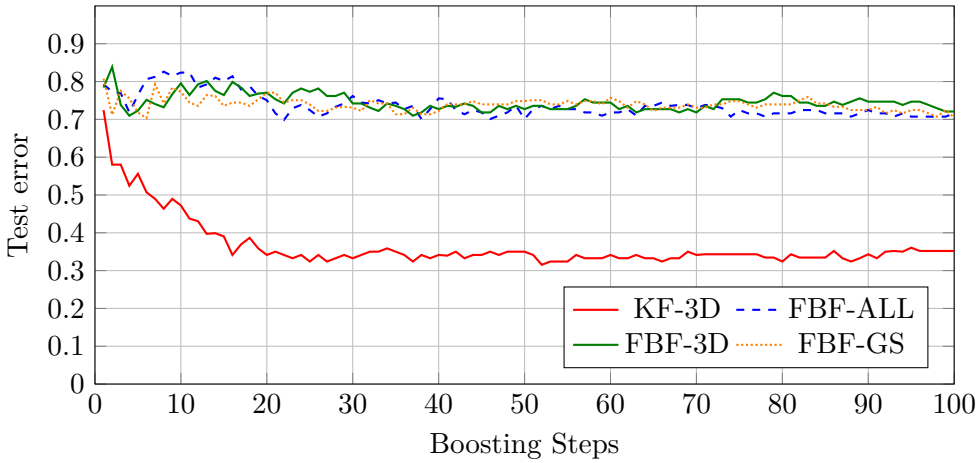


Figure 4.11.: Mean test error over 5 training runs using the k fold cross validation. The test error progress over 100 boosting steps is plotted for four combinations of feature types and extraction strategies.

Figure 4.11 shows the mean test error progress for the *E-SAMME* algorithm over $k = 5$ training runs using the k fold cross-validation. While the test error of frame-by-frame computed features does not improve significantly during the training, the trials with 3D features generated from complete room clouds do show decreasing test errors with increasing numbers of base classifiers in the boosting model. This improvement of the generalization error can be observed in up to approximately 20 boosting steps ($error = 0.34$). After 20 boosting steps, the error shows small oscillations but does not decrease significantly any more.

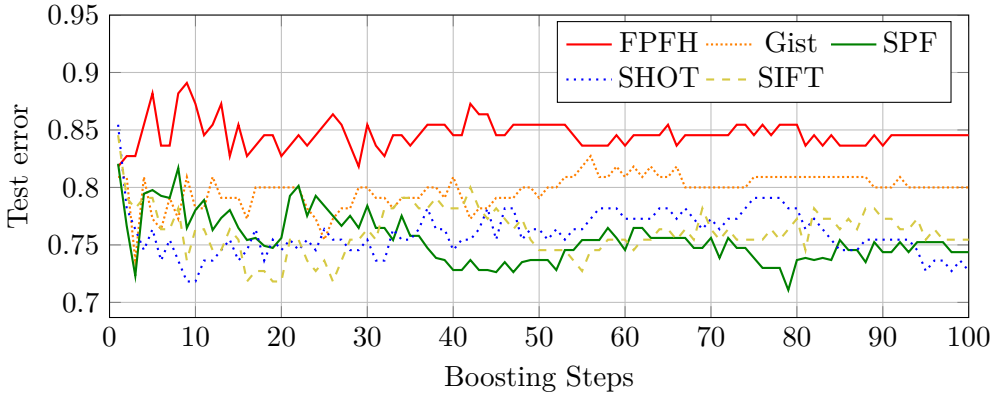
Obviously the training on frame-by-frame generated local features leads to worse generalization than when using merged *point clouds* for feature generation. The reason for this disparity does not seem to be the general superiority of 3D features for room categorization, because using only 3D features generated frame-by-frame does not lead to better results than the usage of all features. A probable explanation of this observation might be that artifacts of the feature computation and merging applied frame-by-frame cause this inferior performance.

Two reasons explaining this issue come to mind: First, each video for the database is recorded with different speeds of the camera movements. Thus, the number of frames for each record differs and hence, the number of features differs within the room categories. This leads to possible corruption in the global *BoW* histogram features created for each room. This case was thought to be avoided by spatial sampling of the feature positions.

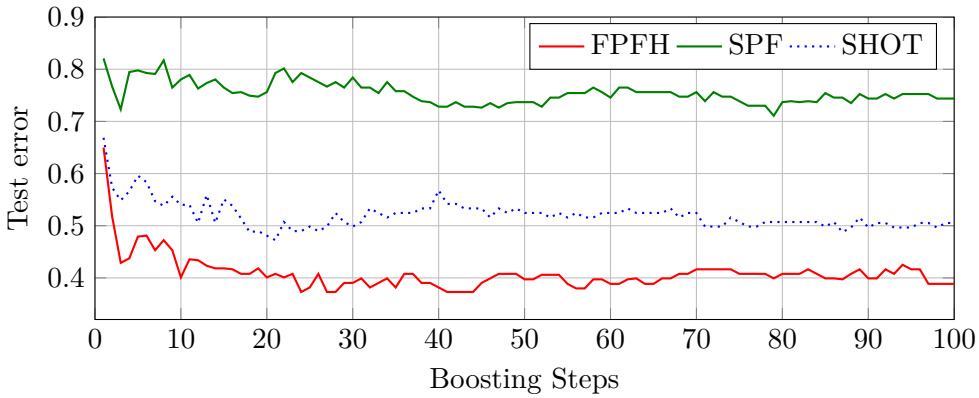
Exactly this spatial sampling of features might be the second explanation of this effect. The positions of features computed in two frames do not necessarily overlap exactly. On the one hand, if the radius for sub sampling is chosen too small or the camera transformation between the two frames is inaccurate, there are still too many local features of one type in the overlapping area which leads to corruption of the global histograms. On the other hand, if the radius is chosen too large, the features do not fully cover the structure of a room.

For a more detailed analysis of the contribution of the different individual features to these results, additional training runs using only one for the features were performed. The plots in Figure 4.12a and 4.12b show the test error progress for these runs, using frame-by-frame extraction and reconstructed *point cloud* extraction respectively. The errors using the frame-by-frame method show no significant improvement. Just as in the combined case (FBF-ALL) from the previous test the oscillation amplitude of the error decreases after approximately 40 training steps, but the generalization error improves only slightly. The test error for features generated from reconstructed room scans shows an improvement over the first 20 boosting steps for *FPFH* and *SHOT* features. However, the error for *SPD* features does not show any improvement. The curves for *SPD* in Figures 4.12a and 4.12b are equal since both underlie the training on plane features computed on the reconstructed clouds.

Despite the poor performance of 3D features in the frame-by-frame case,



(a) Boosting of frame-by-frame computed features



(b) Boosting of features from merged point clouds

Figure 4.12.: Mean test error over $k = 5$ cross-validation steps for different features in the frame-by-frame and reconstruction condition. Notice the different scales on the y-axes.

it can be obtained from the results that using different features combined in a boosting algorithm enhances the classification capabilities compared to single boosted features. Especially the results for the 3D features generated from merged *point clouds* of rooms show the positive influence of different feature types. While the individually tested features reach only an error rate

of 0.38 minimum (FPFH, see Figure 4.11), the usage of all three features leads to a minimal error rate of 0.32 (Figure 4.12b).

Figure 4.14 shows the average final categorization results represented in confusion matrices. It can be seen that the overall classification rate is quite high for most classes in many cases. Again, the poor performance when using features from frame-by-frame analysis is obvious (see Figure 4.14d). The small amount of samples for some categories also seem to decrease the generalization capabilities. Categories with a relatively small amount of samples (dining room and office) have the worst generalization results. When omitting one or both of these categories the overall results gets significantly better (Figures 4.14e and 4.14f).

When comparing the descriptive power of each type of feature, it can be observed that using the frame-by-frame approach, *SHOT* features are less susceptible to the negative effects of frame-by-frame detection than *FPFH* features (already seen in Figure 4.12). In the case of using reconstructed *point clouds*, however, *FPFH* features are better in terms of generalization error (minimal test error: *FPFH*: 0.38, *SHOT*: 0.49). This can also be obtained from Figure 4.13 which displays the mean number of used feature types per trial. The mean is again computed over the five cross-validation steps. It can be seen that the amount of *SHOT* features is higher for the frame-by-frame method, whereas for the merged point clouds, the *FPFH* feature is used more often.

Usability of Base Classifiers

Another purpose of this evaluation is to show the usability of different base classifiers in combination with boosting and different feature types. Fig-

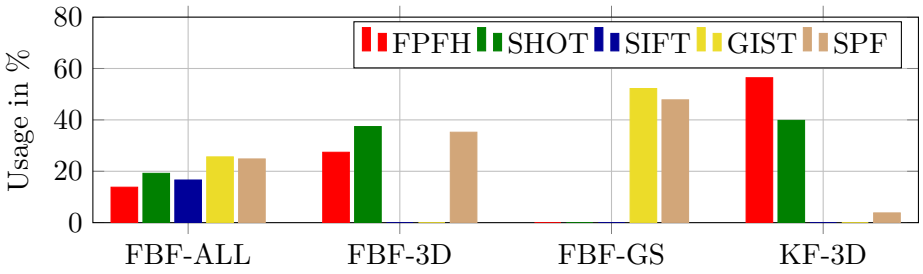
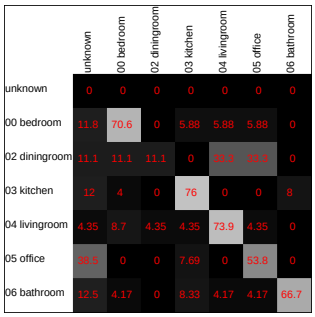
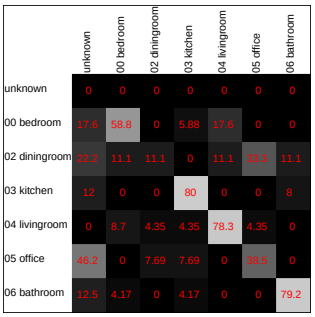


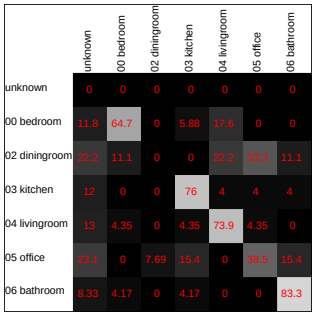
Figure 4.13.: Average amount of features used in a trained E-SAMME.



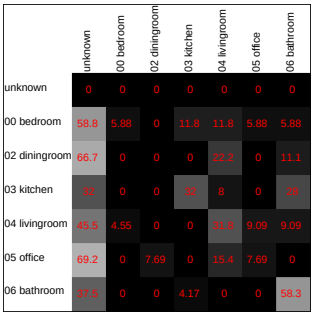
(a) MLP trained on 3D features from reconstructed clouds



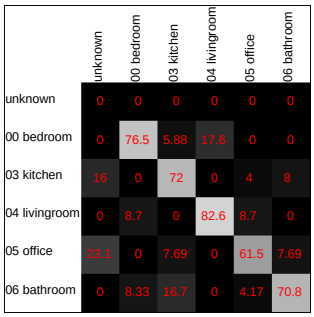
(b) DTree and MLP trained on 3D features from reconstructed clouds



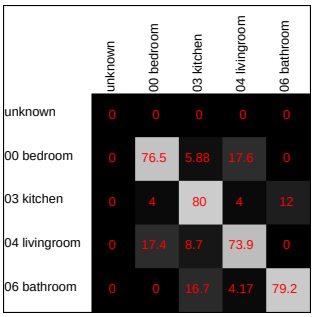
(c) All classifiers trained on 3D features from reconstructed clouds



(d) All classifiers trained on frame-by-frame computed features



(e) All classifiers trained on 3D features from reconstructed clouds, 5 classes

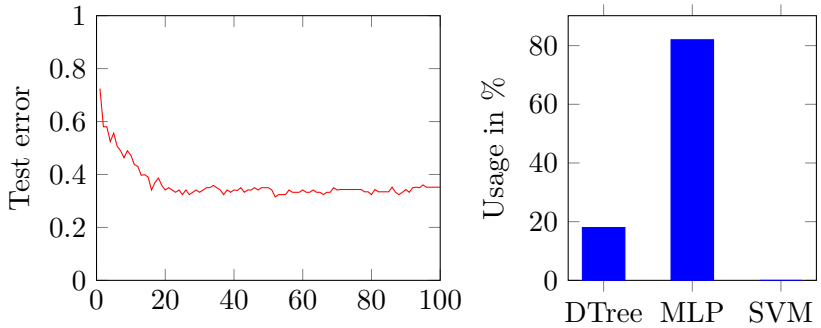


(f) All classifiers trained on 3D features from reconstructed clouds, 4 classes

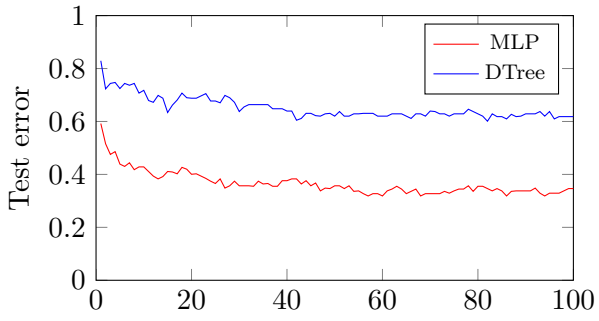
Figure 4.14.: Confusion matrices from the evaluation of E-SAMME. The rows represent the true labels of the candidates, the columns the predicted labels.

ure 4.15 depicts the number of used base classifier types and the corresponding test error. For this test the 3D features *SHOT*, *FPFH* and *SPD* computed on the reconstructed *point clouds* were used. Each type of classifier appears in different settings. The *SVMs* with RBF kernel vary in their parameters C and γ. *MLPs* are represented with one and two hidden layers. The number of neurons per layer varies between 50 and 300. The *DTrees* vary in the minimal number of sample points necessary to split a node and in the settings for accuracy and tree building settings. The plots in Figure 4.15 show the frequency of each type of classifier as a collection of all its settings.

Figure 4.15a shows the combination of all classifier types. Whereas the *MLPs* are used most often, *DTrees* are only sometimes used and *SVMs* not



(a) All types classifiers with 3D features only



(b) Individual results for MLP and DTree

Figure 4.15.: The usage of base classifiers and the corresponding test error.

at all. This might correlate to suboptimal choices of parameters. The usage of *MLPs* only shows a similar final error rate compared to the combined trial, even though the error oscillates more and converges more slowly (Figure 4.15b). The classification error is higher for the *DTrees* which is not surprising since it was also chosen less often by the boosting algorithm. But both classifiers seem to contribute to the overall performance of the resulting combined classifier which confirms the assumption that the combination of different *base classifiers* is beneficial.

Furthermore, the effect of differently configured classifiers might influence the classification results. This is the other advantage of using *E-SAMME* for automatic combination of classifiers. They may not only differ in the used algorithm, but also in their choices for parameters. Figure 4.16 shows the test error progress for boosted uniform *MLP* settings and their combination. In contrast to uniform boosted settings of *MLPs*, the combined usage converges much faster (no relevant changes after 25 steps). However, some of the other configurations finally reach a comparably low error value.

Although no significant superiority of the combination of different types

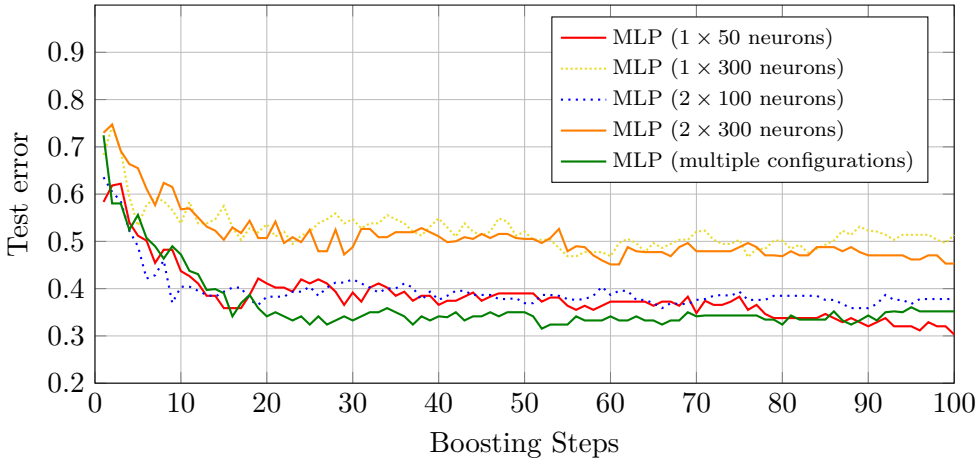


Figure 4.16.: Comparison of the test error progress for a boosted set of classifier configurations and boosting of single classifier configurations. Shown are the average test error progress over $k = 5$ cross validation steps for various MLP settings.

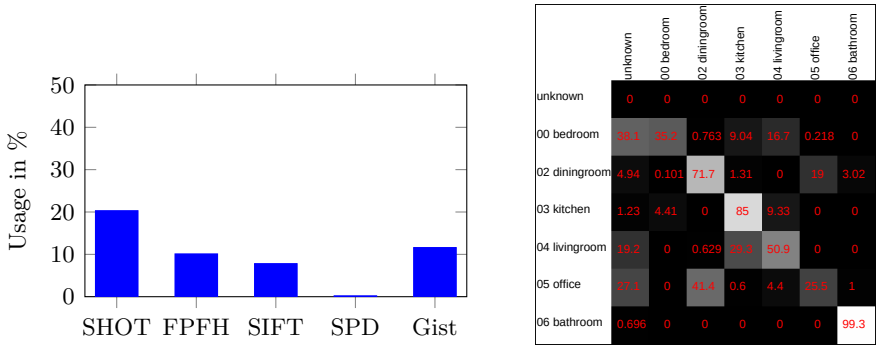
of classifiers and different configurations of those can be proved (except faster convergence), the usefulness of *E-SAMME* is obvious. Since it does not perform worse than the best competitor either, it can be stated that its capability of adapting to the current problem is conspicuous. This dispenses the need for manual determination of the appropriate classifier, its optimal parameters and the used visual features. It can also be obtained from the results that a suboptimal choice of only one of these alternatives may lead to a significantly worse categorization quality. Whereas the usage of *E-SAMME* guarantees the ideal configuration of the available tools.

Comparison to a Voting Based Approach

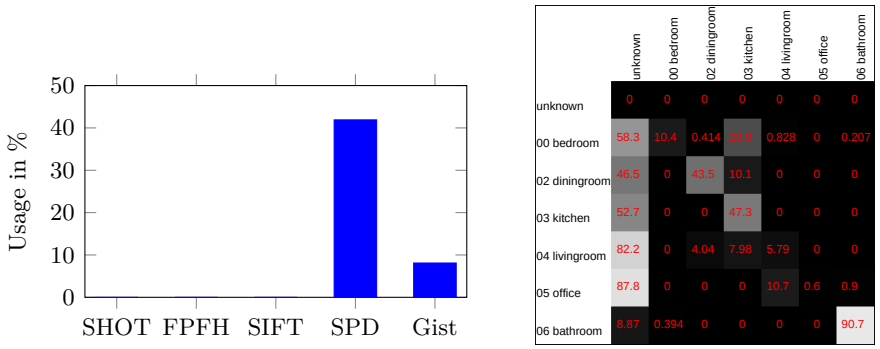
This section analyzes the quality of *E-SAMME* compared to the voting based classification approach by Swadzba and Wachsmuth (2011). For comparability the tests were performed on the IKEA dataset which is publicly available. The features for this experiment were computed frame-by-frame and have not been merged so that each frame contains one global feature for each feature type. Since the scan of one scene was not merged and considered as one sample, each frame of the recorded scans was used as a single sample for training or candidate for testing respectively. A 10 fold cross-validation was performed with one randomly chosen frame set as test candidate per room type.

E-SAMME was trained in three different configurations: using *MLPs* with all available features, using *MLPs* with *gist* and *SPD* features, and using *MLPs* only with 3D features. Figure 4.17 shows the confusion matrices for all three cases and the corresponding histograms of chosen features types.

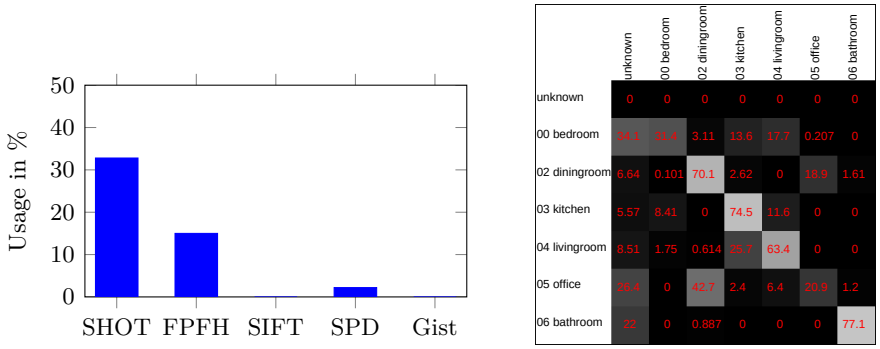
One can extract from the confusion matrices in Figure 4.17a that classification of the bathroom and kitchen works well (99.3% and 85%) when using all feature types. However, the categorization of bedroom and office was unsuccessful. The same tendencies as on the other dataset evaluated in previous sections are visible here. In general, the results are not as good as in the voting based approach proposed by Swadzba and Wachsmuth (2011). Using *E-SAMME* an overall classification rate of **60.83%** (SD: 29.0%) is reached, while the voting based approach reaches **78.0%** (SD: 24.6%). When comparing to the case when only using *gist* and *SPD* features the generalization quality of *E-SAMME* is even worse (Figure 4.17b). When using only 3D features (see Figure 4.17c) the categorization results are acceptable.



(a) Boosting of MLPs with all features on the IKEA dataset.



(b) Boosting of MLPs with *gist* and SPD on the IKEA dataset.



(c) Boosting of MLPs with only 3D features on the IKEA dataset.

Figure 4.17.: Confusion matrices of the training and classification on the IKEA dataset. The rows represent the true labels, columns the predicted results.

It can also be seen that the *SPD* feature is not used very often when local 3D features are available. Thus, it becomes clear that *SHOT* and *FPFH* features describe the 3D scene more accurately than *SPD* features. This is probably due to the fact that the local features do not only describe the planes in the scene but also edges, corners and curved surfaces.

4.4. Summary

In the previous sections solutions for furniture detection and recognition, general object identification and room type categorization have been described. These approaches to assigning semantic labels to the geometric structures in the robot's environment provide necessary information for establishing a useful *situation model*. As seen in the introduction to this chapter, the knowledge about the identities and functional roles of the objects and areas within an apartment is crucial in referential communication. The information from these visual interpretation components can be included in the allocentric representations in the robot's *situation model*. Identified furniture and manipulable objects can be represented in the *allocentric instance representation* described in Section 2.3. The room type is probably best represented in an *allocentric areal representation*. The anchoring process for visual features described in Section 4.3.2 profits from the central representation of situation information. The features can be included in the allocentric representation using the previously proposed mechanism that allows to link-in arbitrary structures to the default representation (c.f. Section 2.3). This way, the information is available to other components within the system.

The topics discussed in this chapter relate to various research questions established in Section 1.3. The considerations about the relevance of semantic knowledge about certain structures in the environment contribute to Question 1. Especially the approach to room type categorization is related to Questions 3 (spatial and temporal integration) and 4 (inclusion of context). The anchoring of features realizes an integration of visual information spatially and over a certain period of time. Also it allows to include the context (in terms of peripheral visual information) in the room categorization task, and also allows to apply the functional role of an area as context information to other interpretation tasks.

In general, all presented approaches to visual interpretation could successfully be applied to their corresponding tasks. The furniture segmentation and recognition approach using 3D *ISM* shows positive results in its generalization capabilities. However, its applicability to detection tasks in cluttered scenes has potential for improvement. According ideas for refinement are discussed in the corresponding sections.

The household object classification approach reaches high success rates in identifying objects in the assembled dataset – also in comparison to other state-of-the-art approaches. It combines different classifiers and feature descriptors in order to find the best combination for distinguishing the target objects.

The categorization of rooms works well on the recorded dataset, albeit the results on the IKEA dataset do not reach the level of accuracy that is reported by Swadzba and Wachsmuth (2011). Here, the spatial anchoring and the set of *base classifiers* and visual features should be revised. However, the concept of spatial anchoring of visual features for a holistic interpretation of geometric structures still seems promising. For the training of the category models this is currently not exploited explicitly, although classifiers that make use of this information are imaginable. The 3D *ISM* approach from Section 4.1 is an example for that.

The anchoring mechanism cannot only be used for recognizing the current room a mobile robot is situated in. It is also imaginable to build up a more comprehensive database of anchored visual features which allows more sophisticated interpretations. An approach to reasoning about the spatial distribution of features within the environment could be able to detect functional areas in the apartment that are not limited to complete rooms. For example when a home does not have this distinct separation of rooms, but includes a kitchen, dining room and living room area in one large non-separated room, the different functional areas could still be detected. Reasonable subsets of the detected features based on their positions could be passed to the classification component in order to identify corresponding areas. This could also enhance search tasks by providing first hints about the presence of target objects in certain locations.

However, this chapter shows that visual interpretation of the robot's environment is far from being solved — also when considering other publications. Integration of visual context into the interpretation processes is certainly a necessary step for improving the robot's capabilities in this field. But we

will not be able to avoid taking other modalities into account when aiming for a universal *situation model* that is able to support a natural interaction with human interlocutors. Therefore the following chapter investigates ways for harnessing the auditory modality for grounding utterances about spatial structures in the environment and rectifying the results from visual interpretation.

Chapter 5

Perception and Communication - A Mutual Benefit

The previous chapters focused mainly on visual perception in order to build up a representation of the environment as a basis for establishing a *situation model*. But the *situation awareness* of an agent also largely depends on the comprehension of speech and the information that can be extracted from the utterances of the other persons in the situation. With the goal of building a multi-purpose service robot in mind the need for a reliable understanding of what is being said is obvious. For the future, much more complex scenarios such as the general purpose assistant for the home are envisioned. However, these will depend on easy operation and a high level of system-human integration (Engelhardt and Edwards, 1992). In order to being accepted by humans as a social companion the robot must be able to comprehend natural language, not only a small predefined set of commands. Further, following the findings of Brennan and Clark (1996) that the conversational content depends on the interlocutors' implicit consensus, it becomes clear that dialog and *situation model* require a proper coordination. This is true not only for human interlocutors but also for artificial ones in order to being able to perform successful multimodal *HRI*. As a consequence, this means that the *situation model* depends on the context of the current situation including the conversation, while the correct interpretation of the communicational situation in turn depends on a sophisticated *situation model* which provides background knowledge. In either case a reliable grounding of the conversational content is required.

Related Work

Coordination of speech with other modalities is a reoccurring topic in research on *HRI*. This includes the production of multimodal cues in order to enrich the communication and to make it more natural for the human interaction partner (e.g. Li and Wrede, 2007). The communicative robot HERMES (Bischoff and Graefe, 2002) serves as a guide and entertainer in museums and fairs. It is able to act appropriately in various situations by switching the conversation context based on the utterances of the visitors. Using gestures and gaze alongside speech, the robot produces rich communicative signals for example to explain directions to the interlocutors. However, for perceiving information from the users it only relies on the auditory modality and does not evaluate non-verbal cues. But the visual domain is in many situations an important source of information for an interaction. There have been various attempts to design robots that interact with the human using the visual domain, e.g. for "programming by demonstration" (Erlhagen et al., 2006). One of the problems here is that the usage of the single domain limits the possibilities of the human for teaching the robot new behaviors apart from very simple ones. For more complex tasks the monomodality of the process inhibits a static communication flow and prevents a dialog between human and robot.

Breazeal (2004) argues that future *HRI* needs to work like human-style social exchange. This obviously includes multimodal interaction in a natural way. It supports a more competent and enjoyable collaboration while working together and enables the robot to engage in various forms of social learning ("socially guided machine learning") without any need for additional training because humans are already experts in social interaction.

Leonardo is a social humanoid robot (see Figure 5.1) that can work alongside people as cooperative teammate (Breazeal et al., 2004). It understands and produces communicative signals through dialog, gestures, and facial expressions which enable it to develop natural human social skills. The supported interactions are limited to task-oriented goals, but include the interpretation of simple referencing of objects in the near vicinity, the comprehension of spatial relations and perspective change.

As already discussed, the grounding of verbally described entities and concepts in the visual perception is important for referential communication. There are several systems that use non-verbal cues like gestures and gaze

Figure 5.1.: The social robot Leonardo. Image taken from Breazeal et al. (2004).

for this grounding process (e.g. Stiefelhagen et al., 2007) or combine these cues with bottom-up visual attention mechanisms (Schmidt et al., 2008) or world knowledge for grounding observed events in the scene (Dominey et al., 2004).

Rickert et al. (2007) demonstrate an integrated dialog system for human-robot collaboration tasks. It contains a semantics module that is responsible for selecting the most likely semantic interpretation of the user's request based on the knowledge about the world model as well as visual and other multimodal input channels. This is used to identify referred to objects in a collaborative construction task of assembling "Baufix" toys. For responding in the performance of the dialog it uses motor abilities to produce non-verbal behaviors like lip movement, gaze, gestures and facial expressions in order to promote the content of the speech.

In referential communication ambiguities in the interpretation of the referent or the involved objects in a spatial reference description occur on a regular basis. This problem is covered by several systems in literature. Perzanowski et al. (2001) present a multimodal *HRI* interface that uses gesture for disambiguation in the referencing of objects in a scene. In the work of Schauerte and Fink (2010) vision and speech are combined in order to establish a joint attention in multimodal *HRI*. They assume that sharing a common point of reference with an interaction partner is fundamental for learning, language and sophisticated social competencies (Mundy and

Newell, 2007). Further they argue that the conversational domain is most important for identifying this referent. However, especially when considering object relations, the perception influences the interpretation of referring acts, because "listeners [...] identify objects on the basis of ambiguous references by choosing the object that was perceptually most salient" (Beun and Cremers, 1998; Clark et al., 1983). Their system combines pointing gestures and information about the appearance of an object from speech in order to establish a biologically-inspired saliency model. This is used to support the visual search for the referent in the scene.

Gorniak and Roy (2005) try to resolve ambiguities in the linguistic form and the conversational content. Therefore they employ probabilistic representations of multiple hypotheses for lexical and grammatical choices. Using a situation model that contains agents and objects nearby as well as the speaker's intentions, their system comprises a probabilistic grounding approach using confusion networks for reference resolution.

Iwahashi (2003) describe a probabilistic framework that is able to acquire language by grounding speech to visual and behavioral information observed by a perceptual system. Their approach utilizes a consistent statistical optimization scheme in order to learn the linguistic knowledge in an unsupervised way. First, the lexical items for the concepts regarding single objects and concepts regarding motion are learned by presenting comprising images in which a person presents or moves an object with according utterances describing the object or motion respectively. After that, the system learns the grammar from more complex scenes and descriptions to establish relations.

Moratz et al. (2003) demonstrate how to use spatial knowledge in a communicative task between a mobile robot and a human. They focus on the fact that both interaction partners have different reference and perceptual systems. Using a semantic network formalism that represents the system's spatial knowledge, they try to disambiguate projective relations in movement commands for the robot. They found in their experiments that humans mostly take the robot's perspective. Therefore, they have equipped their robot with an egocentered reference frame by partitioning the environment along a reference direction into left-right and front-back. This reference direction is defined through a vector from the robot's center of mass to a relatum. A relatum could be the centroid of all perceived objects or a salient object.

Similarly to these systems, this chapter will focus on the disambiguation problem of grounding speech in the perception. Therefore, a probabilistic network model will be presented that handles multiple hypotheses for grounding a certain description. Over time this enables the system to identify the correct choice of hypotheses which contributes to Research Question 3 (see Section 1.3). Here the problem of temporal integration of consecutive information into a consistent knowledge representation is addressed. The model's topology is based on the geometric layout that is represented in the *situation model*. In this sense it makes use of the references deployed in the model and establishes new spatial relations for future reference, e.g. for speech production. This inclusion of spatial context information into the grounding process and vice versa contributes to Research Question 4 (see Section 1.3). The previously gathered semantic knowledge about entities in the environment is used in the interpretation of speech and this information from the auditory modality is in turn used to enhance the *situation model*. This improves the conversational skills of our robot *BIRON* because the *situation model* can now profitably be deployed in a dialog in *HRI*.

As seen in the previous chapter, the reliable automatic visual recognition of indoor scenes with complex object constellations using only sensor data is a nontrivial problem. In order to improve the construction of an accurate semantic 3D model of an indoor scene, it is desirable to exploit human-produced verbal descriptions of the relative location of pairs of objects. This enables the system to coordinate the conversational content with the *situation model*. The grounding of these descriptions requires the ability to deal with different spatial *Reference Frames(RFs)* that humans use interchangeably. In German, both the intrinsic and relative *RF* are used frequently which often leads to ambiguities in referential communication (e.g. "The plant is in front of the chair"). I assume that there are certain regularities that help in specific contexts. These include the actual spatial arrangement of the objects involved and the perspective on the scene. Also the results of visual interpretation – which might also be ambiguous – are assumed to be helpful in the interpretation of the descriptions regarding *RFs*.

5.1. Benefits of Combining Perception and Communication

In *HRI* a large part of the interaction that involves a *situation model* is referential communication. On the one hand, this requires a reliable grounding of the uttered references to the visually perceived geometric and semantic structures surrounding the interlocutors. On the other hand, when proactively shaping the conversation, the speech production requires to transfer the internal representation of the referenced objects into a pronounceable description that is appropriate to the current situation. This involves taking into account which implicit rules and habits apply in the current dialog, depending on the interaction partner, the function of the conversation and the established common ground.

Today, in most multi-purpose robotic systems the grounding process handles speech recognition and visual perception separately. Most systems match spotted labels from the speech recognition to results of the object recognition module. This approach assumes that the robot can identify all relevant objects in the environment reliably and that the trained labels of the objects are the same ones that are used by any interlocutor. But as seen in the previous chapter, object recognition solutions are not yet good enough to satisfy these assumptions. And even if they were better already, a huge amount of training data for all possibly existing objects in a household would be needed.

In referential communication descriptions of the referenced objects or spatial relations can be ambiguous. This is often the case when the scene contains multiple instances of one mentioned category or when the speaker assumes the presence of context information. Especially when using spatial relations the different possibilities of selecting *Reference Frames* for the descriptions constitute ambiguity. Humans are often primed for using certain frames in specific situations, or through the implicit negotiation of a common ground. A canonical approach for resolving these ambiguities is often to apply certain heuristics considering perspective, the conversational context, or conversational repair mechanisms.

In robotics a closer cooperation between the dialog module and the perception module can generate benefit for both parties. In order to resolve ambiguities in the descriptions that need to be processed the grounding

mechanism should consider the visually perceived information. Even if the robot's perception is fragile, it might give a decisive hint for the correct interpretation. Multiple interpretations of a description can be tested against the visual information in order to verify or disprove hypotheses. This can also improve the alignment process to the individual preferences of an agent in the dialog context. By keeping multiple possibilities for describing a certain constellation, an according implementation can capture the preferred description as well as alternatives that might be used in different situations.

Accordingly, once a description is grounded in the perception, the results from the visual interpretation module can be improved by employing the labels that were used in the description for the corresponding entities. Thinking one step further, these new insights could be used to refine the object recognition models by adding images of the newly grounded object to the training set and rerun the training. This way the robot would constantly enhance its own abilities through implicit learning.

5.2. Reference Frame Selection in Human Communication

In ambiguous or uncertain situations, humans often communicate about entities in their immediate surroundings using spatial descriptions and — more precisely — use projective spatial relations like "A is in front of B" or "C is to the left of D." Although the interpretation of such spatial expressions seems to be straight forward to humans, artificial systems like robot companions suffer from a fragile perception of the visual scene and an inherent uncertainty in spatial descriptions. The former results from the robot's constrained sensory data and comparably limited object categorization capabilities as seen in the previous chapter. The latter is caused by the necessary discretization of space and the selection of the appropriate *Reference Frame* (*RF*). While there have been several computational models offering different solutions to the discretization problem (Cohn and Renz, 2007; Mukerjee, 1998), computational models employing a processing strategy considering selection of different *RFs* are still rare. Carlson (1999) defines spatial *Reference Frames* as coordinate systems that parse space and impose an orientation on the environment, people, or objects. According to Logan and Sadler (1996) and Miller and Johnson-Laird (1976) respectively,

RFs have a direction as well as an origin. Levinson (2003) distinguishes the relative, the intrinsic, and the absolute reference frame. The relative reference frame depends on an egocentric coordinate system, whereas the intrinsic reference frame is oriented according to the inherent axes of an object. The absolute reference frame is based on environmental features and corresponds to the allocentric representations described in the previous chapters. This frame will not be considered in this analysis. Figure 5.2 depicts an example for a scene that can be described in ambiguous ways using the relative and intrinsic *RFs*. For describing the plant in relation to the armchair using the relative *RF* one would say "the plant is located to the right of the chair". While the intrinsic *RF* considers the inherent orientation of the reference object and dictates a description like "the plant is behind the chair".

Figure 5.2.: Reference object (chair) with located object (plant): "to the right of" (relative RF) or "behind" (intrinsic RF).

Human *Reference Frame* selection and processing has been investigated intensively (e.g. Carlson-Radvansky and Irwin, 1993; Carlson, 1999). The objects' geometric and functional features are decisive for the assignment and orientation of the intrinsic *RF*. However, the influence of these features on the selection process has been neglected so far. Object features are

attributed to objects conceptually and comprise criteria such as canonical orientation and use (Levinson, 2003). The goal pursued in this chapter is to construct a computational model that uses verbal descriptions by humans and the knowledge about human use of reference systems to improve the semantic representation of a scanned scene containing a furniture arrangement. Therefore, the artificial system needs to have accurate empirical information about how humans use reference systems to describe different arrangements. In cooperation with Katrin Johannsen (Ziegler et al., 2012) objects from a domestic domain were used to elicit the frequency of occurrence of the relative and intrinsic *RF* for different pieces of furniture. The analysis distinguishes between *vehicle objects* (e.g., chair, see Figure 5.3a) and *opposite objects* (e.g., shelf, see Figure 5.3b) that reveal differences in the assignment of the intrinsic left/right axis according to their predominant use (Graf and Herrmann, 1989). While the intrinsic left/right axis of *vehicle objects* is primarily assigned in a way that corresponds to sitting *in* the object, for *opposite objects* it is assigned like standing in front of the object, facing it. However, there are objects that have no inherent orientation (e.g. plant, see Figure 5.3) and therefore cannot be categorized for being a *vehicle object* or *opposite object*. Others may be assigned an inherent orientation by the way they are used (e.g. dinner table vs. desk).

(a) A vehicle object (b) An opposite object

Figure 5.3.: Example for assignment of different left/right axes in intrinsic reference frames. Vehicle object (“left of the chair”) and opposite object (“on the right of the shelf”).

5.2.1. Conducting an Online Study

Spatial verbal descriptions were elicited in an online experiment conducted by Katrin Johannsen (Ziegler et al., 2012). 244 participants were shown pictures of object configurations and were instructed to define the spatial relations by inserting spatial terms in gapped sentences in the format

located object [to be filled in] *reference object*

For example, "The plant is ..... the chair". Pictures were created that consisted of a *reference object* and a *located object*. Different orientations of the *reference object* were obtained by rotating it clockwise by 90° angles on its vertical axis. Where 0° means that the inherent front of the object faces towards the observer, while 180° means that the observer sees the object's back. The *located object* (a plant or a stool) was placed at four different positions: relatively in front, to the left/right of, or behind the *reference object*. This procedure dissociated the relative and the intrinsic *RF* (Figure 5.2). Configurations in which both *RFs* resulted in the same description were excluded. The *located objects* were chosen deliberately from categories that have no inherent orientation so that an effect of this property on the selection could be excluded. Out of the 66 pictures, 36 contained vehicle objects (chair, armchair, and sofa) in four different rotations as *reference object*. 30 pictures showed opposite objects (shelf, wardrobe, and chest of drawers) in three different rotations. The 180° rotation was omitted because opposite objects are usually located at walls, so that they are never seen from behind.

The resulting descriptions of the participants were coded as using "relative *RF*", "intrinsic *RF*", "ambiguous" (using both reference frames at the same time) or "other" (using an illegal or wrong description). Taking into account rotation and position of the *located object*, frequencies of the use of each *RF* were tabulated (Figure 5.4).

5.2.2. Empirical Results

Two rotations (90° and 270°) were used for statistical analysis to ensure a constant dissociation of *RFs*. The descriptions of the participants were coded as using either the intrinsic or the relative *RF*. Descriptions that did not use either were excluded (1.37% of the data).

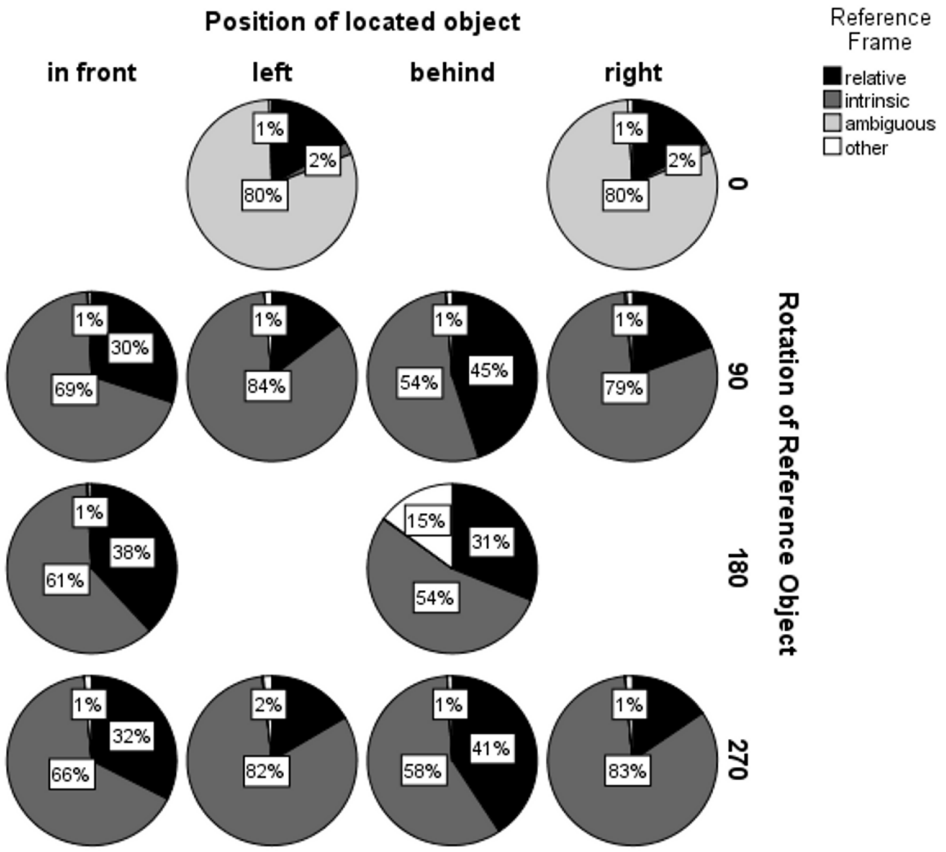


Figure 5.4.: Percentage use for each reference frame for a sample vehicle object.

Fitting a logistic mixed-effects model with full random slopes and intercepts and *RF* selection as dependent variable, model comparison reveals significant main effects of rotation and position of *located object* (both $p < 0.001$) but no main effect of object category. This means that the position of the *located object* and the rotation of the *reference object* obviously influence the selection of *RFs*, while it could not be proven that the categories of the involved objects have any effect on the selection process.

These results reveal that there are regularities in human choice of *RFs*. It

should be possible to exploit these in an artificial system for disambiguation of spatial descriptions for grounding in the system's perception.

5.3. A Probabilistic Model

The developed method for resolving ambiguities in referential communication uses the empirical results about the usage of *Reference Frames* and applies them profitably when interpreting a scene using visual and verbal information. This allows the system to ground possibly ambiguous utterances to the known structures in the environment. In turn it also enables it to adapt to the communication patterns used by the interlocutor. In the following, the data structures and algorithms will be described as part of a complete processing sequence including the perceptual interpretation. This includes a segmentation stage for identifying pieces of furniture and an initial classification stage. The component that represents the third stage converts the absolute representation from the visual analysis into a relative spatial-relation-based graph structure. It then matches verbal descriptions of pairs of items with entities in the graph.

5.3.1. Visual Analysis

For the visual analysis of the scene, the Point Cloud Library (Rusu and Cousins, 2011) is used in conjunction with a ASUS Xtion Pro depth camera. The previously described segmentation strategy from the *ASM* system (c.f. Section 3.2.3) cannot be applied here, because furniture usually does not move within the household. Therefore, their movable nature cannot be observed, which is the only evidence the *ASM* system exploits for segmentation. However, this approach can be used in order to concentrate on the static structures of a scene while ignoring movable and dynamic structures, which in most cases do not represent furniture items (chairs are an exception). Thus, the segmentation algorithms need to apply a heuristic for bottom-up segmentation or employ world knowledge about the indoor setting which the robot is deployed to. Usually the static structures in an apartment are either furniture or parts of the building's structure, namely floor, ceiling, and walls. So, in order to segment the furniture, the first step in the segmentation phase is to extract those bounding structures from the scene.

Figure 5.5.: Examples for result from the segmentation and classification of the furniture arrangement inside an apartment.

In order to do this, plane models are fitted into the *point cloud* using the RANSAC algorithm (Fischler and Bolles, 1981). The algorithm assumes that a found plane can be removed if no points were perceived behind (walls), below (floor), or above it (ceiling), because in these cases, the plane is most likely a bounding structure of the room. Admittedly this assumption does not hold for walls containing open doors or other passages. In this case the camera may perceive objects behind the wall, so here the algorithm needs to rely on a minimum size of the plane structure for classifying it as a wall.

Subsequently, the remaining points are clustered into coherent clouds. Clusters with a size above a certain threshold are assumed to be furniture. In a following verification step the clusters are checked for having contact with the floor or a wall. Occlusion may cause that one furniture item is decomposed into several parts in the *point cloud*. In order to reassemble those structures, the "flying" clusters (which have no contact to the floor or a wall) are merged with the nearest cluster below.

For the estimation of the identity of a found furniture cluster a modified

version of the *ISM* method described in Section 4.1 was used. In contrast to the default behavior described above, in our case it does not combine segmentation and classification in one step. It rather takes the individual already segmented clusters and checks for every class how many corresponding features vote into the center of the cluster. This corresponds to the implementation used in the evaluation of the voting scheme (Section 4.1.3). The *Hough Space* only contains one sphere in the center if the pre-segmented object, and the votes intersecting this sphere enable the categorization. Ultimately, it generates a probability distribution $p(f)$ over all known furniture classes for each cluster.

5.3.2. Maintaining and Updating the Spatial Network

This section describes the algorithm for matching the descriptions to a spatial representation of the scene while this representation, in turn, is also updated according to the descriptions. For this, the visually perceived representation of the scene is transfered to a probabilistic network structure with vertices describing furniture and edges describing spatial relations. The sequentially processed descriptions are then matched to edges and are scored according to the expected probability that the description actually means the currently processed spatial relation represented by the edge. The best match is chosen based on the calculated scores and is used to update the probabilities in the network structure. A history of multiple hypothesis for the matching of each description is stored in order to be able to revert a previous decision.

The Probabilistic Network Structure In order to match the verbal descriptions to the visual results, the 3D representation of the scene from the visual analysis is converted into a probabilistic network structure. This enables the approach to represent entities and their relations. See Figure 5.6 for a sample network graph corresponding to the scene in Figure 5.5. Each vertex $v \in V$ of graph $G = (V, E)$ represents a piece of furniture in the scene. It contains a probability distribution $p(f)$ for the furniture category, which initially corresponds to the classification results. Further, it contains a probability distribution $p(o)$ for the orientation of the object as discrete values for the location of the item's intrinsic front $O = \{$ `left`, `right`, `front`, `back` $\}$ relative to the observer's viewing direction. For example, the

label `back` means, that the intrinsic front of the furniture is at the end facing away from the observer. Through interpolation a rough assumption for a more fine-grained orientation can be obtained. Since the visual analysis does not provide any orientation, in the beginning these probabilities are uniformly distributed. For every category a few properties are defined as additional world knowledge. These contain information about the opposite or vehicle nature of the objects of this category and whether these have a distinct intrinsic front or not. Note that from the way a furniture item without obvious intrinsic front is deployed in a scene, a temporal intrinsic front may be assigned in the communication. This is why the probability distribution for this is present for all vertices.

The edges $e \in E$ in the network represent spatial relations of the vertices in the relative RF. Each edge contains a discrete probability distribution $p(r)$ for four possible spatial relations $R = \{$ `left_of`, `right_of`, `in_front_of`, `behind` $\}$. Each edge is bidirectional, but has a distinct probability distribution for both directions. Using these discrete values, the distribution can describe variance in the usage of certain descriptions of one spatial re-

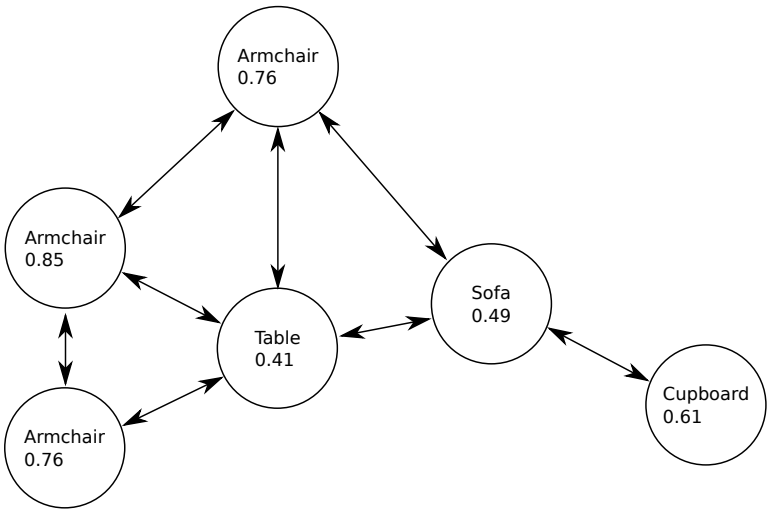


Figure 5.6.: An example for a furniture graph corresponding to the scene in Figure 5.5. Notice that the false classification of the shelf is transferred to the model.

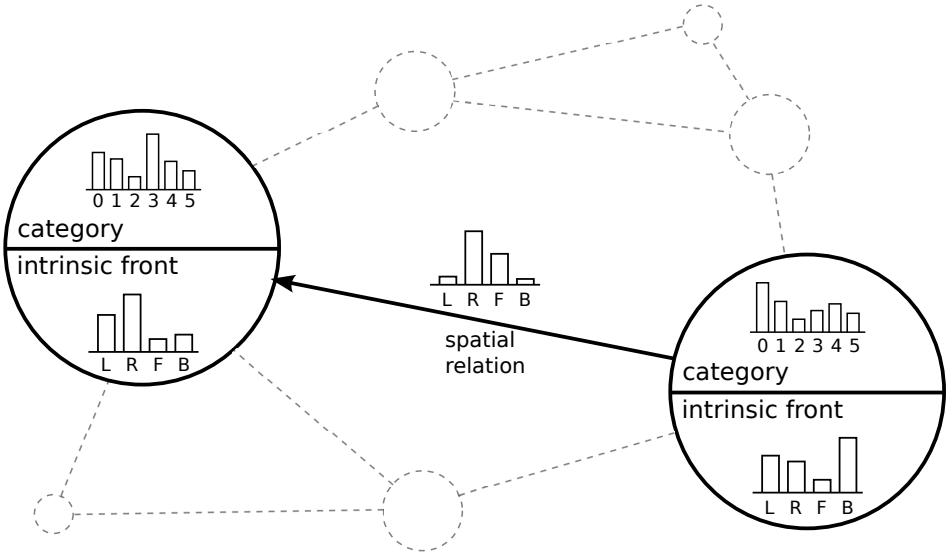


Figure 5.7.: Visualization of the information contained within the probabilistic model.

lation. It enables the system to represent if someone uses two descriptions interchangeably for the same spatial relation, maybe because the objects are located diagonal to each other. Another reason may be a great difference in size. If a small table is located in front, but at one end of a large couch, the description might also (depending on the purpose of the description) be related to the couch's center which results in a "next to" relation. The initial distribution of the believed spatial relation is calculated from the absolute coordinates of the point clusters, which are represented by the involved vertices. All location representations are defined to use the relative RF from a fixed viewpoint. An edge is established if the distance of two objects in the real world does not exceed a certain threshold.

Interpretation of Descriptions Now, verbal utterances containing spatial relations of two objects can be matched to this graph. Incoming descriptions are expected to contain a *located object*, a *spatial relation*, and a *reference object*. For example, the sentence "The table is in front of the shelf" can be matched, where "table" is the *located object*, "in front of" is the

spatial relation, and "shelf" is the *reference object*. The matching process requires that this description is converted into a partial graph as well. This description graph contains two vertices for the *located* and *reference objects* and a directed edge for the corresponding *spatial relation*. The description graph can now be matched to one of the edges in the furniture graph. For each possible edge e, a score value is calculated that represents the accuracy of its match to the incoming utterance using different interpretations.

Since the used *Reference Frame* γ is not known at this point, the description is interpreted as relative and intrinsic. The accuracy is calculated for both interpretations. For the latter, the system distinguishes between the opposite and vehicle nature and creates four interpretations for the four possible orientations of the *reference object*. This gives a total of five different scores for the matching of an description to one edge. See algorithm 5 for details.

Algorithm 5 Calculation of the Matching Score for all Edges

Require: f_{loc} (furniture category of the located object)
Require: f_{ref} (furniture category of the referenced object)
Require: r (spatial relation from the description)
Require: $I(r|\gamma[, o_{\text{ref}}])$ (interpretation of a spatial relation)
Require: $S(f_{\text{loc}}, f_{\text{ref}}, \rho, e)$ (score value calculation function)
Require: $O = \{o_i\}$ (orientations)
Require: $G = (V, E)$ (the graph containing furniture and spatial relations)

$s \leftarrow 0$
for all $e \in G$ **do**
  $s \leftarrow \max(s, S(f_{\text{loc}}, f_{\text{ref}}, I(r|\texttt{relative}), e))$
  for all $o_i \in O$ **do**
    if f_{ref} has opposite nature **then**
      $s \leftarrow \max(s, S(f_{\text{loc}}, f_{\text{ref}}, I(r|\texttt{intrinsic}_{\text{o}}, o_i), e))$
    else
      $s \leftarrow \max(s, S(f_{\text{loc}}, f_{\text{ref}}, I(r|\texttt{intrinsic}_{\text{v}}, o_i), e))$
    end if
  end for
end for
return s (best matching score)

The interpretation of the spatial relation given a *RF* and (in the case of the intrinsic frame) a intrinsic orientation o_{ref} of the *reference object* will be called ρ.

$$\rho \leftarrow I(r|\gamma[, o_{\text{ref}}]) \tag{5.1}$$

The score value for one interpretation of the spatial relation is calculated from the mean value of the probabilities of the three components of the description given the current edge. The probabilities of the objects' categories given the corresponding vertices are defined by $P(f_{\text{loc}}|e)$ for the *located object* and by $P(f_{\text{ref}}|e)$ for the *reference object* accordingly. Both can be read from the current graph. The third component of the score value is $P(\rho|e)$, the probability for the spatial relation's interpretation given the edge in the graph. The complete score value calculation function is then defined as:

$$S(f_{\text{loc}}, f_{\text{ref}}, \rho, e) = \frac{P(f_{\text{loc}}|e) + P(f_{\text{ref}}|e) + P(\rho|e)}{3} \tag{5.2}$$

with

$$P(\rho|e) = \max_{i,o} \left(P(r'|e) \times P(\gamma|f_{\text{ref}}, o_{\text{ref}}, r') \times P(o_{\text{ref}}|e) \right) \tag{5.3}$$

The probability for the spatial relation's interpretation $P(\rho|e)$ uses the probability $P(r'|e)$ of the relation r' in the current edge, which can again be received from the current graph. The relation r' is received from the interpretation of the description given by the *RF* and the orientation of the *reference object*. Additionally, the calculation considers the probability $P(\gamma|f_{\text{ref}}, o_{\text{ref}}, r')$ of the *RF* interpretation, which depends on the *reference object's* category (including the corresponding opposite/vehicle nature), its assumed orientation, and on the evaluated relative spatial relation. The probability distribution for this is a priori knowledge from the empirical study described above. Finally, the calculation also includes the probability $P(o_{\text{ref}}|e)$ of the chosen orientation given the previous evidences.

Finding the Best Match In order to find the best overall match of a interpretation of the description in the current knowledge base, algorithm 5 is applied to all edges of the graph. The calculated ratings provide a raking of pairs of interpretations and matched edges. This way the best interpretation of the description can be found given the current knowledge about the scene and knowledge about the human usage of *RFs*.

Update the Knowledge Base With this new information the knowledge base can be updated. For this, the probability distributions for the relation of the matching directed edge and the categories of the connected vertices

are adjusted. Optionally, if the interpretation uses an intrinsic *RF*, the orientation distribution of the target vertex is also adjusted. The update functions for the probabilities of the vertices and edges of the updated graph are defined as follows:

$$p_{t+1}(x) = p_t(x) + \frac{p_{\text{verbal}}(x) - p_t(x)}{\sigma} \tag{5.4}$$

Subsequently the distribution needs to be normalized again. Here $p_{\text{verbal}}(x)$ is the probabilistic representation of the interpretation of the incoming verbal description. $p_t(x)$ is the probability distribution of the entity to be updated at step t. The equation pushes $p_t(x)$ toward $p_{\text{verbal}}(x)$ with a damping factor represented by σ. The function is applied to all vertices and edges involved in the match, namely to $p(f)$ of the *located object*, $p(f)$ of the *reference object*, and $p(r)$ of the spatial relation. In these cases, $p_{\text{verbal}}(x)$ has a probability of 1.0 for the described category or spatial relation, 0.0 for all others.

In the case of an intrinsic interpretation, the function is also used to update the orientation of the *reference object* represented by the distribution $p(o)$. In this case the update target term $p_{\text{verbal}}(x)$ of the previous update function is

$$p_{\text{verbal}}(x) = P(o = x | r_{\text{verbal}}, r_{\text{visual}}, \gamma). \tag{5.5}$$

This is the probability distribution of the front direction which underlie the interpretation γ of the actual intrinsic *RF* that is expected to be in use. This can be calculated from the uttered spatial relation r_{verbal} and the visually perceived absolute relation r_{visual} between *located* and *reference object*.

By updating this representation with more and more utterances by the human interlocutor, the knowledge about the furniture in the environment becomes more precise over time. Errors in the perception will eventually be overwritten by updates to the corresponding probability distributions. In principle this information could be fed back to the visual interpretation component in order to improve the classification. Further, a new knowledge about the intrinsic orientations of objects will emerge, just from the interpretation of descriptions using the intrinsic *RF*. Preferences of the interlocutor regarding the labeling of certain relations and objects can be captured with

this model. This is enabled through the discrete nature of the probability distributions which allows to represent that an interlocutor uses different labels for relations or furniture items. For example, the distribution can capture that the human calls the cupboard most of the time “cupboard”, sometimes also “shelf”, but never “bed”.

5.3.3. Resolving Conflicts

Not all descriptions by the interlocutor can be matched to the existing graph with a high score. Especially in the beginning when the graph does not contain many observations it is likely that false assumptions lead to wrong matches. Once the graph contains a few updates that originate from wrong matches it becomes even more likely that the matching process makes mistakes. Over time this behavior escalates and results in a useless model. In order to prevent this behavior the system needs a strategy which allows backtracking of multiple hypotheses and the possibility of reverting several matching decisions in favor of different matches.

The approach described here realizes this solution by keeping the n best rated matches including the corresponding updated graphs. This means, for each incoming description the algorithm identifies a set of matches corresponding to the n highest scores. These matches are assigned to respective copies of the current graph. These updated graphs are stored as a basis for the matching process of the next incoming description. Now the system matches the new description not only with the edges of the single current graph configuration, but with those of all n graphs from the previous step. The matching results now also contain a link to the predecessor's graph configuration. As a consequence, the calculated scores represent always the accumulated scores of all predecessors. This means that the best match for one description is also the best match considering the sequence of all stored graph configurations from the previous steps. This procedure leads to a final resulting graph which is the approximate global optimum of all possible interpretation sequences of the given descriptions.

Figure 5.8 displays an example for a sequence of description matching processes with corresponding graph configurations. In this example every description matching process stores the $n = 4$ best matches with their corresponding updated graphs. When matching *description 1*, the initial visually perceived graph G_0 is the base for calculating the matching scores for the

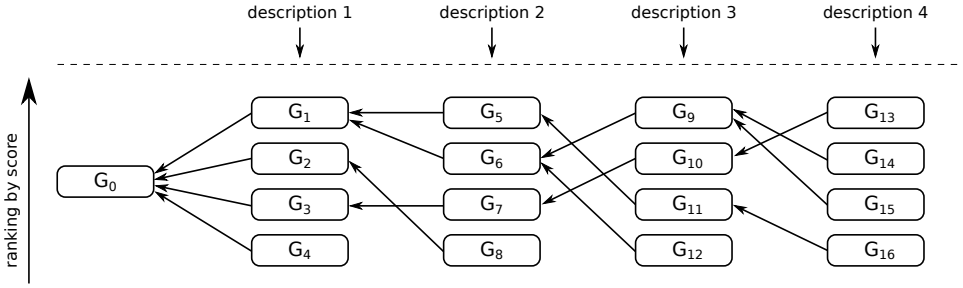


Figure 5.8.: A sequence of graph configurations for backtracking. Every node represents a different graph configuration. Configuration G_0 is the initial graph from the visual analysis. In this example every description matching process stores the $n = 4$ best matches with their corresponding updated graphs. Every graph configuration has exactly one predecessor from the previous step, represented by a link.

possible edges and interpretations. In this case, the new graph G_1 results from the highest rated match and G_4 results from the fourth best match. When the system receives *description 2* the approach tries to match the described spatial relation in all possible interpretations to all edges of graphs G_1, G_2, G_3, and G_4. In this case the highest score is calculated for matching an interpretation of the description to an edge of graph G_1. The fourth best score, however, is calculated for an edge of graph G_2. The most probable configuration at this point, based on the knowledge of the two already matched descriptions, is G_5.

After matching *description 3* to the currently believed graph configurations G_5 to G_8 the first backtracking and reversion takes place. This happens implicitly when the best match is found on a graph that was previously not considered optimal. Now the best rated match involves an edge that belongs to graph G_6 which was the second best rated configuration in the previous step. So at this point the approach implicitly reverted the previous decision of choosing the match corresponding to G_5 in favor of selecting configuration G_6. By performing the backtracking one can conclude that the most probable sequence of matches after matching *description 3* is now:

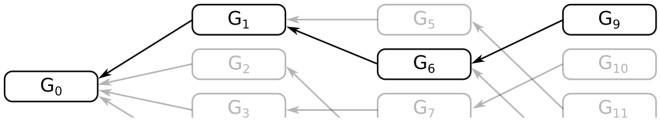


Figure 5.9.: Selected graph configurations after matching three descriptions.

In the last depicted step the approach again reverts previous decisions. Again the best rated match does not align with the previously assumed best match. Instead, the previously second best rated graph G_{10} is selected, which in turn has predecessors which were rated only as third option for their respective descriptions. This means that the approach reverts all previous decisions and assumes a completely different sequence of matches now:

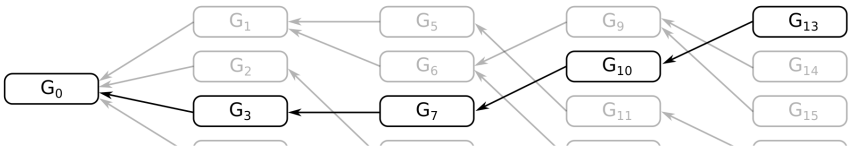


Figure 5.10.: Selected graph configurations after matching four descriptions.

The advantage of this approach is that previous hypotheses can be reclaimed when new evidences occur. In the example the sequence $G_0 \rightarrow G_1 \rightarrow G_6 \rightarrow G_9$ which was preferred after *description 3* is not discontinued because the approach still found matches based on this sequence that seem quite likely. It is possible that in a future step this sequence will be reactivated as the most probable one. The parameter n depends on how many hypotheses should be tracked in every step. The smaller this value, the less possibilities for backtracking and reversion remain for the processing. Also the chance of a hypothesis for being discontinued rises with a lower n. In the example this is the case e.g. for the sequence $G_0 \rightarrow G_2 \rightarrow G_8$. Choosing an appropriate value for n means a trade-off between the possibilities for reversion and computational load.

5.3.4. Adaptation to Personal Preferences

The results of a preliminary study (see Section 5.4.1) reveal the need for adaptation mechanism to the personal preferences of interlocutors. Not ev-

erybody behaves according to the statistics about *RF* selection. Temporary priming effects or other causes lead to a preference to one *RF* throughout a conversation for some people. In these cases the usage of the statistics in the rating of interpretations of a description is counterproductive.

In order to realize an adaptation mechanism that respects this personal preference the set of possible interpretations of a description is extended. For now, when matching a description to the furniture graph, the described spatial relation is interpreted as relative and intrinsic and rated accordingly, respecting the opposite or vehicle nature and the four different possible orientations of the *reference object*. This gives five different scores, all including the probability $P(\gamma|f_{\text{ref}}, o_{\text{ref}}, r')$ of the chosen *RF* given the assumed furniture pair and their relation which is based on the statistics from Section 5.2 (see Equation 5.3). The new mechanism calculates the scores for these five cases in three different versions. First, every score is calculated using the probability $P_s(\gamma|\dots)$ which is based on the statistics about selection of *RFs* just like before. But additionally, the scores are calculated with the alternative probabilities $P_r(\gamma|\dots)$ and $P_i(\gamma|\dots)$ which were defined manually and represent a preference either for the relative *RF* or the intrinsic *RF*. So, now the algorithm considers 15 possibilities to match a described spatial relation to one edge in the graph: The five interpretations of the description times three possible assessments of the *RF*.

However, the algorithm now has the liability to make sure that consecu-

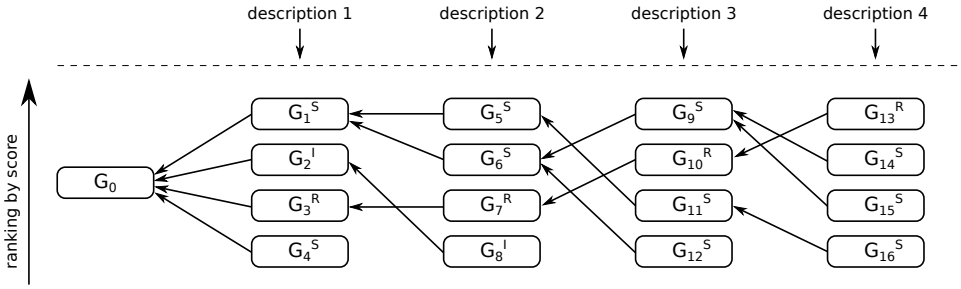


Figure 5.11.: Sequences of graph configurations assuming different preferences of RF selection. Statistical distribution (S), relative RF preferring distribution (R), intrinsic RF preferring distribution (I).

tive choices for interpretations of spatial descriptions are based on the same assessment scheme of *RFs*. Each considered sequence of matches to the graph (see Section 5.3.3) must be based on the same assumptions about the selection of *RFs* because otherwise this would not represent a temporal preference, but a spontaneous switch of assumptions for every description, which is not desired. In order to realize this, each match and the corresponding updated graph is labeled with the assumed assessment scheme its rating was based on. Each subsequent match can only be based on the previous graph configuration that was generated using the same scheme as the currently assumed assessment of *RFs* (see Figure 5.11). This way the algorithm assures that each matching sequence considers the same assumptions about preference in the selection of *RFs*.

5.3.5. Application in Human-Robot Interaction

For enabling the robotic system *BIRON* to use this approach for grounding descriptions of spatial relations a few prerequisites must be fulfilled. The robot needs advanced perception in order to pre-process the signals from two modalities so that they can be provided as input for the described approach. The visual analysis which supplies the initial configuration of the furniture graph was already described in Section 5.3.1. It analyzes the 3D information from a ASUS Xtion Pro depth sensor mounted on top of the robot in order to find furniture items in the field of view. It is also possible to register multiple scans that are obtained by turning the robot in place in order to get a wider view on the scene before starting the analysis. Just like already described in Section 3.3.2 the registration profits from the localization abilities of the robotic system which provides an initial guess for the correct matching. The *ICP* algorithm just needs to perform the fine adjustment of the *point clouds*.

The robot is also equipped with a directional microphone which allows to perceive utterances containing the spatial descriptions. For interpretation of the auditory signal the system uses the the SPHINX-4 speech recognition toolkit (Lamere et al., 2003). It is supplied with multiple task-specific grammars which are defined manually for the situations the robot is expected to face. For every recognized utterance a grammar tree containing the matched non-terminal and terminal symbols is created. As seen above, the grounding approach expects descriptions in the form *located object* $\rightarrow$ *spatial relation*

→ *reference object*. These can easily be created from the matched grammar trees using a few simple rules. Thereby it is not only possible to use canonical utterances that describe the relation. The descriptions can also be obtained from complex constructions like "please bring the chair which you put in front of the shelf yesterday" or from utterances using a different order than expected like "in front of the shelf you can see the chair".

However, the approach currently assumes that the perspectives of the agent generating the model and the agent describing the scene are the same. This means that the model is only valid if the person describing the furniture layout is located in the same spot where the initial furniture graph was perceived by the robot. The current state of the approach is not designed to handle perspective change. This is partly based on the findings of Moratz et al. (2003) who have found in their experiments that humans mostly take the robot's perspective, thus a future extension of this feature is desirable. But it is possible to use the perceived furniture instances with their respective locations and classification results for establishing new models from different perspectives. The initial graph just has to be initialized with different initial assumption about spatial relations.

A similar approach as with the egocentric models in the *ASM* system would be imaginable here as well (see Section 3.3.1). Once the robot identified a small set of typical interaction locations within the apartment, it can establish a set of disambiguation models for grounding descriptions accordingly. The same set of identified furniture in the environment could be used for initialization of all of the models. Even the probability distributions for the furniture's categories can be shared across the several models, because they are independent from the perspective. The furniture locations and viewpoint for each model are anchored in the allocentric representation, while the models themselves represent independent egocentric representations of the scenes. When descriptions of spatial relations occur, they can be matched using the model corresponding to the person's location.

In the interaction with a human interlocutor the robot should not only understand the human's utterances, it should also be able to answer or proactively formulate requests. The model for disambiguation can also be used for speech production. Obviously the verified labels of the furniture ensure the correct naming of the objects, but also the correct choice of spatial relations in the formulation supports the alignment with the interlocutor and therefore also the successful communication. From the information

about the relations in the graph the system can choose the most frequently used spatial relation for describing two objects. From the statistics about the usage of relative or intrinsic *Reference Frames* the formulation can be influenced additionally.

5.4. Evaluation

The algorithm for grounding spatial descriptions was evaluated by performing two different studies. Johannsen and De Ruiter (2013) found that the scene type has significant influence on the choice of *Reference Frames* when humans describe spatial relations. Conforming with this finding Levinson (1996) claims that "relative systems of spatial description build in a viewpoint", which implies that using the relative *RF* demands an embodied viewer in order to establish this viewpoint. Accordingly it might also have an effect on the *RF* selection whether the describing person sees a picture of the scene to describe or is actually situated in the scene. Furthermore, describing the scene to a virtual, not specifically named entity or an actual robot might influence the selection process as well. So in a preliminary study, an online survey was conducted in which the participants had to describe a depicted scene using gapped sentences. In a second study the participants were invited to a real apartment to describe the furniture to a real robot.

5.4.1. Online Evaluation

This preliminary online study has the goal to evaluate the performance of the grounding algorithm. The statistics about usage of *RFs* in various situations described in Section 5.2.1 was generated from results of an online study in which participants had to describe different visualizations of specific furniture constellations using gapped sentences. It seems reasonable to use the same technique for generating a first set of descriptions in order to evaluate the algorithm.

Goals

This initial evaluation has the goal to investigate whether the algorithm can disambiguate the received information which is highly inaccurate. Both the initially perceived identities of the furniture and the spatial descriptions are ambiguous. It is interesting to see whether the algorithm can extract the relevant information and make the correct interpretations in order to generate a valid scene model. Further, the evaluation shall expose the relevance of the knowledge about usage of *RFs* in various situations or if this has no

effect on the quality of the matching process. It may also be that humans are primed to one *RF* when describing different relations of the same scene. In this case the deployment of the statistic may be counterproductive for the correct interpretation.

Method and Procedure

The scene which the participants had to describe was captured in a real apartment outside the university's campus. The furniture in the living room of this apartment was relocated a little in order to being able to perceive the arrangement easily from one viewpoint. Also it was considered that the furniture is aligned parallel or orthogonal to the line of view to conform with the assumptions of the *RF* selection analysis from Section 5.2. The scene was captured with an ASUS Xtion Pro depth sensor and a consumer camera, both being located at the same viewpoint. The 3D information was used to segment and classify the pieces of furniture from the scene (see Figure 5.5) using the approach described in Section 5.3.1. The pictures from the consumer camera served as a visualization of the scene for the participants in the online study (see Figure 5.12).

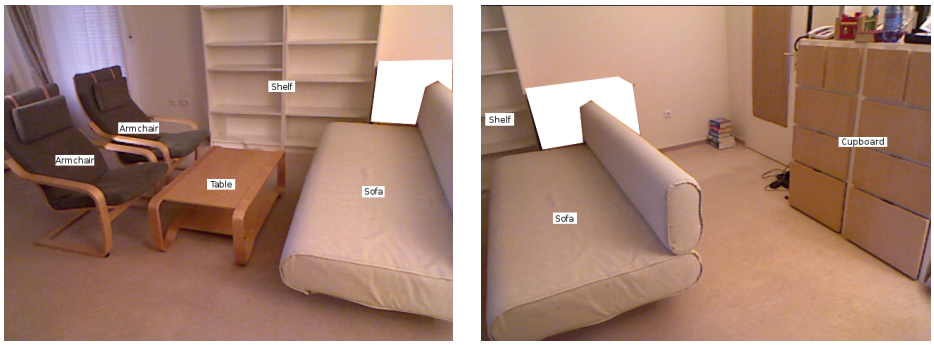


Figure 5.12.: Picture of the room that needed to be described in the online study. The labels were translated for this figure.

The participants were asked to take an online survey which was accessible over the internet so that they could participate from everywhere. At the beginning of the survey they were instructed for the task ahead. They were

asked to confirm that German is their first language. After that they had to fill in gapped sentences in order to describe the spatial relations of the form

located object [to be filled in] *reference object*

For example, "The table is ..... the armchair". The sentences were all German as well as the labels on the images which were visible to the participants throughout the whole survey. Only one sentence was shown at a time and the order of the sentences was shuffled by the operator regularly so that priming effects from the specific sequence of descriptions could be excluded. Using these gapped sentences it could be controlled that all possible relation were described.

For analysis the algorithm was fed with the visual results and the descriptions from one participant. For comparison the descriptions were labeled with a ground truth interpretation. The final decisions of the algorithm for grounding each description were compared to the ground truth and the resulting furniture graph was compared to the real furniture arrangement.

Analysis

The survey was taken by 52 participants (mainly students, average age: 28, SD: 3.6, male: 72%) who had to fill in 20 gapped sentences. As described above, the resulting descriptions were fed to the grounding algorithm, together with the results from the visual interpretation. Figure 5.13 depicts the resulting probability distributions from the visual interpretation used for the analysis. Notice that the shelf (SHELF_01) is misclassified by the computer vision component as armchair.

Finally **90.87% of all descriptions** are matched to the correct edge in the graph, which means that the algorithm matched for every participant on average 1.83 descriptions to wrong edges. Accordingly **36.54% of the entire description sequences** uttered by one participant can be grounded correctly by the algorithm (see Figures 5.14b and 5.14a for visualization of the data). For comparison, if instead of the statistic from the *RF* selection experiment a uniform distribution of expected *RFs* is used, even **96.83% of all descriptions** are matched correctly and **61.54% complete sequences** are grounded correctly. So here the uniform distribution seems to work better than the recorded statistics about human usage of *RFs*.

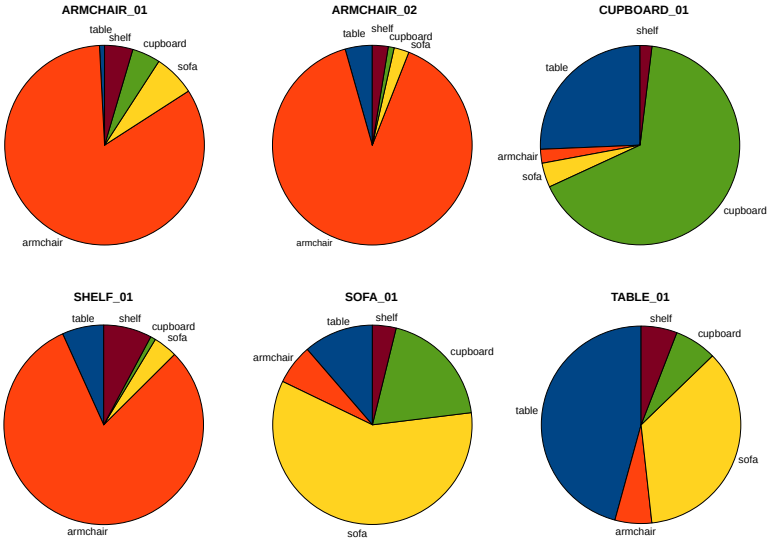
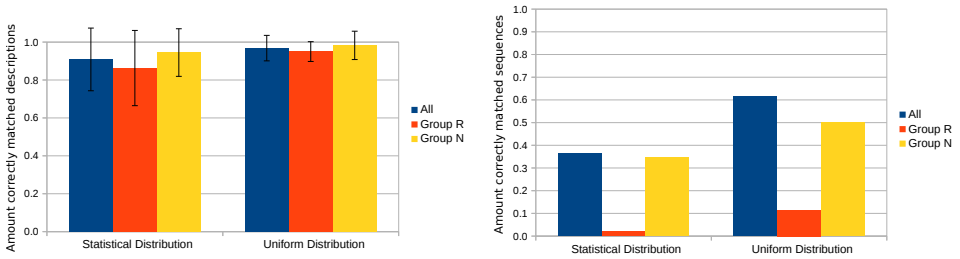


Figure 5.13.: Probability distributions from visual analysis for the categories of the six segmented furniture objects in the scene for the online study.

However, one can observe that many participants never or only very rarely used the intrinsic *RF*. In fact, the rate of participants that used the intrinsic *RF* in at least 10% of the descriptions is **55.77%**. Accordingly, nearly half of the participants almost exclusively used the relative *RF*. This observation suggests the assumption that some precondition of the test leads to an alternative *RF* selection preference compared to the observed choices described in Section 5.2. This may be due to the fact that in this study a photography of the complete scene was used instead of rendered images without background as in the selection evaluation study. For a more detailed analysis of the implications of this circumstance the participants are divided into two groups: *group R* contains the description sequences that use less than 10% intrinsic *RFs* and therefore show a clear preference of the relative *RF*. The remaining sequences form *group N* which does not show any obvious preference.

When only looking at *group R*, the rate of overall correctly matched descriptions is **86.30%** and only **4.35% correctly matched complete sequences** are observed. For *group N* **94.48%** of all descriptions are matched



(a) Correctly grounded descriptions (b) Correctly grounded sequences

Figure 5.14.: Average amount of correctly grounded descriptions per sequence and total amount of correctly grounded complete sequences. Statistical and uniform distribution refer to different models for rating of *RF* selection.

correctly and even **62.07% of the complete sequences** are grounded correctly. So here a large difference in the performance can be observed when the participants are obviously primed to one *Reference Frame* compared to the other group that showed no general preference and therefore acted more as predicted by the statistics.

The resulting furniture graphs are correct for **84.62% of all sequences**. This splits up to 82.61% for *group R* and 86.21% for *group N*. This means that in most cases the algorithm can fix the false categorization of the visual interpretation component. In the remaining cases two vertices are confused. Of all vertices **38.78%** are assigned the correct inherent orientation through the algorithm's evaluation of intrinsic *RFs*. However, 21.15% are provided with wrong inherent orientations. Nevertheless, it can be seen that the visual perception can benefit from the grounding of verbal descriptions.

In summary, the algorithm is able to ground most of the descriptions correctly and to generate correct beliefs about the identities of objects in the environment. This works particularly well for description sequences that conform with the expectations about usage of *Reference Frames*. When participants behave differently than expected by preferring a certain *RF* the algorithm makes many mistakes. These results lead to the improvements of the algorithm described in Section 5.3.4. The implemented adaptation mechanism to personal temporal preferences in the selection process will be tested in the real-world evaluation (see Section 5.4.2).

5.4.2. Real-World Evaluation

As stated previously, there are concerns about the validity of the online evaluation with respect to the applicability of the system in real-world situations. The participants of the online study were not embodied in the situation and they described the scene not in a dialog but essentially to nobody but the computer screen. This might have influence on the choices they make regarding the selection of the appropriate reference frame. Further the preliminary study revealed a few shortcomings of the approach which were incorporated in a revised strategy for adapting to individual preferences or temporary priming of *RFs* (see Section 5.3.4).

So I conducted a second experiment in which the participants were actually present in the apartment looking at the furniture which had been to describe. Also a real, embodied interlocutor has been present in the form of our robot companion *BIRON*. This enabled the participants to describe the scene in a real-world situation in an unrestricted dialog to an actual robot.

Goals

Again, the main goal of this evaluation is to find out how well the grounding algorithm works for interpreting spatial descriptions correctly. In contrast to the previous test, the descriptions originate from a real-world situation and might be chosen differently. Also, since there was no guideline for which pairs of furniture had to be described, the set of available information varies from test to test. So more specifically, a goal is to see whether the selection of *RFs* differs from the previous test and whether the system performs differently regarding the quality of the matching decisions. The focus of this evaluation lies on the assessment of the system's performance under different preconditions. How does the condition under which the descriptions were generated, the quality of the visually perceived information and the strategy for adapting to selection preferences influence the performance of the algorithm?

Method

As setting for the tests the "intelligent apartment" laboratory was chosen (see Figure 5.15). This is a realistically furnished research apartment inside the university building, equipped with multiple sensors and actuators.

Figure 5.15.: Scene from the evaluation while a participant is describing spatial relations in a real-world apartment.

The tests took place in the living room. A viewpoint was defined from which the participants should describe the scene. Just like for the online study, the furniture was arranged in a way that it could all be perceived from this viewpoint (see Figure 5.16). The robot companion *BIRON* was placed at the viewpoint facing towards the furniture arrangement. The participants were asked to stand right next to the robot and describe the furniture arrangement inside the apartment's living room.

The German descriptions were recorded and annotated afterwards. I renounced on using a speech recognition system for automatic transformation of speech to a machine-readable format because the focus of this evaluation lies on the analysis of the grounding algorithm and not a complete system. Errors in the recognition would distort the performance of the algorithm which is not desirable at this point. Again, the scene was captured with an ASUS Xtion Pro depth sensor mounted on top of the robot in order to segment and classify the pieces of furniture from the scene using the approach described in Section 5.3.1. Like in the online study the algorithm was fed with the visual results and the descriptions from one participant. Thereby

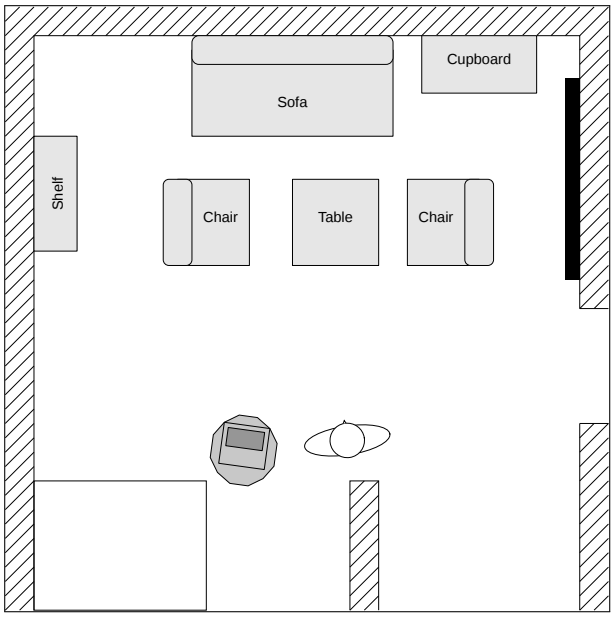


Figure 5.16.: Furniture layout for evaluation including the robot and a participant

different preconditions were tested regarding the quality of the visual perception, as well as different adaptation strategies. The final decisions of the algorithm for grounding each description were again compared to the ground truth and the resulting furniture graph was compared to the real furniture arrangement.

Procedure

The participants were introduced to the setting and the robot. They were asked to describe as many reasonable pairs of furniture as they can think of until the robot signals that it heard enough. Thereby they were explicitly made aware that both objects of one pair can be the located objects of a description. Further, they were asked not to proceed in a systematic way when choosing the pairs, but to describe the arrangement in a random order. In order to keep up the awareness that the descriptions should be addressed to the robot which had the same perspective on the scene,

it interacted with the participant throughout the test. This behavior was controlled in a "wizard of oz" manner by the operator from a remote computer. The behavior contained a short introduction, the invitation to start the descriptions, short feedback prompts in between the descriptions to signal attention, and the closing of the tests. The operator decided to quit the test when at least 12 descriptions were made covering all reasonable spatial relations. Finally, the participants were asked to complete a questionnaire about their experience with robotic systems and to fill in a data privacy statement (see Appendix E).

Analysis

The study was performed with 30 participants (mainly students and university staff, average age: 27.3, SD: 6.1, male: 55.2%) who produced on average 16.80 (SD: 2.78) descriptions of the furniture layout. It is interesting that in this experiment the rate of intrinsic *RFs* used for the descriptions is even lower than in the online study. While the rate of description sequences using the intrinsic *RF* in **at least 10%** of the descriptions (*group N*) was **55.77%** in the online study, it is now even just **16.67%**. So in this experiment *group R* (sequences using less than 10% intrinsic *RFs*) is much larger than *group N*. I can only speculate about the reasons for this effect, because the prerequisites for this test are different in many ways to the online experiment, but it might be that the actual embodiment in the scene and the presence of an interlocutor influenced the choice of *Reference Frames*. It is therefore even more crucial to apply the adaptation mechanism (Section 5.3.4), because the participants behave differently than expected by the statistics from the *RF* selection study.

The algorithm was run with and without adaptation mechanism, with only uniformly distributed *RF* probabilities and with different outcomes of the visual interpretation of the scene. The parameters of the *ISM* approach for classification of furniture were manipulated in a way that different outcomes in terms of recognition quality could be produced. Hereby a set of three different visual interpretation results with descending quality could be produced: Result *A* arises from optimized parameters and represents a correct interpretation with quite clear preference of the correct class (avg. entropy: 1.815). Result *B* contains a mistake for one item and the probabilities are distributed a little more evenly (avg. entropy: 1.867). Result

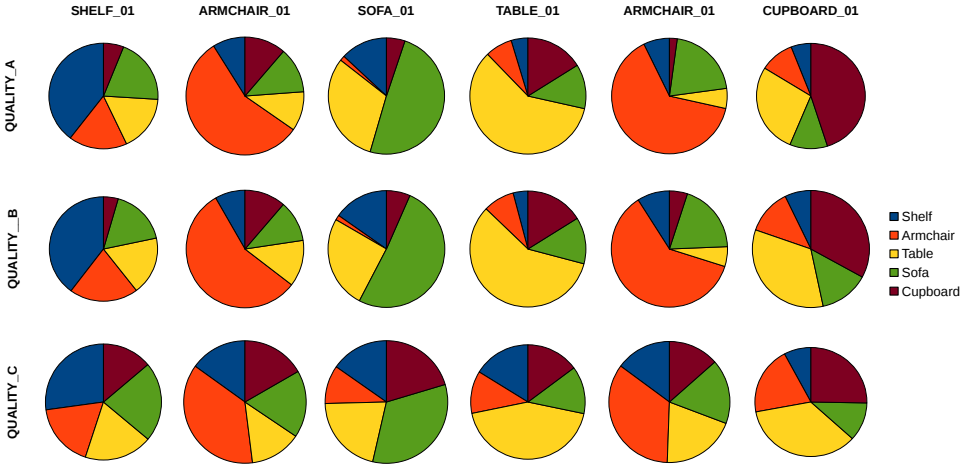
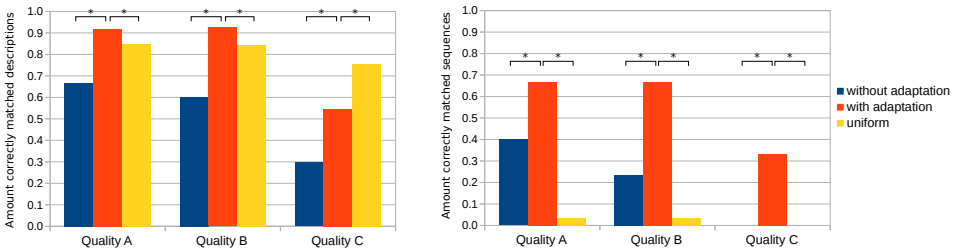


Figure 5.17.: Probability distributions of the visual interpretation of furniture in the real-world study in three different qualities.

C contains a mistake with a quite strong belief and significantly more balanced probability distributions (avg. entropy: 2.199). See Figure 5.17 for a visualization of the three different results.

The results of the grounding algorithm under different preconditions are displayed in Figure 5.18. In principle, in the case with activated adaptation mechanism, more descriptions are grounded correctly than without. For example, in the case of visual recognition result A, 40% of all sequences are grounded completely correct without adaptation, while the rate rises to 66.7% when the adaptation mechanism is applied (Figure 5.18b). It is surprising that the version with uniform probability distribution almost never grounds a complete sequence correctly. These differences in sample means between the non-adaptation and adaptation case, as well as between adaptation and uniform case have a statistical significance ($p < 0.05$) according to the McNemar's test for paired nominal data with dichotomous traits (McNemar, 1947). The same tendency holds for the absolute numbers of matched descriptions. In total **66.7%** of all descriptions are matched correctly **without adaptations**. Compared to this, the case with **activated adaptation reaches a success rate of 91.9%** (Figure 5.18a). The value for the uniform case lies between the non-adaptation and adaptation case. The large difference between the performance on complete sequences and



(a) Correctly grounded descriptions (b) Correctly grounded sequences

Figure 5.18.: Amount of correctly grounded complete overall descriptions and sequences. Qualities A, B and C refer to the different results from the initial visual analysis.
* significant difference ($p < 0.05$)

overall descriptions of the uniform case can be explained by analyzing the logs. Apparently, the algorithm using uniform probability distribution has problems grounding descriptions of the two armchairs and the table correctly. Here, the *RFs* are chosen very poorly, which leads to many false matches that, however, do not influence the overall rating of the complete furniture graph so much. Here, the algorithm always makes a few mistakes in the matching of these particular descriptions, but performs very well for all others. This explains why only few complete sequences are correct, but the overall amount of correctly matched descriptions is high. Using the Wilcoxon signed-rank test for paired data that cannot assumed to be normally distributed (Wilcoxon, 1946), these differences are as well found statistically significant ($p < 0.05$). These effects can be observed for all visual interpretation qualities.

However, it is obvious that the amounts of correct matches diminish with lower quality of the visual interpretation results. Thereby the adaptation mechanism seems to help rectifying some of the misinterpretations that come from false *a priori* information, but at least in quality case *C* the success rates diminish as well. Interestingly the algorithm using uniform probability distribution seems to be less sensitive to the initial graph's quality.

The difference between *group R* and *N* is very large in the non-adaptation case. On average over all visual interpretation qualities, the absolute match-

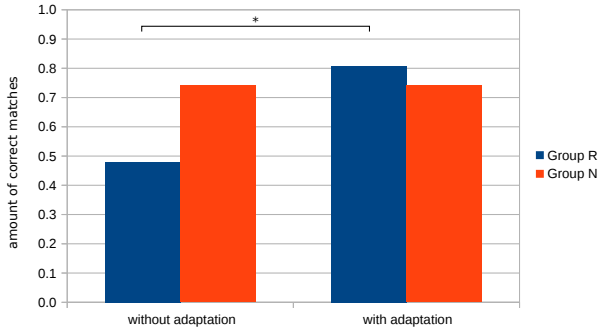


Figure 5.19.: The overall amount of correct matches by *groups* R and N.
* significant difference ($p < 0.05$)

ing rate for the former sums up to 47.9% and for the latter to 74.7%. It is clear that here, as well, the description sequences with a clear preference for the relative RF make the difference. In the adaptation case the rate for *group* R rises to 80.7% while the rate for *group* N stays with 74.7% unchanged (see Figure 5.19). The difference in sample means in *group* R is again found statistically significant using the Wilcoxon signed-rank test ($p < 0.05$). This proves that the adaptation algorithm correctly assigns a different expectation of RF usage to the correct description sequences.

The quality of the resulting furniture graphs varies as well. In the non-adaptation case the amount of misclassified vertices is on average 13.89% for visual interpretation quality A and continuously rises to 38.22% of quality C. Whereas, when using adaptation the error rate is at under 3% for visual results A and B, and goes up to 24.44% for result C (see Figure 5.20). It is interesting to note, though, that the results for *group* N are again much better than those for *group* R. For results A and B in both cases (with and without adaptation) the algorithm does not make a single mistake in terms of correct grounding of furniture. Since only few intrinsic RFs were used for the descriptions, accordingly few furniture objects are assigned a correct orientation. Averaging all trials, the amount of vertices that are assigned a correct orientation is 17.78%. No effects from the different conditions can be observed.

Additionally, the information gain by the grounding algorithm is analyzed. As stated before, the entropy values of the probability distributions from

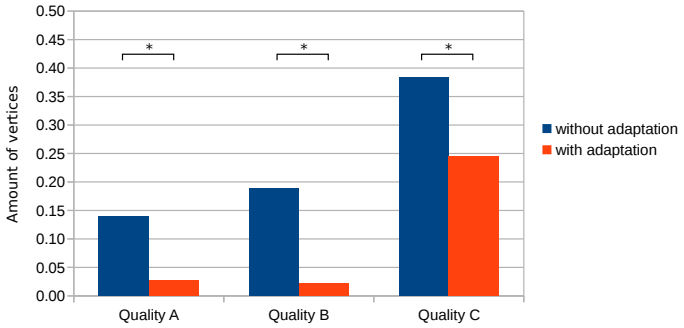


Figure 5.20.: Overall amount of vertices that were assigned a wrong label.
* significant difference ($p < 0.05$)

the visual interpretation component range from 1.815 to 2.199. The average entropy value of all graphs from all trials is 0.335. A low entropy means a strong uneven distribution of probabilities, in other words few classes have a high probability and many others have a very low probability. This means that the model is more sure about the identity of the objects because the probability of the correct class is very high in comparison to the probabilities of the other classes. Since the entropy value rises significantly after applying the grounding algorithm, one can say that this generated a large information gain.

Discussion Summarizing these insights, one can say that a robotic system should be able to adapt to an interlocutor in terms of expected usage of *Reference Frames* because these vary strongly between humans. The presented system proved to be able to perform this adaptation successfully. The implemented adaptation algorithm performs in most cases significantly better than the versions without adaptation and uniform probability distributions. Furthermore, it generates a valid symbolic representation of the furniture in its surrounding and is able to improve the results from the visual analysis through grounding of the verbal descriptions. The quality of the grounding process and the resulting furniture graph depends, however, strongly on the initial quality of the probability distributions. The system is able to rectify minor mistakes in the visual perception, but since the algorithm represents a probabilistic approach, larger errors lead to an establishment of wrong

maxima in the probabilistic model.

The hypothesis maintaining strategy of resolving conflicts proved to be very useful for the given task. When looking at the actual final sequences of chosen matching hypotheses, it is striking that previous hypotheses are often reverted in favor of others that were previously rated much worse. Reverting to hypotheses there were originally rated only 60th or 70th in applicability is not uncommon. This is certainly one key feature of the algorithm that enables these good results.

5.5. Summary

We have seen that the robot's abilities in perceiving a situation improves through the usage of the newly gathered instance and referential information. The dialog component of the robot is now able to establish a common ground with the human interlocutor. The speech production considers the probability distributions of the object's labels and those of the spatial relations used to describe the locations of furniture as described in Section 5.3.5. Also preferences in the usage of certain reference frames can be adjusted according to the human's habits. This represents an implicit agreement on the usage of those linguistic properties between robot and human. Considering these preconditions in speech production, the produced formulations can be adjusted to the way the interlocutor formulates his or her requests. This increases the chances for successful communication and is a further step towards a more natural communication scheme for artificial systems and social behavior.

Similarly, the previous sections revealed that the collaboration between components from the auditory and visual domain can enhance the interpretation of the communicative signals from a human interlocutor. The visually perceived layout of the scene is used in the proposed system to disambiguate verbally uttered descriptions of spatial relations. Using the isolated speech recognition results, this disambiguation would not be possible and the interaction between human and robot would become very inefficient.

The proposed disambiguation method establishes a probabilistic network structure representing the entities in the environment which are relevant for the interpretation of the descriptions. The initial topology of the network arises from the visual, three-dimensional analysis of the scene. Now

consecutively emerging descriptions of pairs of objects in the robot's (extended) field of view are matched and merged with the probabilistic network. While preserving a number of hypotheses for each match, the proposed algorithm finds the optimal sub-tree of the graph for matching the incoming descriptions. The process considers the located and referenced object of the descriptions, as well as the spatial relation between them. Especially for the disambiguation of the spatial relations, the system uses empirical results from an analysis about common usage of different *Reference Frames* in human speech under certain preconditions regarding the entities to be described. These results are used to improve the interpretation process. In a temporal integration process the system evaluates approximately optimal choices considering multiple preserved hypotheses for all previous interpretation decisions.

Apart from the improved skills in *HRI*, this raises mutual benefits for the robot's visual and auditory subsystems alike. The semantic interpretation of the speech signal improves through the matching with the results from the visual perception component. In turn, the semantic interpretation of the visually perceived objects profits from the correct interpretation of speech, because labels can be refined through the understood references in the utterances. Further, it would be imaginable to re-run the training of the models that facilitate the object recognition capabilities of the robot with additional imagery from the referenced and labeled objects. In the long run one could also refine the statistical model for selection of *Reference Frames* or even establish personal models for individual human interaction partners from the results of the grounding and interpretation process.

A further extension of the proposed approach to more complex scenes and scenarios is imaginable. When applied to a scene containing more furniture, the complexity does not rise exponentially, because the effort for calculation of the hypotheses depends on the number of evaluated edges in the graph. The spatial relations are only established with the furniture in the direct surrounding. Therefore, the number of edges rises linearly with the number of furniture objects.

Furthermore, an extension for employing different perspectives of the involved agents is possible. The relative *RF* could be split into one for the perspective of agent A and one for agent B's perspective. The matching process would just need to consider these additional interpretations of the uttered descriptions. This would, however, require justification from a psy-

chological point of view.

Another possible extension is imaginable for proceeding from furniture to other entities within the apartment. For example, processing of descriptions of manipulable household objects in respect to each other or their supporting structures would be interesting. Swadzba et al. (2009) already present previous work on this topic.

Chapter 6

Discussion & Conclusion

In this thesis, I investigate aspects of how a *situation model* should be constituted for a general-purpose service robot that can interact naturally with humans. The considerations include finding a way to digitally represent the information the robot has gathered over time, a selection of types of information required for such a model, and mechanisms that allow the deployment of the aggregated knowledge in real-world interaction. In order to recap the contributions of this work I will review the five dimensions of situations defined by Zwaan and Radvansky (1998) — time, space, causality, intentionality, and protagonist. All five dimensions are treated in this thesis, though they are not evenly weighted. Clearly, the main focus is on the space dimension, namely, the perception of the geometric structure and semantics of the space surrounding the robot. The time dimension is approached primarily through temporal integration of gathered information, like utterances, visual features and dynamic properties in egocentric models. Causality is covered only marginally in high-level components that try to relate changes in egocentric models to the actions of the interlocutor and the assumptions about human usage of reference frames. The same applies for intentionality, which is considered in the high-level robot behavior for altering the attention to situations in which a human shows the intention to manipulate objects. The protagonist dimension, however, reoccurs on a regular basis when interpreting actions to extract dynamic properties and when grounding utterances in interaction.

As a first approach towards a general *situation model* for a mobile service robot, one must consider the analysis and representation of the geometric

structure in the robot's environment. Thereby, the focus is on partitioning the workspace in a way that allows the robot to distinguish between individual entities. Through observation, the system is able to apply dynamic properties to certain structures in its field of view, allowing it to detect enclosed movable structures. Based on this capability, one can make a proposal for a unified spatial representation incorporating different spatial scopes and data structures. This relates mainly to the first of four posed research questions that outline the work described in this thesis (see Section 1.3). This question is probably the most fundamental one and addresses deliberations about the representation of spatial knowledge. Chapter 3 contributes to this question by presenting a representation based on an areal geometric view of the environment. Additionally, distinct egocentric models are anchored in this view, enabling components to store information in a self-centered way. To complete the set of requirements identified for persistent representations on a typical mobile robot, the unified spatial representation includes a layer representing instance-based information allocentrically.

The relevance of relations between those parts of the comprehensive representation is particularly addressed. Transforming information between several models and between points in time is considered eminently crucial. Therefore, I present an algorithm for transferring knowledge from an egocentric model to other viewpoints at different times. It allows a robot to apply previously-gathered knowledge in new situations. This contribution relates to Research Question 3.

Furthermore, Chapter 3 grapples with how to decide which information might be relevant in future tasks and when a robot should pro-actively claim new facts (Research Question 2). Therefore, a robot behavior is presented that makes use of generalized spatial representation to identify situations in which a human is about to perform a possibly informative action. It enables the robot to position itself accordingly to be able to perceive the action.

The same question is addressed in Chapter 4. Specifically, the relevance of different types of semantic knowledge about the object identities and areas in the robot's surroundings is analyzed. I particularly focus on the importance of the functional roles of regions or complete rooms within an apartment. Obviously for referential communication, a large part of *HRI*, the knowledge about the identities of objects is also very important. Chapter 4 presents respective tools for detection and recognition of furniture using a 3D *ISM* approach (section 4.1), a more general object classification

mechanism based on *AdaBoost*, incorporating a vast amount of arbitrary visual features (Section 4.2), and a categorization approach for room type recognition (Section 4.3).

With the solutions described so far, the robot is able to visually perceive many aspects of its environment. It can perceive geometric layout, including dynamic properties, it is able to identify semantic labels of objects and areas, and it has strategies to transfer information between several representations and points in time. However, rich and natural *HRI* can only be achieved when also considering the auditory modality (Li and Wrede, 2007; Breazeal, 2004). To establish a usable *situation model*, the human interaction partner's verbal utterances must be grounded in perception in order to further improve the knowledge about the relevant entities. This includes gathering facts about objects, but also aligning to the habits of the interlocutor and establishing a common ground. This problem is approached in Chapter 5.

Here, I present a system for combining verbal and non-verbal cues. It uses the possibly inaccurate results from the visual perception of a scene to ground spatial descriptions of humans by considering generally observed human conventions, personal habits and alignment in a common ground. This approach has been shown to interpret even ambiguous descriptions correctly and can rectify mistakes from the visual perception. This contributes to Research Question 3 through temporal integration to establish a common ground, as well as Question 4 by integrating information of different modalities into the interpretation process. The system uses a custom instance-bases egocentric network representation, but solutions for integrating it into the central spatial representation and in multi-perspective applications are discussed in Section 5.5.

To demonstrate most of the aspects described in this thesis, an integrated robot behavior was implemented realizing the *lost key scenario*. The robot is able to analyze an indoor scene visually (using tools from Chapter 4). Uttered spatial descriptions of the furniture are used to refine the spatial model of furniture's location. The resulting information is stored allocentrically for later reference. The robot then constantly observes the actions of the humans in the room and repositions itself when it expects a manipulation action. Performed actions can be described by the human leading the robot to assign of a label to the manipulated object. The high-level reasoning strategies described in Section 3.3.4 are used to update the locations of the reference objects. At any time, the robot can describe where

tracked objects are in relation to the furniture in their immediate vicinity by using the rules from the common ground established through the grounding system in Chapter 5. This behavior was successfully demonstrated during the on-site inspection of the collaborative research cluster "Alignment in Communication" at Bielefeld University.

For future development, proceeding the focus from referential communication to collaboration between robot and human is imaginable. This requires further developing the grounding process to more sophisticated mechanisms to gather perspective. The interpretation of action must also be enhanced so we may infer intention and predict goals. From a complex scenario involving cooperative problem solving, additional interesting research questions arise. What are the requirements for planning actions? How can the interlocutor be included in the closed-loop control of the actions? Which additional information must be maintained by a corresponding *situation model*?

In conclusion, this thesis contributes to better understanding what a domestic service robot companion requires in terms of representation and application of spatial knowledge in real-world situations. The general topic has large potential for further investigation in future research.

Bibliography

Agarwal, S. and Roth, D. (2006). Learning a sparse representation for object detection. *Computer Vision—ECCV 2002*, pages 1–15. 4

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). FREAK: Fast retina keypoint. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 510–517. 4.2

Albrecht, S. and Wiemann, T. (2011). Matching CAD object models in semantic mapping. In *ICRA 2011 Workshop Semantic Perception, Mapping and Exploration*, pages 1–6, Shanghai, China. 3.1

Bader, M., Prankl, J., and Vincze, M. (2013). Visual Room-Awareness for Humanoid Robot Self-Localization. In *37th Annual Workshop of the Austrian Association for Pattern Recognition*, pages 1–10, Innsbruck, Austria. 1.1

Ballard, D. (1981). Generalizing the Hough transform to detect arbitrary shapes. 4.1.2

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417. 4.2

Besl, P. and McKay, H. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256. 3.3.3

Beun, R.-J. and Cremers, A. H. (1998). Object reference in a shared domain of conversation. *Pragmatics & Cognition*, 6(1):121–152. 5

Beuter, N., Swadzba, A., Kummert, F., and Wachsmuth, S. (2011). Using articulated scene models for dynamic 3d scene analysis in vista spaces. *3D Research*, 1(3):4. 3.3.3

Biederman, I., Mezzanotte, R. J., and Rabinowitz, J. C. (1982). Scene perception: detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14:143–177. 4

Bischoff, R. and Graefe, V. (2002). Dependable multimodal communication and interaction with robotic assistants. In *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, pages 300–305. IEEE. 5

Björkman, M. and Kragic, D. (2010). Active 3D scene segmentation and detection of unknown objects. In *2010 IEEE International Conference on Robotics and Automation*, pages 3114–3120. IEEE. 3.2.2

Branigan, H. P., Pickering, M. J., and Cleland, A. A. (2000). Syntactic co-ordination in dialogue. *Cognition*, 75(2):13–25. 1

Breazeal, C. (2004). Social interactions in HRI: The robot view. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):181–186. 5, 6

Breazeal, C., Brooks, A., Chilongo, D., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., and Lockerd, A. (2004). Working collaboratively with humanoid robots. *Humanoid Robots, 2004, 4th IEEE/RAS International Conference on*, 1:253 – 272. 5, 5.1

Brennan, S. E. and Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of experimental psychology. Learning, memory, and cognition*, 22(6):1482–1493. 1, 5

Bronstein, A. M., Italiana, S., and Guibas, L. J. (2011). Shape Google : geometric words and expressions for invariant shape retrieval. *Methods*, 30:1–20. 4

Burgess, N., Spiers, H. J., and Paleologou, E. (2004). Orientational manoeuvres in the dark: dissociating allocentric and egocentric influences on spatial memory. *Cognition*, 94(2):149–166. 3.1

Campbell, N. D. F., Vogiatzis, G., Hernández, C., and Cipolla, R. (2010). Automatic 3D object segmentation in multiple views using volumetric graph-cuts. *Image Vision Comput.*, 28(1):14–25. 3.1

Carlson, L. (1999). Selecting a reference frame. *Spatial cognition and computation*, 1:365–379. 5.2, 5.2

Carlson-Radvansky, L. A. and Irwin, D. E. (1993). Frames of reference in vision and language: where is above? *Cognition*, 46:223–244. 5.2

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2724–2729. IEEE Comput. Soc. Press. 3.3.3

Chitta, S., Jones, E. G., Ciocarlie, M., and Hsiao, K. (2012). Perception, Planning, and Execution for Mobile Manipulation in Unstructured Environments. *IEEE Robotics and Automation Magazine, Special Issue on Mobile Manipulation*, 19(2). 3.1, 3.3

Clark, H. H., Schreuder, R., and Buttrick, S. (1983). Common ground at the understanding of demonstrative reference. *Journal of Verbal Learning and Verbal Behavior*, 22(2):245–258. 5

Cohn, A. G. and Renz, J. (2007). Qualitative Spatial Reasoning. In van Harmelen, F., Lifschitz, V., and Porter, B., editors, *Handbook of Knowledge Representation*, pages 551–596. Elsevier. 5.2

Colby, C. L. and Goldberg, M. E. (1999). Space and attention in parietal cortex. *Annual review of neuroscience*, 22:319–49. 3.1

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision*, pages 59–74. 4

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, pages 303–312. 3

Dautenhahn, K. (2007). Socially intelligent robots: dimensions of human-robot interaction. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 362(1480):679–704. 1

De Graef, P., Christiaens, D., and D'Ydewalle, G. (1990). Perceptual effects of scene context on object identification. *Psychological Research*, 52:317–329. 4

Dix, A., Finlay, J., Abowd, G. D., and Beale, R. (2004). *Human-Computer Interaction*, volume Third. 1

Dominey, P., Boucher, J.-D., and Inui, T. (2004). Building an adaptive spoken language interface for perceptually grounded human-robot interaction. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, volume 1, pages 168 – 183. IEEE. 5

Elfring, J., van den Dries, S., van de Molengraft, M., and Steinbuch, M. (2012). Semantic world modeling using probabilistic multiple hypothesis anchoring. *Robotics and Autonomous Systems*, 61(2):95–105. 3

Engelhardt, K. G. and Edwards, R. A. (1992). *Human-Robot Integration for Service Robotics*. 5

Erlhagen, W., Mukovskiy, A., Bicho, E., Panin, G., Kiss, C., Knoll, A., van Schie, H., and Bekkering, H. (2006). Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning. *Robotics and Autonomous Systems*, 54(5):353–360. 5

Espinace, P., Kollar, T., Roy, N., and Soto, A. (2013). Indoor scene recognition by a mobile robot through adaptive object detection. *Robotics and Autonomous Systems*, 61:932–947. 4, 4

Fei-Fei, L. and Perona, P. (2005). A Bayesian Hierarchical Model for Learning Natural Scene Categories. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:524–531. 4

Fernaeus, Y., Håkansson, M., Jacobsson, M., and Ljungblad, S. (2010). How do you play with a robotic toy animal? In *Proceedings of the 9th International Conference on Interaction Design and Children - IDC '10*, page 39. 1.1

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395. 3.4.2, 5.3.1

Fisher, M. and Hanrahan, P. (2010). Context-based search for 3D models. 4

Freund, Y. and Schapire, R. E. (1997). A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computing Systems and Science*, 55:119–139. 4, 4.2, 4.2.1

Fritsch, J., Kleinehagenbrock, M., Lang, S., Fink, G. A., and Sagerer, G. (2004). Audiovisual person tracking with a mobile robot. In Groen, F., Amato, N., Bonarini, A., Yoshida, E., and Kröse, B., editors, *Proc. Int. Conf. on Intelligent Autonomous Systems*, pages 898–906, Amsterdam. IOS Press, Citeseer. 1.1

Fritsch, J., Kleinehagenbrock, M., Lang, S., Plötz, T., Fink, G., and Sagerer, G. (2003). Multi-modal anchoring for human-robot interaction. *Robotics and Autonomous Systems*, 43(2-3):133–147. 3.4.1, 3.4.1, 3.4.1

Gates, B. (2007). A Robot in Every Home. *Scientific American Magazine*, 296(1):58–65. 1

Gelfand, N. and Guibas, L. J. (2004). Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pages 214–223, New York, NY, USA. ACM. 3.1

Gorniak, P. and Roy, D. (2005). Probabilistic Grounding of Situated Speech using Plan Recognition and Reference Resolution. In *In Proceedings of the Seventh International Conference on Multimodal Interfaces (ICMI 2005)*, pages 138–143, New York, NY, USA. ASM. 5

Goron, L. C., Marton, Z.-C., Lazea, G., and Beetz, M. (2012). Robustly Segmenting Cylindrical and Box-like Objects in Cluttered Scenes using Depth Cameras. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. 3.2.2

Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2009). Mechatronic design of NAO humanoid. *2009 IEEE International Conference on Robotics and Automation*. 1.1

Graf, R. and Herrmann, T. (1989). *Zur sekundären Raumreferenz: Gegenüberobjekte bei nicht-kanonischer Betrachterposition*. Arbeiten aus dem Sonderforschungsbereich 245, "Sprechen und Sprachverstehen im Sozialen Kontext". Lehrstuhl Psychologie, Mannheim. 5.2

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1458–1465. IEEE Computer Society. 4

Grisetti, G., Stachniss, C., and Burgard, W. (2007). Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1):34–46. 3

Gross, H. M., Boehme, H., Schroeter, C., Mueller, S., Koenig, A., Einhorn, E., Martin, C., Merten, M., and Bley, A. (2009). TOOMAS: Interactive shopping guide robots in everyday use - Final implementation and experiences from long-term field trials. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 2005–2012. 1.1

Haasch, A., Hohenner, S., Hüwel, S., Kleinehagenbrock, M., Lang, S., Toptsis, I., Fink, G., Fritsch, J., Wrede, B., and Sagerer, G. (2004). BIRON – The Bielefeld Robot Companion. In Prassler, E., Lawitzky, G., Fiorini, P., and Hägele, M., editors, *Proc. Int. Workshop on Advances in Service Robotics*, pages 27–32, Stuttgart, Germany. Fraunhofer IRB Verlag. 1.1

Hartley, T., Trinkler, I., and Burgess, N. (2004). Geometric determinants of human spatial memory. *Cognition*, 94(1):39–75. 3.1

Hawes, N., Hanheide, M., Sjöö, K., Aydemir, A., Jensfelt, P., Göbelbecker, M., Brenner, M., Zender, H., Lison, P., Kruijff-Korbayová, I., and Others (2010). Dora The Explorer: A Motivated Robot. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 1617–1618, Toronto, ON, Canada. International Foundation for Autonomous Agents and Multiagent Systems. 1.1

Helmer, S., Meger, D., Viswanathan, P., McCann, S., Dockrey, M., Fazli, P., Southey, T., Muja, M., Joya, M., Little, J., Lowe, D., and Mackworth, A. (2009). Semantic Robot Vision Challenge: Current State and Future Directions. Technical report. 1.1

Hornung, A., Phillips, M., Jones, E. G., Bennewitz, M., Likhachev, M., and Chitta, S. (2012). Navigation in Three-Dimensional Cluttered Environments for Mobile Manipulation. In *IEEE Int. Conference on Robotics and Automation*, Minneapolis, Minnesota. IEEE. 3, 3.1

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206. 3

Huber, D., Kapuria, A., Donamukkala, R., and Hebert, M. (2004). Parts-based 3D object classification. In *CVPR*, volume 2, pages 82–89. 4

Hunter, G. M. and Steiglitz, K. (1979). Operations on images using quad trees. *IEEE transactions on pattern analysis and machine intelligence*, 1:145–153. 2.3

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. 1.1

Iwahashi, N. (2003). Language acquisition through a human-robot interface by combining speech, visual, and behavioral information. *Information Sciences*, 156:109–121. 5

Izadi, S., Davison, A., Fitzgibbon, A., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., and Freeman, D. (2011). Kinect Fusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of the 24th annual ACM*

symposium on User interface software and technology - UIST '11, page 559. 4.3.1, 4.3.1

Johannsen, K. and De Ruiter, J. P. (2013). The role of scene type and priming in the processing and selection of a spatial frame of reference. *Frontiers in psychology*, 4:182. 5.4

Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:433–449. 4

Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. 1

Kaneko, K., Kanehiro, F., Morisawa, M., Akachi, K., Miyamori, G., Hayashi, A., and Kanehira, N. (2011). Humanoid robot HRP-4 - Humanoid robotics platform with lightweight and slim body. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4400–4407. 1.1

Kapoor, A. and Picard, R. W. (2005). Multimodal affect recognition in learning environments. *Proceedings of the 13th annual ACM international conference on Multimedia MULTIMEDIA 05*, page 677. 4

Kuo, W.-J., Tseng, S.-H., Yu, J.-Y., and Fu, L.-C. (2009). A hybrid approach to RBPF based SLAM with grid mapping enhanced by line matching. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1523–1528. IEEE. 3

Kuzmic, E. S. and Ude, A. (2010). Object segmentation and learning through feature grouping and manipulation. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 371–378. IEEE. 3.2.2, 3.3

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., Warmuth, M., and Wolf, P. (2003). The CMU SPHINX-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5. 5.3.5

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178. 4

Lee, I. and Kesner, R. P. (2003). Time-Dependent Relationship between the Dorsal Hippocampus and the Prefrontal Cortex in Spatial Memory. *J. Neurosci.*, 23(4):1517–1523. 3.1

Lehnert, G. and Zimmer, H. D. (2006). Auditory and visual spatial working memory. *Memory & Cognition*, 34(5):1080–1090. 3.1

Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. *Workshop on Statistical Learning in ...*, 4170(May):1–16. 4.1.1

Leonard, J. and Durrant-Whyte, H. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, volume 3, pages 1442–1447. Ieee. 3

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2548–2555. 4.2

Levinson, S. C. (1996). Frames of Reference and Molyneux's question: crosslinguistic evidence. In Bloom, P., Peterson, M., Nadel, L., and Garrett, M., editors, *Language and Space*, pages 109–169. MIT press, Cambridge, MA, 1st edition. 5.4

Levinson, S. C. (2003). *Space in Language and Cognition. Explorations in Cognitive Diversity*. Language Culture and Cognition. Cambridge University Press. 5.2, 5.2

Li, Q., Meier, M., Haschke, R., Ritter, H., and Bolder, B. (2012). Object dexterous manipulation in hand based on finite state machine. In *2012 IEEE International Conference on Mechatronics and Automation, ICMA 2012*, pages 1185–1190, Chengdu, China. 1.2

Li, S. and Wrede, B. (2007). Why and how to model multi-modal interaction for a mobile robot companion. In *Proc. AAAI Spring Symposium on Interaction Challenges for Intelligent Assistants, Stanford*, pages 71–79, Stanford. AAAI Press, AAAI Press. 5, 6

Li, X., Wang, L., and Sung, E. (2008). AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21:785–795. 4.2

Logan, G. D. and Sadler, D. D. (1996). A computational analysis of the apprehension of spatial relations. In *Language and Space*, pages 493–529. 5.2

Lohse, M., Siepmann, F., and Wachsmuth, S. (2013). A Modeling Framework for User-Driven Iterative Design of Autonomous Systems. *International Journal of Social Robotics*, 6(1):121–139. 1.4, 3.4.3

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110. 4, 4.2

Lütkebohle, I., Hegel, F., Schulz, S., Hackel, M., Wrede, B., Wachsmuth, S., and Sagerer, G. (2010). The Bielefeld anthropomorphic robot head "Flobi". In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3384–3391. 1.1

Ma, Y., Ju, H., and Cui, P. (2009). Research on Localization and Mapping for Lunar Rover Based on RBPF-SLAM. In *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 306–311. IEEE. 3

Martinez Mozos, O., Mizutani, H., Kurazume, R., and Hasegawa, T. (2012). Categorization of indoor places using the Kinect sensor. *Sensors (Basel, Switzerland)*, 12(5):6695–6711. 4, 4.1.1

Marton, Z., Pangercic, D., Blodow, N., Kleinehellefort, J., and Beetz, M. (2010). General 3D modelling of novel objects from a single view. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3700–3705. IEEE. 3.2.2

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157. 5.4.2

Meagher, D. (1982). Geometric modeling using octree encoding. 2.3

Meger, D., Muja, M., Helmer, S., Gupta, A., Gamroth, C., Hoffman, T., Baumann, M., Southey, T., Fazli, P., Wohlkinger, W., and Others (2010). Curious George: An Integrated Visual Search Platform. *Computer and Robot Vision (CRV), 2010 Canadian Conference on*, 0:107–114. 1.1

Meier, M., Schöpfer, M., Haschke, R., and Ritter, H. (2011). A probabilistic approach to tactile shape reconstruction. *IEEE Transactions on Robotics*, 27(3):630–635. 1.2

Metta, G., Natale, L., Nori, F., Sandini, G., Vernon, D., Fadiga, L., von Hofsten, C., Rosander, K., Lopes, M., Santos-Victor, J., Bernardino, A., and Montesano, L. (2010). The iCub humanoid robot: An open-systems platform for research in cognitive development. *Neural Networks*, 23:1125–1134. 1.1

Meyer Zu Borgsen, S., Schopfer, M., Ziegler, L., and Wachsmuth, S. (2014). Automated Door Detection with a 3D-Sensor. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 276–282. IEEE. 3.3.4, 3.5.1

Miller, G. A. and Johnson-Laird, P. (1976). *Language and Perception*. Belknap Press. 5.2

Montello, D. R. (1993). Scale and multiple psychologies of space. In Frank, A. U. and Campari, I., editors, *Spatial Information Theory A Theoretical Basis for GIS*, volume 716 of *Lecture Notes in Computer Science*, pages 312–321. Springer Berlin Heidelberg. 1.2

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: a factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, Edmonton, Canada. American Association for Artificial Intelligence. 3

Moratz, R., Tenbrink, T., Bateman, J., and Fischer, K. (2003). Spatial knowledge representation for human-robot interaction. *Spatial Cognition III*, 2685:263–286. 5, 5.3.5

Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74. 2.3, 3

Mou, W., McNamara, T. P., Rump, B., and Xiao, C. (2006). Roles of Egocentric and Allocentric Spatial Representations in Locomotion and Reorientation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(6):1274–1290. 3.1, 3.3.1

Mozos, O. M., Marton, Z. C., and Beetz, M. (2011). Furniture Models Learned from the WWW. *Robotics & Automation Magazine, IEEE*, 18(2):22–32. 4

Mukerjee, A. (1998). Neat Versus Scruffy: A Review of Computational Models for Spatial Expressions. In Oliver, P. and Gapp, K.-P., editors, *Representation and Processing of Spatial Expressions*, pages 1–35. L. Erlbaum Associates Inc. 5.2

Mundy, P. and Newell, L. (2007). Attention, joint attention, and social cognition. *Current Directions in Psychological Science*, 16(5):269–274. 5

Murphy, K. (1999). Bayesian Map Learning in Dynamic Environments. In *Advances in Neural Information Processing Systems NIPS*, volume 12, pages 1015 – 1021, Cambridge, MA. MIT Press. 3

Murphy, K., Torralba, A., and Freeman, W. (2003). Using the forest to see the trees: a graphical model relating features, objects and scenes. *Advances in Neural Information ...*, 53:107—-114. 4

Nalpantidis, L., Bjorkman, M., and Kragic, D. (2012). YES - YEt another object segmentation: Exploiting camera movement. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2116–2121. IEEE. 3.2.2

Nan, L., Xie, K., and Sharf, A. (2012). A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics*, 31(6):1. 3.2.2

Nguyen, V., Harati, A., and Siegwart, R. (2007). A lightweight SLAM algorithm using Orthogonal planes for indoor mobile robotics. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 658–663. IEEE. 3

Nielsen, J. (1993). *Usability Engineering*, volume 44. 1

Nieuwenhuisen, M., Droeschel, D., Holz, D., Stückler, J., Berner, A., Li, J., Klein, R., and Behnke, S. (2013). Mobile Bin Picking with an Anthropomorphic Service Robot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2327 – 2334, Karlsruhe. IEEE. 3.3

Nüchter, A. and Hertzberg, J. (2008). Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926. 3

Nüchter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2007). 6D SLAM—3D mapping outdoor environments: Research Articles. *Journal of Field Robotics*, 24(8-9):699–722. 3

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175. 3.1

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175. 4, 4.2, 4.3

Oliver, N., Garg, A., and Horvitz, E. (2004). Layered representations for learning and inferring office activity from multiple sensory channels. In *Computer Vision and Image Understanding*, volume 96, pages 163–180. 4

Opelt, A., Pinz, A., Fussenegger, M., and Auer, P. (2006). Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:416–431. 4

Palmer, t. E. (1975). The effects of contextual scenes on the identification of objects. 4

Pangercic, D., Haltakov, V., and Beetz, M. (2011). Fast and Robust Object Detection in Household Environments Using Vocabulary Trees with SIFT Descriptors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World*, San Francisco, CA, USA. 3.1

Payne, A. and Singh, S. (2005). Indoor vs. outdoor scene classification in digital photographs. *Pattern Recognition*, 38:1533–1545. 4

Perzanowski, D., Schultz, A. C., Adams, W., Marsh, E., and Bugajska, M. (2001). Building a multimodal human-robot interface. *IEEE Intelligent Systems and Their Applications*, 16(1):16–21. 5

Philippsen, R., Nejati, N., and Sentis, L. (2009). Bridging the Gap Between Semantic Planning and Continuous Control for Mobile Manipulation Using a Graph-Based World Representation. In Ferrein, A., Pauli, J., Siebel, N., and Steinbauer, G., editors, *Proceedings of the HYCAS 2009 workshop : 1st International Workshop on Hybrid Control of Autonomous Systems: Integrating Learning, Deliberation and Reactive Control*, pages 77–81, Pasadena, California, USA. 3

Pickering, M. J. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *The Behavioral and brain sciences*, 27(2):169–90; discussion 190–226. 1

Pronobis, A., Sjoo, K., Aydemir, A., Bishop, A., and Jensfelt, P. (2009). A framework for robust cognitive spatial mapping. In *2009 International Conference on Advanced Robotics*, pages 1–8, Munich. IEEE. 1.1

Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 413–420. 4

Quigley, M., Gerkey, B., Conley, K., Josh Faust, Foote, T., Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng (2009). ROS: An Open-Source Robot Operating System. In *ICRA workshop on open source software*, page 5. 1.1

Rensink, R. A. (2002). Change Detection. *Annual Review of Psychology*, 53:245–277. 3.2.3

Rickert, M., Foster, M. E., Giuliani, M., By, T., Panin, G., and Knoll, A. (2007). Integrating language, vision and action for human robot dialog systems. *Universal Access in HumanComputer Interaction Ambient Interaction*, 4555:987–995. 5

Röhlig, T. (2013). *Indoor Room Categorization using Boosted 2D and 3D Features*. Master thesis, Bielefeld University. 4.3

Roskos-Ewoldsen, B., McNamara, T. P., Shelton, A. L., and Carr, W. (1998). Mental representations of large and small spatial layouts are orientation dependent. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(1):215–226. 3.1

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571. 4.2

Ruetschi, U.-J. and Timpf, S. (2005). Modelling Wayfinding in Public Transport: Network Space and Scene Space. *Spatial Cognition IV. Reasoning, Action, Interaction*, 3343:24–41. 1.2

Rusu, R., Marton, Z., Blodow, N., Holzbach, A., and Beetz, M. (2009a). Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3601–3608. IEEE. 3.2.2

Rusu, R. B., Blodow, N., and Beetz, M. (2009b). Fast Point Feature Histograms (FPFH) for 3D registration. In *Robotics and Automation, 2009. ICRA '09. IEEE Int. Conf. on*, pages 3212–3217. 4, 4.2

Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2009c). Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–6. IEEE. 3, 3.2.2

Rusu, R. B., Bradski, G., Thibaux, R., and Hsu, J. (2010). Fast 3D recognition and pose using the Viewpoint Feature Histogram. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2155–2162. IEEE. 4

Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE. 3.4.2, 5.3.1

Rusu, R. B. R., Holzbach, A., Beetz, M., and Bradski, G. (2009d). Detecting and segmenting objects for mobile manipulation. *Vision Workshops (ICCV*, pages 47–54. 1.1

Sacks, H., Schegloff, E. A., and Jefferson, G. (1974). A Simplest Systematics for the Organization of Turn-Taking for Conversation. *Language*, 50(4):696–735. 1

Salti, S., Tombari, F., and Di Stefano, L. (2010). On the Use of Implicit Shape Models for Recognition of Object Categories in 3D Data. In Kimmel, R., Klette, R., and Sugimoto, A., editors, *Computer Vision – ACCV 2010*, volume 6494 of *Lecture Notes in Computer Science*, pages 653–666. Springer Berlin Heidelberg. 4.1.1, 4.1.1, 4.1.3, 4.1.3, 4.1.3

Sanders, B. C. S., Nelson, R. C., and Sukthankar, R. (2002). A theory of the quasi-static world. In *Pattern Recognition, 2002. Proceedings. 16th Int. Conf. on*, volume 3, pages 1–6 vol.3. 3.1

Schaal, S. (2007). The New Robotics-towards human-centered machines. *HFSP journal*, 1(2):115–26. 1

Schauerte, B. and Fink, G. a. (2010). Focusing computational visual attention in multi-modal human-robot interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction on - ICMI-MLMI '10*, page 1, New York, NY, USA. ACM. 5

Schmidt, J., Hofemann, N., and Haasch, A. (2008). Interacting with a mobile robot: Evaluating gestural object references. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3804 – 3809, Nice, France. IEEE. 5

Schwenk, H. and Bengio, Y. (1997). Adaptive Boosting of Neural Networks for Character Recognition. *Neural Networks*, pages 1–9. 4.2

Serrano, N., Savakis, A., and Luo, A. (2002). A computationally efficient approach to indoor/outdoor scene classification. *Object recognition supported by user interaction for service robots*, 4. 4

Sheikh, Y., Javed, O., and Kanade, T. (2009). Background Subtraction for Freely Moving Cameras. In *Computer Vision, 2009 IEEE 12th Int. Conf. on*, pages 1219–1225. 3.1

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. 3.4.1

Siepmann, F. and Wachsmuth, S. (2011). A Modeling Framework for Reusable Social Behavior. In De Silva, R. and Reidsma, D., editors, *Work in Progress Workshop Proceedings ICSR 2011*, pages 93–96, Amsterdam. Springer. 1.4

Siepmann, F., Ziegler, L., Kortkamp, M., and Wachsmuth, S. (2014). Deploying a Modeling Framework for Reusable Robot Behavior to Enable Informed Strategies for Domestic Service Robots. *Robotics and Autonomous Systems*, 62(5):619–631. 3.4.2, 3.4.3, 4.2, 4.2.2

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor Segmentation and Support Inference from RGBD Images. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *ECCV*, pages 1–14. Springer Berlin Heidelberg. 4.1.3

Silberman, N., Sontag, D., and Fergus, R. (2014). Instance Segmentation of Indoor Scenes Using a Coverage Loss. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, volume 8689 of *Lecture Notes in Computer Science*, pages 616–631. Springer International Publishing, Cham. 3.2.2

Sjoo, K., Zender, H., Jensfelt, P., Kruijff, G., Pronobis, A., Hawes, N., and Brenner, M. (2010). *The Explorer System*, volume 8 of *Cognitive Systems Monographs*, page 395. Springer Verlag. 1.1

Somanath, G. and Kambhamettu, C. (2011). Abstraction and Generalization of 3D Structure for Recognition in Large Intra-Class Variation. In Kimmel, R., Klette, R., and Sugimoto, A., editors, *Computer Vision – ACCV 2010*, volume 6494 of *Lecture Notes in Computer Science*, pages 483–496. Springer Berlin Heidelberg. 4

Squire, L. R. (1992). Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychological Review*, 99(2):195–231. 3.1

Stark, M. (1996). Impairment of an Egocentric Map of Locations: Implications for Perception and Action. *Cognitive Neuropsychology*, 13(4):481–524. 3.1

Steder, B., Grisetti, G., Van Loock, M., and Burgard, W. (2009). Robust on-line model-based object detection from range images. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ Int. Conf. on*, pages 4739–4744. 3.1

Stein, J. F. (1989). Representation of egocentric space in the posterior parietal cortex. *Quarterly journal of experimental physiology (Cambridge, England)*, 74(5):583–606. 3.1

Stiefelhagen, R., Ekenel, H. K., Fügen, C., Gieselmann, P., Holzapfel, H., Kraft, F., Nickel, K., Voit, M., and Waibel, A. (2007). Enabling multimodal human-robot interaction for the Karlsruhe humanoid robot. *Robotics, IEEE Transactions on*, 23(5):840–851. 5

Strom, J., Richardson, A., and Olson, E. (2010). Graph-based segmentation for colored 3D laser point clouds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2131–2136. IEEE. 3.2.2

Stückler, J. and Behnke, S. (2011). Improving people awareness of service robots by semantic scene knowledge. *RoboCup 2010: Robot Soccer World Cup XIV*, pages 157–168. 1.1

Stückler, J. and Behnke, S. (2014). Multi-resolution surfel maps for efficient dense 3D modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147. 1.1

Stückler, J., Droeschel, D., Gräve, K., Holz, D., Schreiber, M., Topalidou-Kyniazopoulou, A., Schwarz, M., and Behnke, S. (2014). Increasing Flexibility of Mobile Manipulation and Intuitive Human-Robot Interaction in RoboCup@Home. 1.1

Sturm, J., Konolige, K., Stachniss, C., and Burgard, W. (2010). Vision-based detection for learning articulation models of cabinet doors and drawers in household environments. In *Robotics and Automation (ICRA), 2010 IEEE Int. Conf. on*, pages 362–368, Anchorage, AK. IEEE. 3.1

Swadzba, A. (2011). *The robot's vista space: a computational 3D scene analysis*. dissertation, Bielefeld University. 1.2, 3.5.1, 4.3

Swadzba, A., Beuter, N., Wachsmuth, S., and Kummert, F. (2010). Dynamic 3D scene analysis for acquiring articulated scene models. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, volume 1, pages 134–141, Anchorage, AK, USA. IEEE, IEEE. 3.2, 3.2.3, 3.6

Swadzba, A. and Wachsmuth, S. (2008). Categorizing perceptions of indoor rooms using 3d features. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 734–744. 4

Swadzba, A. and Wachsmuth, S. (2011). Indoor scene classification using combined 3d and gist features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6493 LNCS, pages 201–215. 4, 4.2, 4.3, 4.3.3, 4.3.3, 4.3.3, 4.4

Swadzba, A., Wachsmuth, S., Vorwerg, C., and Rickheit, G. (2009). A computational model for the alignment of hierarchical scene representations in human-robot interaction. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1857–1863, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 5.5

Szummer, M. and Picard, R. (1998). Indoor-outdoor image classification. *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*. 4

Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 321—-328, San Francisco, CA. IEEE. 3

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler,

A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692. 1, 1.1

Tombari, F. and Di Stefano, L. (2010). Object recognition in 3D scenes with occlusions and clutter by Hough voting. In *Proceedings - 4th Pacific-Rim Symposium on Image and Video Technology, PSIVT 2010*, pages 349–355. 4.1.2

Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, ECCV'10, pages 356–369, Berlin, Heidelberg. Springer-Verlag. 4, 4.1.1, 4.1.1, 4.2

Tombari, F., Salti, S., and Di Stefano, L. (2011). A combined texture-shape descriptor for enhanced 3D feature matching. In *Proceedings - International Conference on Image Processing, ICIP*, pages 809–812. 4.2

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision*, 53:169–191. 4, 4, 4.3

Torralba, A., Murphy, K., Freeman, W., and Rubin, M. (2003). Context-based vision system for place and object recognition. *Proceedings Ninth IEEE International Conference on Computer Vision*. 4

Treptow, A. and Zell, A. (2004). Combining Adaboost Learning and Evolutionary Search to Select Features for Real-Time Object Detection. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 2107–2113, Portland, Oregon. IEEE Press. 4

Uckermann, A., Haschke, R., and Ritter, H. (2013). Realtime 3D segmentation for human-robot interaction. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2136–2143. IEEE. 3.2.2, 3.5.1

van Dijk, T. A. and Kintsch, W. (1983). *Strategies of Discourse Comprehension*. Academic Press, Inc. 1

Vasudevan, S., Gächter, S., Nguyen, V., Siegwart, R., and Gachter, S. (2007). Cognitive maps for mobile robots – an object based approach. *Robotics and Autonomous Systems*, 55:359–371. 3, 3.3

Viola, P. and Jones, M. (2001). Robust real-time object detection. *International Journal of Computer Vision*, 57:137–154. 4

Viswanathan, P., Southey, T., Little, J., and Mackworth, A. (2010). Automated Place Classification Using Object Detection. *Computer and Robot Vision (CRV), 2010 Canadian Conference on*. 4

Vogel, J. and Schiele, B. (2004). A Semantic Typicality Measure for Natural Scene Categorization. *Cognitive Psychology*, 3175:195–203. 4

Wachsmuth, I., Kopp, J. d. R., Jaecks, P., and Stefan (2013). *Alignment in Communication: Towards a new theory of communication,*. Advances in Interaction Studies. Benjamins, Amsterdam. 1

Wachsmuth, S., Fink, G. A., and Sagerer, G. (1998). Integration of parsing and incremental speech recognition. In *Proceedings of the European Signal Processing Conference*, pages 371–375, Rhodes, Greece. 1.1

Wada, K. and Shibata, T. (2007). Living with seal robots - Its sociopsychological and physiological influences on the elderly at a care house. In *IEEE Transactions on Robotics*, volume 23, pages 972–980. 1.1

Wang, R. F. and Simons, D. J. (1999). Active and passive scene recognition across views. *Cognition*, 70(2):191–210. 3.1

Wang, R. F. and Spelke, E. S. (2000). Updating egocentric representations in human navigation. *Cognition*, 77(3):215–250. 3.1

Wiemann, T. (2013). Automatic Generation of 3D Polygon Maps for Mobile Robots. *KI - Künstliche Intelligenz*, 28(1):53–57. 3

Wienke, J. and Wrede, S. (2011). A middleware for collaborative research in experimental robotics. In *2011 IEEE/SICE International Symposium on System Integration, SII 2011*, pages 1183–1190. 1.4

Wilcoxon, F. (1946). Individual comparisons of grouped data by ranking methods. *Journal of economic entomology*, 39:269. 5.4.2

Wisspeintner, T., van der Zant, T., Iocchi, L., and Schiffer, S. (2009). RoboCup@Home: Scientific Competition and Benchmarking for Domestic Service Robots. *Interaction Studies*, 10(3):392–426. 1.1

Wittrowski, J., Ziegler, L., and Swadzba, A. (2013). 3D Implicit Shape Models Using Ray Based Hough Voting for Furniture Recognition. In *2013 International Conference on 3D Vision*, pages 366–373. IEEE. 4.1

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE. 1.1

Zender, H., Martínez Mozos, O., Jensfelt, P., Kruijff, G.-J., and Burgard, W. (2008). Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502. 3, 3.3

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class AdaBoost. *Statistics and Its Interface*, 2(3):349–360. 4.2.1, 4.2.1

Ziegler, L. (2010). *Developing a Vision-Based Object Search Behavior for a Mobile Robot*. Master's thesis, Bielefeld University. 2.3, 3.1

Ziegler, L., Johannsen, K., Swadzba, A., De Ruiter, J. P., and Wachsmuth, S. (2012). Exploiting spatial descriptions in visual scene analysis. *Cognitive processing*, 13 Suppl 1:369–374. 5.2, 5.2.1

Zwaan, R. A. and Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological bulletin*, 123(2):162–185. 1, 6

Appendices

Appendix A

Situation cases for ASM evaluation

Performance analysis ASM

- CASE-A0 (Setting: SE, TA, CA, KI1, KI2, DO, SO)
  - Object is being moved
- CASE-A1 (Setting: SE, TA, CA)
  - Object is moved only slightly
- CASE-A2 (Setting: SE, TA, CA)
  - Very small object is moved
- CASE-A3 (Setting: TA)
  - Tall object is placed in front of unknown area
- CASE-A4 (Setting: SE, TA, CA)
  - Two movable objects close together

Naiive matching analysis NM-ASM

- CASE-N0 (Setting: SE)
  1. one object is being moved (visible)
  2. object is at same place
- CASE-N1 (Setting: TA)
  1. raw scene (no change)
  2. previously hidden object is in front known background

Performance analysis MV-ASM

- CASE-M0 (Setting: SE, TA, CA)
  1. multiple objects are moved (visible)
  2. objects are at same place

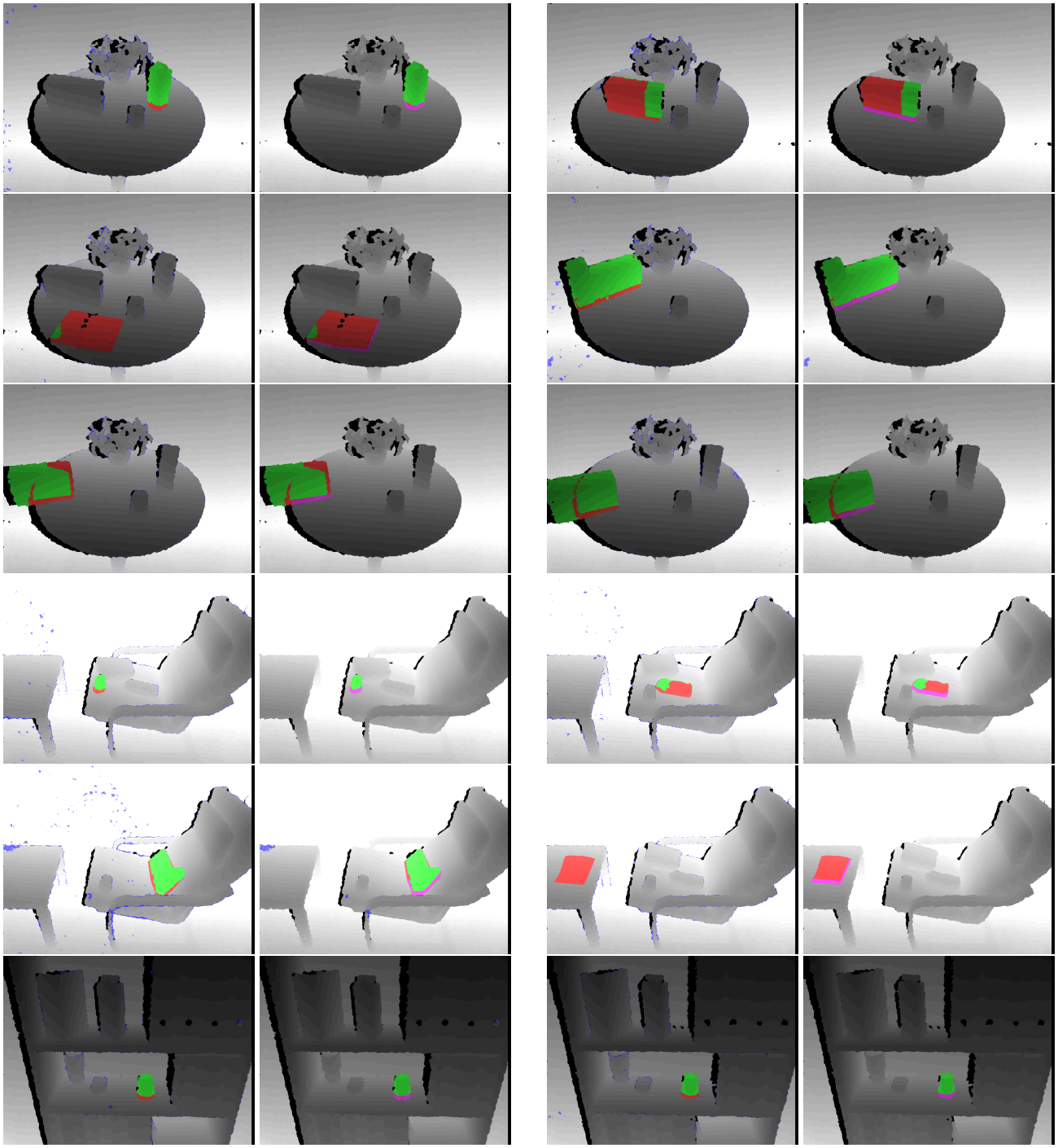
- CASE-M1 (Setting: SE, TA, CA)
  1. raw scene (no change)
  2. previously hidden object is in front known background
- CASE-M2 (Setting: SE, TA, CA)
  1. one object is beeing moved and stacked
  2. object is at same place
- CASE-M3 (Setting: SE, TA, CA)
  1. two objects are moved and stacked
  2. objects are at same place
- CASE-M4 (Setting: SE, TA, CA)
  1. one object is moved to occlusion
  2. object is at same place (visible)
- CASE-M5 (Setting: SE, TA, CA)
  1. raw scene (no change)
  2. object was added in meantime
- CASE-M6 (Setting: SE, TA, CA)
  1. raw scene (no change)
  2. object was removed in meantime
- CASE-M7 (Setting: SE, TA, CA)
  1. raw scene (no change)
  2. object was added in meantime at a place that was occluded before
- CASE-M8 (Setting: TA)
  1. raw scene (no change)
  2. tall object was added in meantime at a place with unknown background
- CASE-M9 (Setting: TA)
  1. raw scene (no change)
  2. object moved to occlusion (was visible before)
  3. object is at same place (visible)

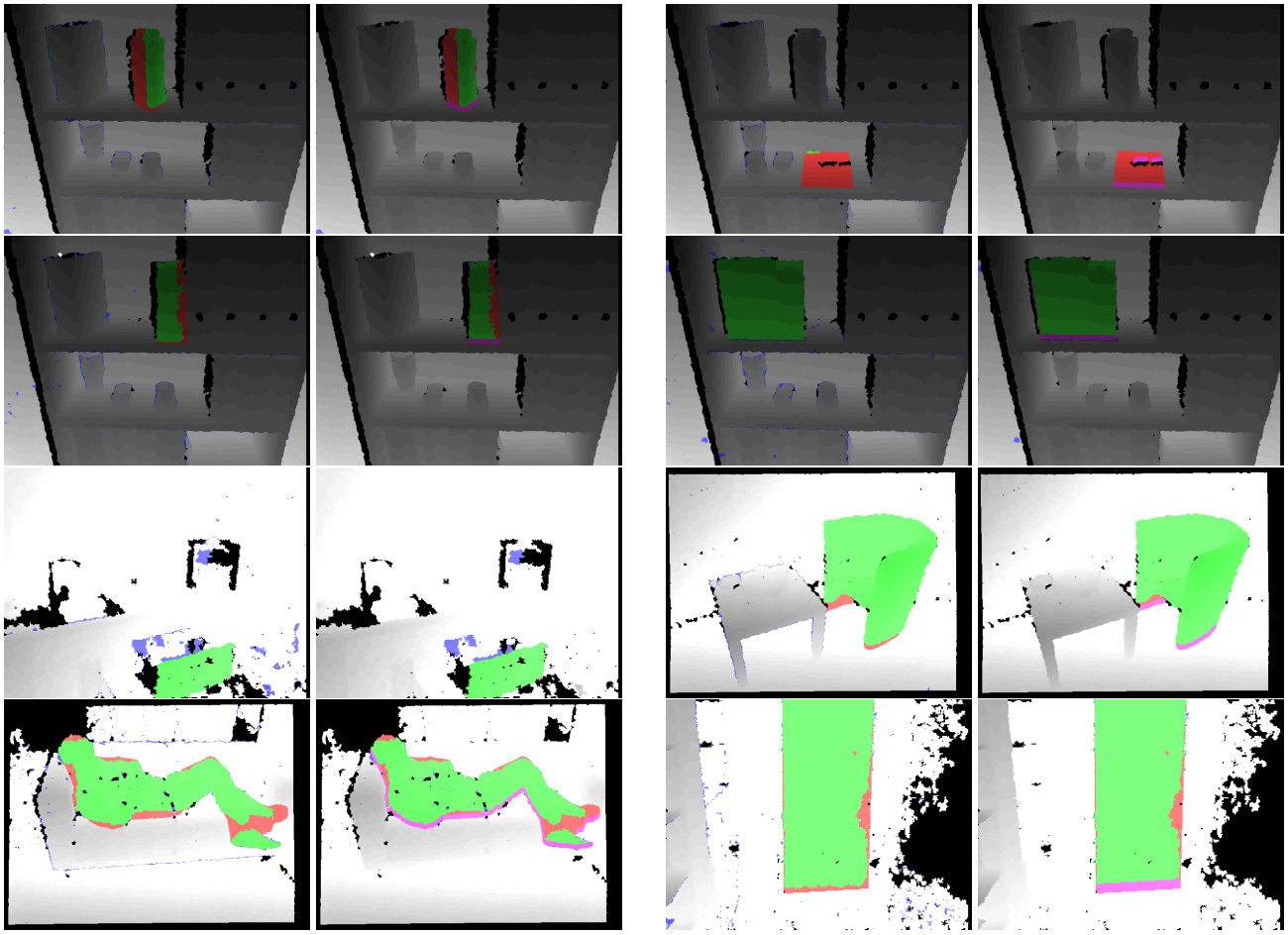
Appendix B

Results from Multi-View ASM Evaluation

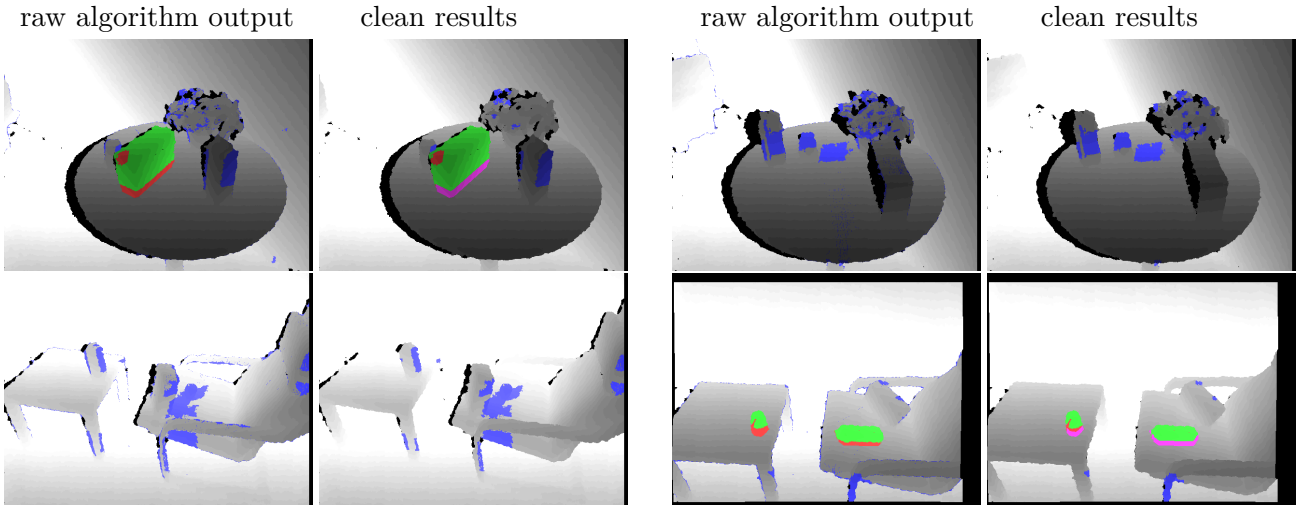
B.1. Evaluation of simple ASM

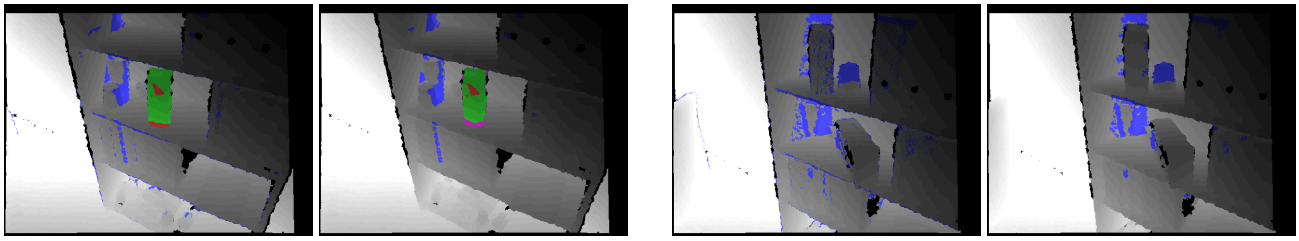
raw algorithm output clean results raw algorithm output clean results





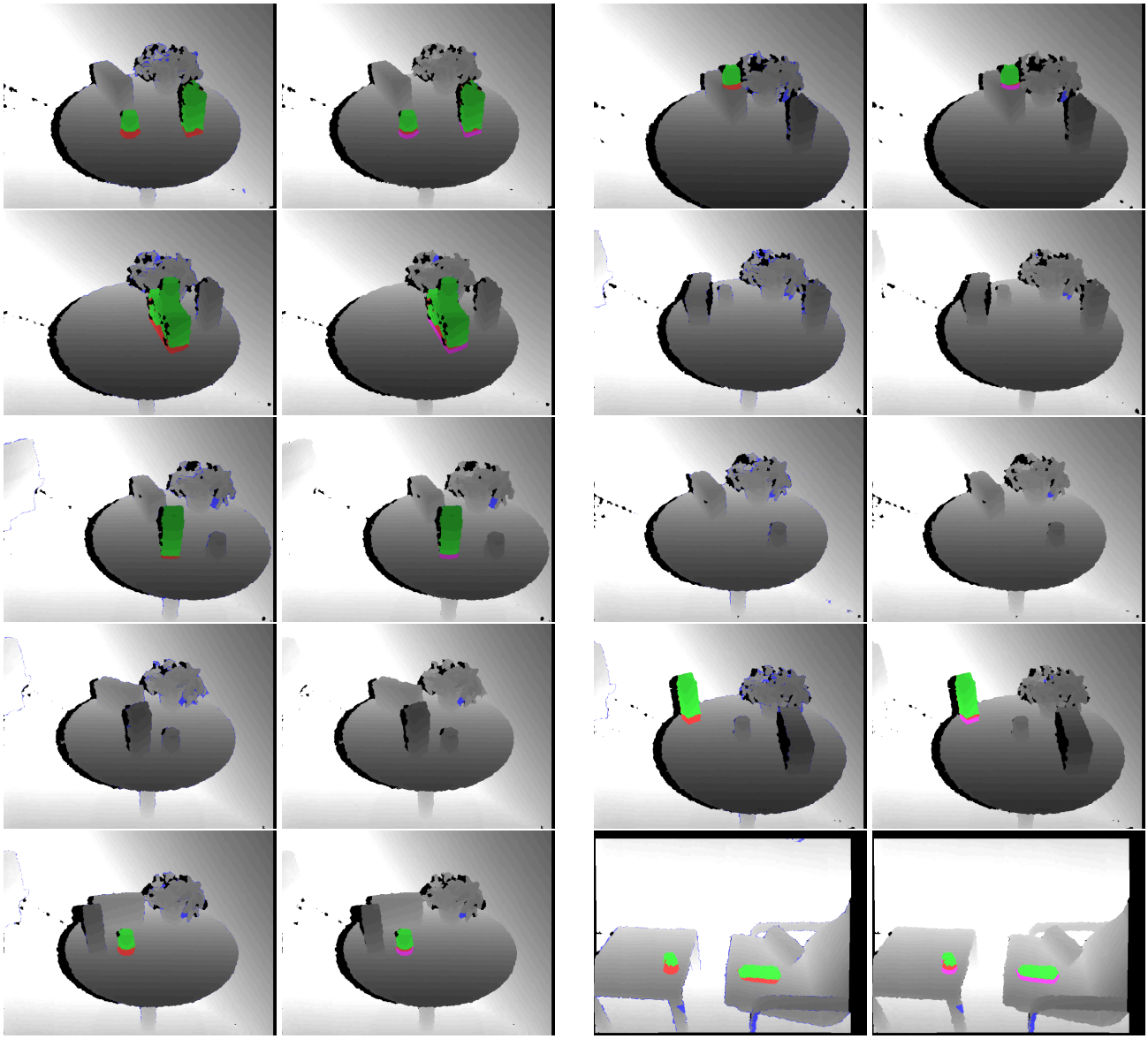
B.2. Evaluation of naive matching ASM

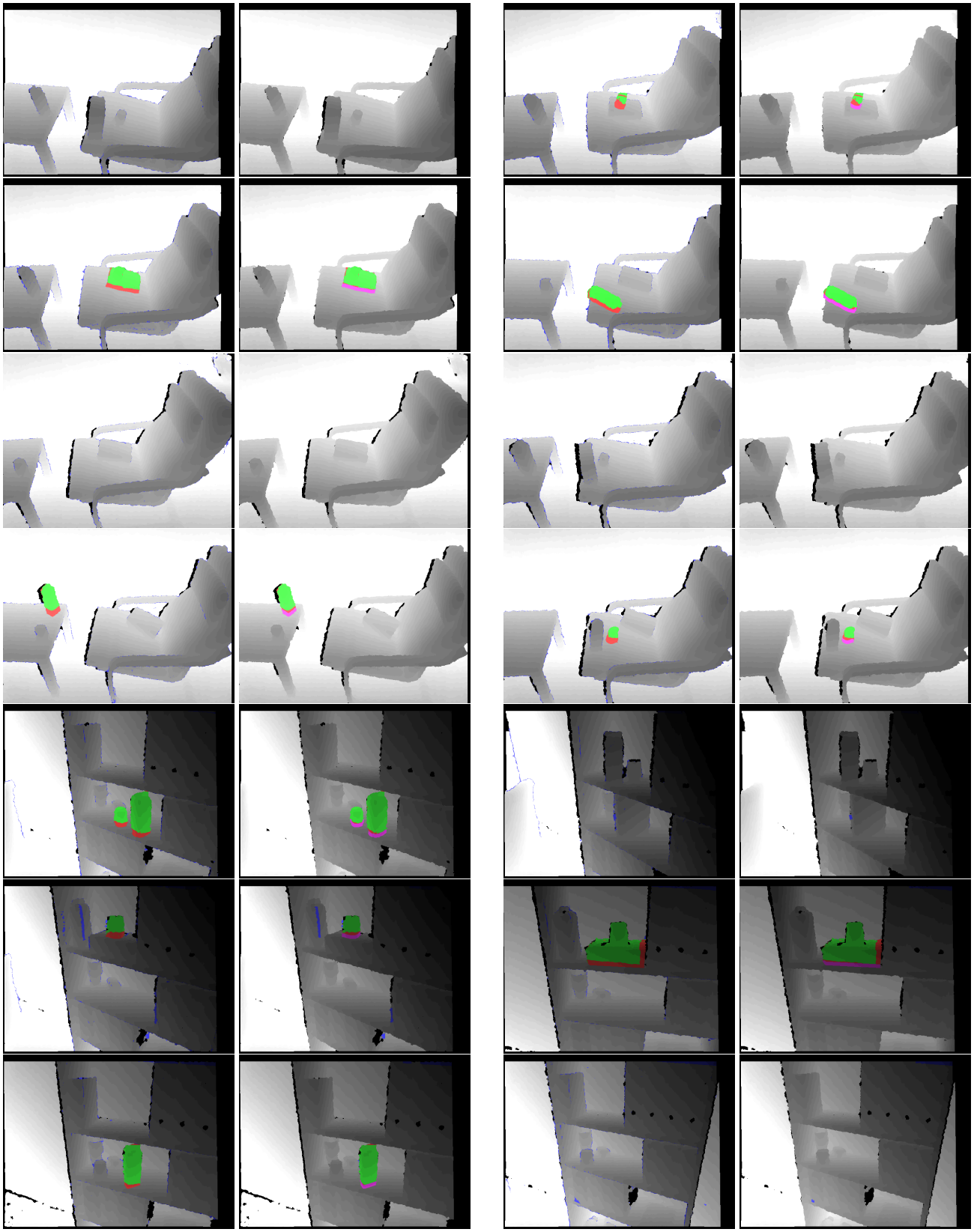


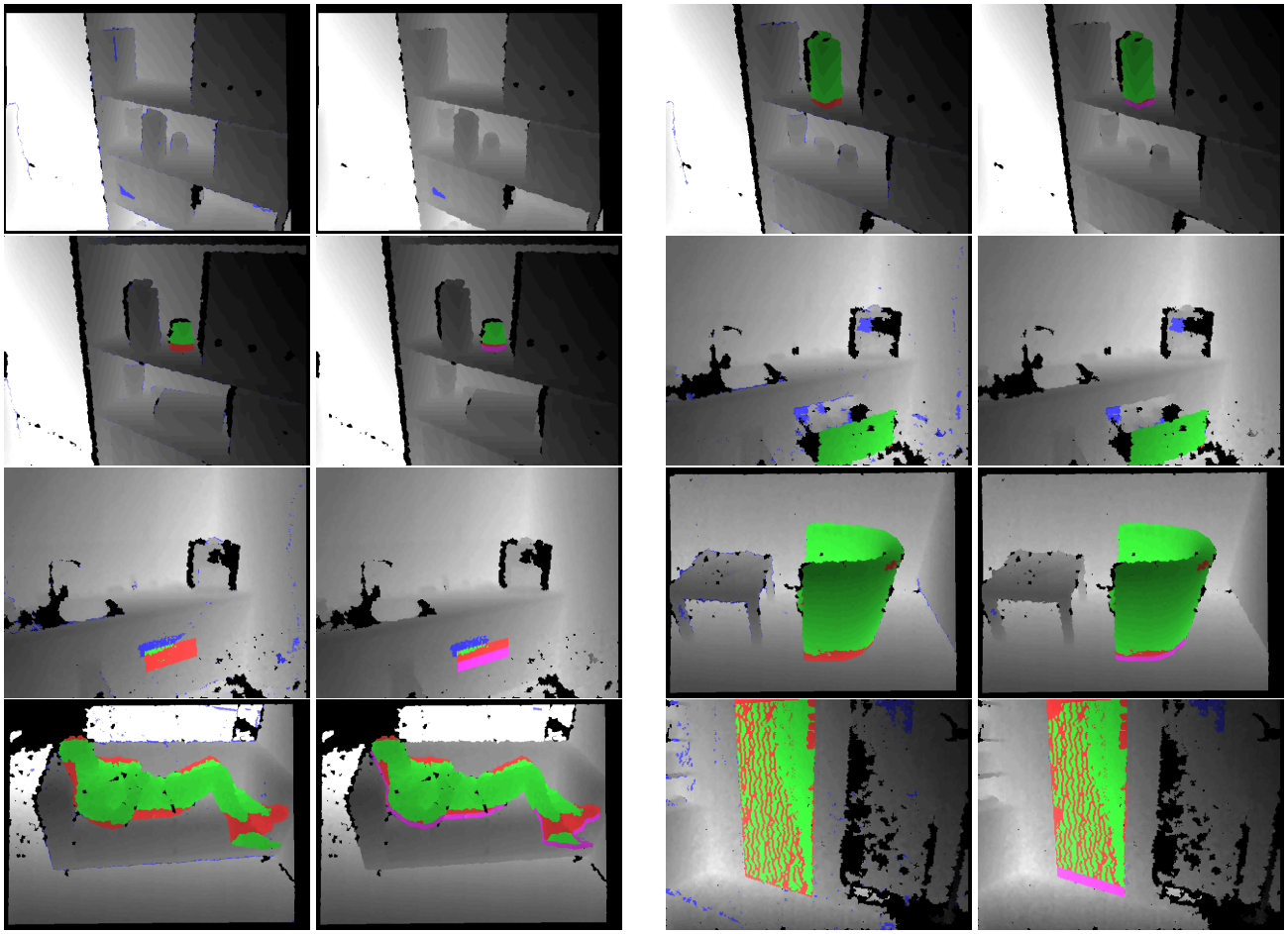


B.3. Evaluation of multi-view ASM

raw algorithm output clean results raw algorithm output clean results







Appendix C

Models for 3D ISM Training

The following models from Princeton Shape Database were used for training the ISMs:

Appendix D

Results from Evaluation of Household Object Classification

	unknown	tea	paper cup	cup1	duster	wall mount	tape	milk	coke	cup2	book	stapler	sponge	cup0	hair gel	joy pad
tea	0.889			0.044			0.022			0.022				0.022		
paper cup		0.771	0.057				0.029		0.086		0.029				0.029	
cup1		0.017	0.833				0.033		0.100		0.017					
duster				0.986											0.014	
wall mount	0.060				0.840		0.020			0.020				0.040	0.020	
tape	0.018		0.018			0.782	0.018	0.018	0.018		0.036	0.055		0.018	0.018	
milk	0.020						0.900	0.020	0.040					0.020		
coke			0.022			0.022	0.111	0.778	0.022	0.022		0.022				
cup2	0.033		0.033						0.917	0.017						
book	0.022	0.022	0.022		0.022		0.022	0.044		0.778				0.044	0.022	
stapler		0.014	0.014	0.014	0.014	0.014				0.029	0.886	0.014				
sponge				0.018		0.018					0.073	0.873		0.018		
cup0						0.022				0.044	0.022		0.911			
hair gel	0.040	0.040		0.020	0.020		0.020	0.060		0.080	0.020			0.660	0.040	
joy pad							0.022	0.022						0.022	0.933	

Table D.1.: Confusion matrix of the E-SAMME-2D configuration. Rows: categories tested. Columns: classification results.

	tea	paper cup	cup1	wall mount	milk	coke	cup2	book	hair gel
tea	0.889	0.022		0.044	0.022	0.022			
paper cup		0.800	0.029			0.057	0.086		0.029
cup1		0.050	0.900	0.017		0.017	0.017		
wall mount	0.040		0.020	0.900		0.020			0.020
milk					0.920	0.040			0.040
coke		0.025	0.025	0.025	0.100	0.750		0.075	
cup2			0.083		0.017		0.867	0.033	
book	0.022	0.044	0.067	0.022	0.022	0.022		0.800	
hair gel				0.080	0.120	0.040		0.020	0.740

Table D.2.: Confusion matrix of the E-SAMME-OBJ-T configuration. Rows: categories tested. Columns: classification results.

	duster	tape	stapler	sponge	cup0	joy pad
duster	0.986		0.014			
tape	0.036	0.727	0.036	0.055	0.091	0.055
stapler	0.029		0.800	0.114	0.043	0.014
sponge	0.036	0.091	0.091	0.745		0.036
cup0		0.022	0.022		0.889	0.067
joy pad		0.133				0.867

Table D.3.: Confusion matrix of the E-SAMME-OBJ-S configuration. Rows: categories tested. Columns: classification results.

	tea	paper cup	cup1	duster	wall mount	tape	milk	coke	cup2	book	stapler	sponge	cup0	hair gel	joy pad
tea	0.889	0.022		0.022			0.067								
paper cup		0.743	0.057				0.029	0.029	0.029		0.057		0.029	0.029	
cup1		0.033	0.850				0.017		0.050	0.033	0.017				
duster	0.043			0.957											
wall mount	0.020		0.020		0.920		0.020							0.020	
tape			0.018	0.018		0.764		0.018	0.018			0.055	0.018	0.055	0.036
milk	0.040	0.020	0.020		0.020		0.860	0.020		0.020					
coke		0.044				0.044	0.111	0.756		0.022			0.022		
cup2	0.017	0.033	0.050					0.033	0.850	0.017					
book	0.022	0.044	0.022		0.022		0.044	0.067		0.733	0.022	0.022			
stapler		0.014	0.029	0.014		0.029			0.029	0.029	0.843	0.014			
sponge		0.018	0.055	0.036		0.055		0.018		0.018	0.055	0.745			
cup0					0.022		0.022		0.022	0.022		0.911			
hair gel	0.040	0.020	0.020	0.020		0.060	0.020	0.040		0.100	0.040		0.020	0.580	0.040
joy pad		0.022		0.089		0.044	0.022	0.067						0.022	0.733

Table D.4.: Confusion matrix of the SVM-SURF configuration. Rows: categories tested. Columns: classification results.

Appendix E

Questionnaire for RSM evaluation

see next page

Fragebogen
Evaluationsstudie „Room Structure Model“

Leon Ziegler / SFB 673 Teilprojekt A4

Dieser Fragebogen dient zusammen mit den aufgezeichneten Daten zur Auswertung der Studie. Es gelten die Richtlinien, die in der *Einwilligungserklärung für Video- und Tonaufnahmen – Evaluationsstudie „Room Structure Model“* festgehalten wurden.

Identitätsnummer: _____

Alter: _____

Geschlecht: _____

Wie gut schätzen Sie ihre Kenntnis über **Computer** ein?

geringe Kenntnis ☐ ☐ ☐ ☐ ☐ große Kenntnis

Wie gut schätzen Sie ihre Kenntnis über **Softwareentwicklung** ein?

geringe Kenntnis ☐ ☐ ☐ ☐ ☐ große Kenntnis

Wie gut schätzen Sie ihre Kenntnis über **Roboter** ein?

geringe Kenntnis ☐ ☐ ☐ ☐ ☐ große Kenntnis

Wie gut schätzen Sie ihre Kenntnis über **räumliche Kognition** von Menschen ein?

geringe Kenntnis ☐ ☐ ☐ ☐ ☐ große Kenntnis

Wie gut schätzen Sie ihre Kenntnis über **Softwaresysteme** ein, die versuchen **räumliche Kognition** von Menschen zu analysieren oder nachzubilden?

geringe Kenntnis ☐ ☐ ☐ ☐ ☐ große Kenntnis