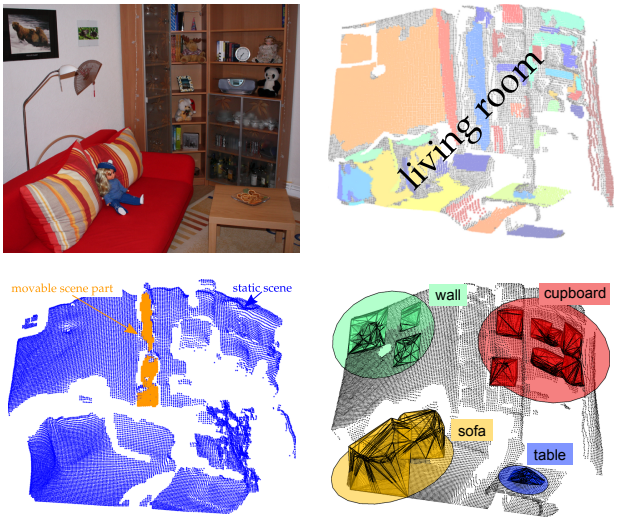
AGNES SWADZBA

THE ROBOT'S VISTA SPACE – A COMPUTATIONAL ANALYSIS

Universität Bielefeld

THE ROBOT'S VISTA SPACE

A COMPUTATIONAL 3D SCENE ANALYSIS



Dissertation zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Technischen Fakultät der Universität Bielefeld

vorgelegt von

AGNES SWADZBA

DIPL.-INF. AGNES SWADZBA
Applied Informatics
Faculty of Technology
Bielefeld University
aswadzba@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
das akademischen Grades Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Technischen Fakultät der Universität Bielefeld,
vorgelegt am 26.01.2011,
verteidigt am 23.03.2011.

GUTACHTER:

PD Dr. Sven Wachsmuth, Universität Bielefeld
Prof. Dr. Jim Little, University of British Columbia
Prof. Dr. Christian Wöhler, Technische Universität Dortmund

PRÜFUNGSAUSSCHUSS:

Prof. Dr. Mario Botsch, Universität Bielefeld
PD Dr. Sven Wachsmuth, Universität Bielefeld
Prof. Dr. Jim Little, University of British Columbia
Dr. Hendrik Koesling, Universität Bielefeld

Gedruckt auf alterungbeständigem Papier nach ISO 9706.

ACKNOWLEDGMENTS

At this point, I would like to take the opportunity to thank for all the support I have experienced. First of all, I would like to thank Sven for all the fruitful discussions and his confidence in my various ideas and explorations. I would like to thank Jim Little for his time spent on reading my thesis and joining my defense and the encouraging feedback on my work. Last but not least, I thank Christian Wöhler that he could review my thesis within one week allowing me to keep the 23.03.2011 as date for my defense.

I want to thank the Applied Informatics group, especially, Gerhard Sagerer for inviting me to Bielefeld which resulted in the opportunity to work in the Collaborative Research Center 673 "Alignment in Communication" and to travel to many interesting conferences and meetings. When studying Computer Science in Erlangen I have had no idea that Bielefeld can offer such a wonderful workplace. I would like to thank Niklas Beuter for being open-minded about all my ideas how to combine our work on 3D data processing. I thank Frederic Siepmann and my student helper Christian Thöns for supporting the integration of some of my work on our robot BIRON and Marco Kortkamp, Julia Peltason, Frederic Siepmann, and Marko Tscherepanow for their comments to this thesis improving its readability. I thank all the members of AI and CLF for creating such a nice working and socializing atmosphere. Altogether, I thank the CRC 673 for the environment inspiring interdisciplinary research and my colleagues Constanze Vorwerg and Gert Rickheit from the A4 project "Alignment of Situation Models" for introducing me to the linguistic perspective of my work. In memory of Gert who died suddenly in April I feel honored that as one of his last research activities he has attended my defense.

Last, I would like to thank my family, especially, my mother for her effort to make my transition from Poland to Germany as a 6-year-old child as smooth as possible. Her support during the first year at school and the confidence of my primary school teacher in my capabilities has contributed to the fact that I have not lost a year at school. I thank my parents and my brother for encouraging me to follow my way. I thank Hans for his love, support, and patience in listening to my problems. Hans, thanks for complementing my view on life. I hope I can do the same for you. It's not always easy but we will make it!

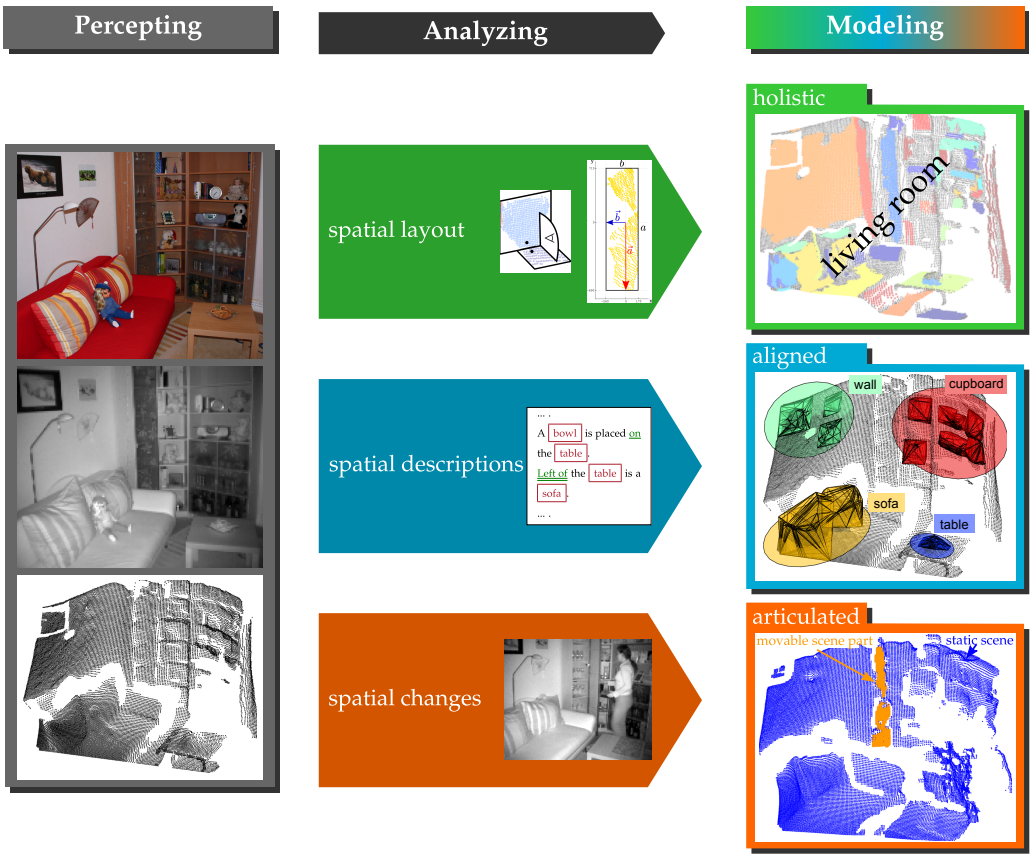
Agnes Swadzba
Bielefeld, May 2011

ABSTRACT

The space that can be explored quickly from a fixed view point *without locomotion* is known as the *vista space*. In indoor environments single rooms and room parts follow this definition. The vista space plays an important role in situations with agent-agent interaction as it is the directly surrounding environment in which the interaction takes place. A collaborative interaction of the partners in and with the environment requires that both partners know where they are, what spatial structures they are talking about, and what scene elements they are going to manipulate. This thesis focuses on the analysis of a robot's vista space. Mechanisms for extracting relevant spatial information are developed which enable the robot to recognize in which place it is, to detect the scene elements the human partner is talking about, and to segment scene structures the human is changing. These abilities are addressed by the proposed holistic, aligned, and articulated modeling approach. For a smooth human-robot interaction, the computed models should be aligned to the partner's representations. Therefore, the design of the computational models is based on the combination of psychological results from studies on human scene perception with basic physical properties of the perceived scene and the perception itself. The *holistic modeling* realizes a categorization of room percepts based on the observed 3D spatial layout. Room layouts have room type specific features and fMRI studies have shown that some of the human brain areas being active in scene recognition are sensitive to the 3D geometry of a room. With *the aligned modeling*, the robot is able to extract the hierarchical scene representation underlying a scene description given by a human tutor. Furthermore, it is able to ground the inferred scene elements in its own visual perception of the scene. This modeling follows the assumption that cognition and language schematize the world in the same way. This is visible in the fact that a scene depiction mainly consists of relations between an object and its supporting structure or between objects located on the same supporting structure. Last, the *articulated modeling* equips the robot with a methodology for articulated scene part extraction and fast background learning under short and disturbed observation conditions typical for human-robot interaction scenarios. Articulated scene parts are detected model-less by observing scene changes caused by their manipulation. Change detection and background learning are closely coupled because change is defined phenomenologically as variation of structure. This means that change detection involves a comparison of currently visible structures with a representation in memory. In range sensing this comparison can be nicely implement as subtraction of these two representations. The three modeling approaches enable the robot to enrich its visual perceptions of the surrounding environment, the vista space, with semantic information about meaningful spatial structures useful for further interaction with the environment and the human partner.

CONTENTS

1 INTRODUCTION



Spatial awareness and the ability to communicate about the environment are key capabilities enabling an agent to perform day-to-day navigation tasks. As navigation is essential for agents, this explains the development and importance of a spatial language [Sku04]. In general, space can be partitioned along the actions that are required to perceive it [Kui00, Mon93]. For example, locomotion is needed to record data about the *large-scale* space. While, the *vista* space can be explored quickly from a single view point by eventually moving the gaze. Applying this definition to domestic environments, a complete apartment has the dimension of a large-scale space. Percepts of single rooms or room parts can be assigned to the vista space. This distinction of space can also be applied to the perception of a robot. Since the ability of navigation and localization is essential for a mobile robot, much research has concentrated on approaches for modeling the robot's large-scale space. Especially, algorithms for Simultaneous Localization And Mapping (SLAM), e. g., [Thr00], and motion planning, e. g., [Phi03], have

been developed often using 2D representations like occupancy grids, metric maps, or topological maps [Moz07]. The vista space becomes relevant for a robot if obstacles must be avoided [Yua09] or an unknown environment needs to be explored. For example, navigation from one room to another room requires not only a path planning on a global map but also an analysis of local scans to react to suddenly appearing objects. An analysis of the vista space is even more important in situations with agent-agent interaction as it is the directly surrounding environment in which the interaction takes place. A collaborative interaction of the partners in and with the environment requires that both partners know where they are, what spatial structures they are talking about, and what scene elements they are going to manipulate. Therefore, a semantic modeling of the vista space, especially in 3D, is essential. My thesis focuses on the computational modeling from a robot's perspective as less work can be found in the area of 3D spatial modeling for human-robot interaction. The goal is to design mechanisms for extracting relevant spatial structures allowing the robot to recognize in which place it is, to detect the scene elements the human partner is talking about, and to segment scene structures the human is changing. For a smooth human-robot interaction, the models computed by the robot should be aligned to the partner's representations [Vas07a], which means that similar scene information needs to be encoded [Pic04]. Towards a complete spatial awareness, the models of a set of vista spaces could be fused with a representation of the covered large-scale space, for example, using the Spatial Semantic Hierarchies (SSH) proposed by Kuipers and colleagues [Bee07, Kui00].

A commonly used scenario where a robot is expected to acquire spatial knowledge in an interactive way is the so-called *"home tour" scenario* [COG04]. A human is instructed to guide a robot around while showing it relevant objects and places in an apartment. This thesis is going to develop methods for analyzing three situations that can arise during a "home tour". The eye-catcher image at the beginning of this section illustrates these situations. First, the robot should be able to recognize the type of a room that has been entered (e. g., "this is a living room"). Second, a description of the room should be analyzed for relevant scene structures and grounded in the visual perception of the room. For example, the relevant scene structure in "there is a bowl on the table" is the "table" because it is the supporting structure for the "bowl". The intention of a room description could be an initial introduction of the room to the robot or an explanation for a subsequent task instruction. In the third situation the robot observes scenes where the human is acting and causing changes to the environment, e. g., opening a cupboard door. The robot should be able to detect the articulated parts of a scene. This is useful in a tutoring situation as it might be easier to show something than to explain it verbally. Furthermore, movable scene parts are relevant in situations where robot and human are going to execute a task together. The following specific research questions arise from these three situations faced:

How can the type of a room be recognized independent from specific furniture arrangements and contained objects?

How can the relevant supporting structures be inferred from a depiction and grounded visually, so that the resulting model is aligned to the describer's representation?

How can articulated scene parts, that are manipulated by a human, be extracted under short observation conditions?

The holistic, aligned, and articulated modeling approaches developed in this thesis suggest solutions to the raised questions. They utilize results from psychological studies on these topics and combine them with basic physical properties in 3D room perception.

The holistic modeling is based on the finding that a brain area being involved in scene recognition is sensitive to the 3D geometry of a scene [Heno8]. As man-made environments mostly consist of planar surfaces, these patches assemble the 3D geometry of a room. Therefore, I am going to introduce a new 3D feature vector which is computed on a set of bottom-up extracted planar patches capturing the **spatial layout** of a room. The challenge is to define the feature in a way that the encoding is independent from the view on the scene and the arrangement of furniture in the scene. The contribution of my approach is a global representation of a room similar to the well-known Gist feature vector [Olio1] but based on the real 3D geometry. The advantage of global representations is that it is independent from the detection of objects and their assignment to a specific room type [Zeno8].

The aligned modeling relies on parallels in the way language and cognition schematize the spatial world [Tve98]. This means that a hearer can build from a description a model of a scene that is similar to the scene model the speaker has built from perception [Wal8o]. Therefore, the goal is to equip the robot with skills for inferring semantically meaningful spatial structures from **spatial descriptions**. The challenge is to meet the given level of detail and to ground the estimated structures in the visual perception. A spatial description mostly consists of relations between an object and its supporting structure or relations between objects located on the same structure. My contribution is a definition of rules handling these two types of relations. The rules transform a depiction into a set of trees encoding its hierarchical character. The grounding of relevant supporting structures into the visual world is realized by utilizing object detection and planar surface extraction. This approach allows a more flexible scene modeling than state-of-the-art approaches that extract semantic labels for spatial structures in a bottom-up way by assigning labels to 3D points using classification [Trio7] or to 3D planar surfaces using an ontology [Nüco8]. Furthermore, ambiguities in the scene descriptions can be resolved by establishing the link between depicted scene knowledge and their visual counterparts.

The articulated modeling proposes to represent a dynamic scene in three layers: one for the moving entities, one for the static scene background, and one for the articulated scene parts. This partition follows the distinction of scene dynamics into *change* and *motion* [Reno2]. Motion is defined as variation of location and change as variation of structure. Therefore, moving persons are determined by a particle filtering of small motion-annotated 3D clusters where velocity vectors are provided by optical flow computation [Luc81]. The challenging problem is to detect articulated scene parts like a cupboard door. Since the articulation is an essential feature of the modeled scene parts, these scene structures can be detected by observing **changes in the spatial environment** caused by their manipulation. As phenomenologically a comparison of currently visible structures with a representation in memory is involved for change detection, this can be realized by comparing a learned static background model with the current perception. For range sensing, this comparison can be nicely implemented as subtraction because the farthest static depth values of one view point belong to the static scene background. Static depth measurements that appear in front of a known static background are assumed to belong to an articulated scene part. Subtracting the estimated background from the current perception gives a methodology for fast segmentation of arbitrary articulated scene parts without knowing them or the associated activities in advance [Peu04].

The thesis is organized as follows: Chapter 2 presents 3D perception suitable for sensing the vista space. Furthermore, some bottom-up processing like planar surface extraction and velocity annotation is introduced. Each of the three modeling approaches is discussed in an own chapter. The chapters describe the relevant related work, the computational models, and their evaluation. Chapter 3 deals with the holistic, Chapter 4 with the aligned, and Chapter 5 with the articulated scene modeling. Chapter 6 summarizes the contributions of the thesis and identifies possible future work.

2

PERCEPTION OF THE VISTA SPACE

As the vista space is in focus of this thesis, first, a definition of the vista space is given in Section 2.1. The vista space models are designed for a mobile robot like the BIelefeld Robot CompaniON (BIRON) which is introduced in Section 2.2. Furthermore, the section describes the "home tour" scenario which is an important application of the robot. Scenarios within the "home tour" which fulfill the definition of the vista space are presented in Section 2.1. Scene models providing semantic information for spatial structures require a sensor system that is able to provide dense 3D data from less-textured surfaces. Section 2.3 presents such a 3D sensor used for sampling depth measurements from the environment. It outlines the working principle of the camera and preprocessing methods for the provided data. For a further spatial analysis it is necessary to extract some basic information. Section 2.4 shows how planar surfaces can be extracted from one frame. Section 2.5 shows for a sequence of frames how 3D data can be enhanced with 3D velocity vectors (in the case of observing a dynamic scene with a static camera) and how camera transformations can be computed (in the case of observing a static scene with a moving camera).

2.1 DEFINITION OF THE VISTA SPACE

The importance of scale to the psychology of space (perception, thinking, memory, behavior) has been discussed by Montello in [Mon93]. He points out that properties of space in human perception are scale-dependent. Based on this assumption, he proposes definitions for the different types of space perceivable with a sensory system. He distinguishes four major classes of psychological space: *figural*, *vista*, *environmental*, and *geographical*. The distinction is based on the *projective* size of the space in relation to the human body neglecting the space's actual or absolute size. The *figural* space is projectively smaller than the body. Its properties can be perceived directly from one place without appreciable locomotion. Figural space is the space of pictures, small objects, and distant landmarks. The *vista* space is projectively as large or larger than the body but can be visually apprehended from a single place without locomotion. Similar to the vista space, Ullman [Ull96] has defined the so-called *visual* space which is the immediately surrounding environment that can be explored quickly by moving the gaze. It is the space of single rooms, town squares, small valleys, and horizons. The *environmental* space is projectively larger than the body and surrounds it. Locomotion is necessary to apprehend this type of space and information has to be integrated over a significant period of time. In literature, this type of space is also often referred to as *large-scale* space, e. g., [Kui00]. It is the space of buildings, neighborhoods, and cities. Last, the *geographical* space is projectively much larger than the body and cannot be apprehended directly through locomotion. It must be learned via symbolic representations such as maps which reduces the geographical space to the figural space.

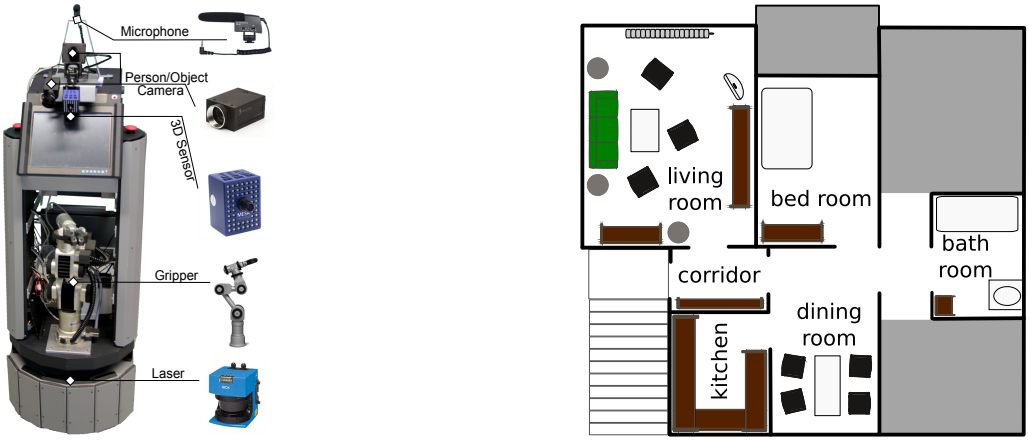
2.2 BIRON – THE BIELEFELD ROBOT COMPANION

The scene models in this thesis are developed with the robot platform BIRON, the BIelefeld Robot companiON, in mind. Section 2.2.1 introduces the platform and Section 2.2.2 describes the "home tour" scenario which is an important application for BIRON. The focus of the thesis is to develop scene models for vista space scenarios within the "home tour ". Section 2.2.3 describes the scenarios in the "home tour" which match the definition given in Section 2.1.

2.2.1 *The Robot Platform*

Figure 2.1(a) shows the current generation of the BIRON platform with the senors attached to it [Wac10]. It is based on the research platform GuiaBot™ [1]. The base is a PatrolBot™ unit which is 59cm in length, 48cm in width, weighs approx. 45kg and is maneuverable with 1.7 meters per second maximum translation and 300+ degrees rotation per second. On top of the platform two MP CCD FireWire

1 http://www.mobilerobots.com



(a) BIRON platform

(b) home-tour apartment

Figure 2.1: (a) shows the platform of the mobile robot BIelefeld Robot CompaniON (BIRON) with close-ups of important hardware components. (b) shows the floor plan of our apartment used for experiments within the "home tour" paradigm. The robot can freely drive around and can enter the living room, the kitchen, the corridor, and the dining room.

cameras (Point Grey Grasshopper) are mounted which are used for person/object detection and recognition. One is facing down for object detection/recognition and one is facing up for face detection/recognition. A SwissRanger™ SR3100 from Mesa Imaging [2] is located on top of the robot providing 3D data used in object grasping and scene analysis. The 3D sensor is either mounted on a pan-tilt unit so that the camera can be moved independently from the robot's body or the camera is fixed facing down by ca. 10°. A chess pattern placed on the floor in front of the robot can be used to compute exactly the transformation of the camera coordinate system to the robot's vertical body and the horizontal ground plane the robot is driving on. Practical details can be found in [Yua09]. For grasping and manipulating objects the robot is equipped with a Katana IPR 5 degrees-of-freedom (DOF) arm, a small and lightweight manipulator driven by 6 DC-Motors. The end-effector is a sensor-gripper with distance and touch sensors being able to lift objects of up to 400g. The upper part of the robot's body comprises a touch screen and a system speaker. The on-board microphone has a hyper-cardioid polar pattern and is mounted on top of the upper part of the robot. The overall height is approximately 140cm.

The software architecture of the BIRON system consists of many different components, each a piece of software providing functionality allowing the robot to interact with humans, to move collision-free around, and to solve tasks given in a "home tour" scenario (→ Section 2.2.2). Example components are speech recognition, person following, map building using SLAM, face recognition, obstacle avoidance, and so on. All components follow the concept of Information-Driven-Integration (IDI) [Wre08] by sharing their data via an Active Memory (AM) [Wre06] on the basis of flexible event notification and XML-based representations with a document-oriented data model. All information generated and revised by components in the system are mediated through this active

2 http://www.mesa-imaging.ch

memory, where it can persistently be stored and retrieved from. The event-driven AM concept is directly supported by the open-source integration framework XCF [3] [Fri07]. On top of this memory architecture, the functional API BirON Sensor Actuator Interface (BonSAI) is defined that abstracts from specific components. This Java API encapsulates hardware sensor information and cross-modal sensors in a sensor class. In the same way actuators are defined that directly control the hardware, e. g., the Pan-/Tilt-/Zoom-Camera or provide cross-modal actuators such as the NavigationActuator that employs different components of the system to get the robot to a certain location. The BonSAI abstraction layer facilitates the implementation of new components and applications making use of all available components of the BIRON platform. So far, less components are realized for processing data of the 3D sensor. Hence, my motivation was also to extend BIRON's abilities for analyzing this type of data.

2.2.2 *The "Home Tour" Scenario*

A main scenario BIRON is designed to deal with is widely known as the "robot home tour" [Han08]. It follows the vision of a (service) robot being delivered to peoples' houses without prior knowledge about the particular environment. It is equipped with capabilities that allow it to learn in an interactive fashion. Hence, the user has to show the robot around the domestic environment and teach it rooms and objects that are relevant. Afterwards, the robot is expected to be able to provide services in a user- and situation-aware manner. The "home tour" is mainly focused on the interactive acquisition of human-adequate representations and on the interaction itself rather than on any specific service. Knowledge of interest thereby comprises topological representations of the living space, models about relevant objects and functional spaces [Zie10], and about different users. The "home tour" scenario recently also gained particular interest by the RoboCup@Home competition [4] where BIRON has participated successfully as part of the Team of BIelefeld (ToBI) [5] [Wac10, Wac09].

Figure 2.1(b) shows the apartment of the Applied Informatics group permanently rented to tackle real-world challenges and to move out of the lab into realistic settings. Non-expert users should interact with BIRON regularly. The goal of the "home tour" is to enable non-expert users to teach a robot about their own living environment in a rather intuitive way, emulating human-human interaction to a certain extent. The user is engaged in interaction with the robot by means of verbal dialog, joint spatial exploration, and gestural reference, to mention only some relevant abilities. Resulting real-world challenges range from small doorways, uncontrolled visual and acoustic conditions to unpredictable human behaviors and reactions.

3 http://xcf.sf.net
4 http://www.robocupathome.org
5 http://www.citec.de/ToBI

2.2.3 *Vista Space Scenes in the "Home Tour"*

A robot acting in a "home tour" scenario (→ Section 2.2.2) mostly encounters *vista* and *large-scale* space situations. A large-scale space in an apartment is the complete apartment as locomotion is required to perceive it in total. Pieces of data are integrate in one global map, for example, using SLAM. Such maps are mostly used for navigation so that details are often skipped for efficient path planning. All observations of indoor rooms taken from a certain view point belong to the vista space. Within an apartment these areas are room parts or single rooms. They are either observed with a static camera or a camera rotated around its axes. I assign methods for analyzing percepts from one view point to the field of vista space analysis. Methods that integrate vista space percepts along paths and examine the integrated data belong to the field of large-scale space analysis. Vista space and large-scale space analysis can differ in the level of details that are modeled.

The goal of this thesis is to provide modeling abilities resulting in a spatial awareness for scenes belonging to the vista space. The robot gains knowledge about meaningful structures enabling later resource conserving strategies for providing services. I focus on three specific situations. First, the robot is instructed to explore the apartment by itself. The goal is to get an overall impression of the apartment. Therefore, the robot needs abilities to determine the room type of visited rooms. Second, a certain room part is described explicitly to the robot in order to enable the robot to solve a specific task for which knowledge about meaningful spatial structures is essential. Here, methods are needed which allow the robot to process verbal descriptions given during an interaction between human and robot. These descriptions reveal important structures like a table where objects can be placed. Third, scene changes need to be handled which provide a further input to a spatial analysis system. For example, if the robot has "spare time", it can observe the human interacting with the environment when he/she cleans up the room. The robot learns from observing scene changes which room parts are static and which have been relocated. This general scene encoding provides additional information about functional structures like, e. g., a door, just by opening and closing it.

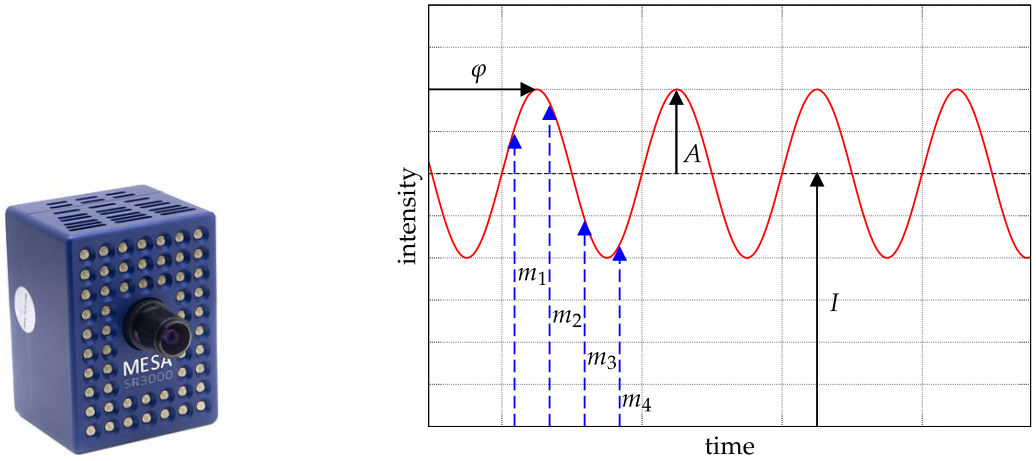
Definition. ***Scene.***

Throughout this thesis, a scene is a specific observation during one of the described situations, namely, exploring autonomously an apartment, listening to spatial descriptions about spatial arrangements, and monitoring spatial changes.

2.3 A SENSOR FOR PERCEIVING SPATIAL STRUCTURES IN 3D

My goal is to develop modeling approaches for 3D percepts acquired from vista space situations as described in Section 2.2.2. The models should equip the robot with a spatial awareness, which means that 3D perception should be able to perceive spatial structures that are present in every-day indoor environments. Together with the specification that the vista space can be perceived quickly at a glance this leads to some requirements for a suitable acquisition of 3D data. First, it must be able to provide reliable 3D data from less-textured surfaces as indoor environments are oft assembled by homogeneous surfaces like tables, walls, modern sofas, etc. Second, the 3D sensing has to happen at a proper frame-rate allowing a quick gaze on scenes and observations of dynamics scenes like moving persons. Third, a dense 3D point cloud should to be delivered ideally. Basically, three different sensor types for 3D perception are in use on robot platforms: Time-of-Flight cameras, 2D camera based vision like stereo cameras or Structured-Light cameras, and Laser range finders. Each of the three sensor types operates best under specific conditions. The Time-of-Flight sensors fulfill the three requirements listed above. They sample from scene surfaces a dense 3D point cloud using infrared light at a frame rate of up to 30fps. In contrast to ToF cameras, stereo cameras estimate depth by computing the disparity of the two cameras from detected point correspondences. The stereo cameras deliver precise 3D data for textured objects but often fail to provide dense and reliable 3D data for homogeneous areas. This could be handled by enhancing the sensor system with Structured-Light projectors. For example, WillowGarage has mounted on their PR2 robot [6] a LED Texture Projector that is triggered with a stereo camera. However, such projections might disturb the interaction between human and robot. Recently, Microsoft has launched the Kinect camera [7] [8] [9] which uses an infrared emitter to project a point pattern being invisible for the human eye. The depth information is calculated from the distortion of the pattern. In principle, it is also possible to get dense 3D from a Laser range finder, but its operation mode does not allow to get it in real-time. As the Laser beam has to be redirected for every measurement only one distance can be sampled in parallel. The algorithms presented in Chapter 3 to Chapter 5 are developed without a special device in mind. They only require a dense 3D point cloud which is delivered with at least 10fps. Currently, Time-of-Flight cameras like the PMD[vision]® camera [10] or the SwissRanger™ camera [11] and Structured-Light cameras like the Kinect camera are best suited to deliver in real-time reliable and dense 3D point clouds. Due to its small and lightweight body, the SwissRanger™ SR3100 (→ Figure 2.2(a)) has been mounted on the BIRON platform (→ Section 2.2.1). The following sections shortly introduce the working principle of the SwissRanger camera and some preprocessing of the output data.

6 http://www.willowgarage.com/pages/pr2/overview
7 http://www.xbox.com/en-US/kinect
8 http://openkinect.org/wiki/Main_Page
9 http://openni.org
10 http://www.pmdtec.com
11 http://www.mesa-imaging.ch



(a) SR3100 (b) measurement principle

Figure 2.2: The measurement principle of a SwissRanger camera comprises a measurement of four values $\{m_i\}_{(i=1,\ldots,4)}$ per period done at equal intervals. They allow to recover the measured modulated sinusoidal that is entirely determined by its phase shift φ, its average intensity I, and its amplitude A [Weio4].

2.3.1 *Working Principle of the SwissRanger Camera*

The use of SwissRanger cameras in the field of robotics has been initially presented by Weingarten [Weio4]. The camera is a solid-state imaging device that delivers distances and intensity images [Oggo4, Lano1]. The camera is based on CMOS pixel sensors arranged into one image plane and a modulated light source. The SR3100 assembles 176×144 pixel sensors allowing the camera to deliver simultaneously 25344 distance measurements. The SwissRanger camera relies on the time-of-flight principle. The distance d between the camera and an object is determined by measuring the time Δt an emitted light signal needs from the camera to the object and back:

$$\Delta t = \frac{2d}{c} \quad \text{with } c \text{ the speed of light.} \tag{2.1}$$

As the infrared light emitted by the camera is modulated by a single frequency f_{m} the time-of-flight Δt can be directly computed from the phase shift φ between the signal sent and received:

$$\varphi = 2\pi \cdot f_m \cdot \Delta t \; . \tag{2.2}$$

This phase shift is determined by sampling per pixel sensor the amount of modulated light reflected by objects in the scene. This is done four times every period of the modulation signal at equal intervals.

The four measurements – m_1, m_2, m_3, and m_4 – allow a recovering of the incoming sinusoidal signal. The phase shift φ and the corresponding distance d are computed by:

$$\varphi = \arctan\left(\frac{m_4 - m_2}{m_1 - m_3}\right), \tag{2.3}$$

$$d = d_{\max} \cdot \frac{\varphi}{2\pi}, \tag{2.4}$$

$$d_{\max} = \frac{c}{2f_{\mathrm{m}}}. \tag{2.5}$$

where $d_{\max}$ is the non-ambiguity range of the sensor determined by the modulation frequency of the emitted light. The amount of reflected light can be used to recover the intensity I and amplitude A of the measured sinusoidal:

$$I = \frac{m_1 + m_2 + m_3 + m_4}{4}, \tag{2.6}$$

$$A = \frac{\sqrt{(m_3 - m_1)^2 + (m_4 - m_2)^2}}{2}. \tag{2.7}$$

The amount of reflected light is later used to estimate the reliability of a distance measurement as different materials have different reflection properties disturbing the measurement by more or less noise.

The pixel sensors of the SR3100 have a height of $h = 40\mu$m and a width of $w = 40\mu$m. The default modulation frequency of 20MHz results in a non-ambiguity range of $d_{\max} = 7.5$m. For distances between 0.3 and 3m the frame rate deviates between 12 and 29 Hz with a depth resolution of 2.5 to 22mm at the central pixel. The central pixel is located at position $(\,92,\ 60\,)^{\mathrm{T}}$. The modulated illumination is generated by a set of 48 near-infrared LEDs. The lens of the camera has a focal length of 8mm and a field of view of about 43° horizontally and 46° vertically. Figure 2.4 shows an example output of the SwissRanger camera. An amplitude and a distance image can be seen.

2.3.2 *Preprocessing of SwissRanger Data*

The raw output of the SwissRanger is quite noisy. The following paragraphs introduce some basic preprocessing of the output data. Distance values are smoothed using a *distance-adaptive median* filter, unreliable measurements are determined with respect to their amplitude values and their membership to depth edges. Further, back-projection is utilized to convert the distance values to real 3D world coordinates.

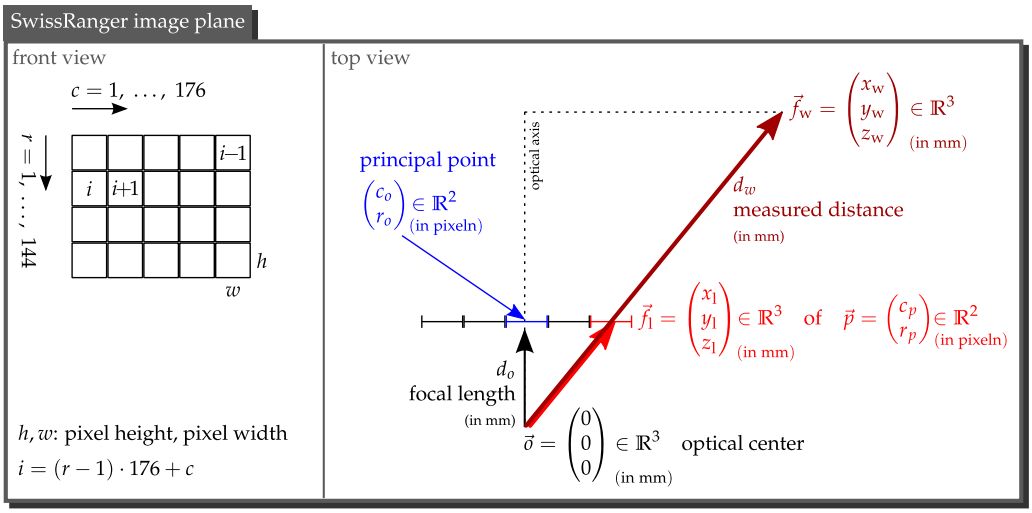


Figure 2.3: The figure illustrates schematically the SwissRanger image plane and the ray proportions necessary to transform the measured distance d_w to a world coordinate $\vec{f}_w$. Front view: each pixel in the image plane is uniquely tagged with $[\,c\;r\,]$ or a index $i = (r-1)\cdot 176 + c$ iterating the image plane row-wise. Top view: the origin of the world coordinate system is located in the optical center $\vec{o}$ of the camera. The ray proportions in the spanned triangle are used to scale the local pixel coordinates to global world coordinates.

BACK-PROJECTION. For later use in this thesis the measured distances of one frame $\mathcal{F}$ have to be transformed into 3D world coordinates $\{\,\vec{f}_i\,\}$. The origin of the coordinate system is aligned to the optical center $\vec{o}$ of the camera. Given some parameters of the camera – like the focal length d_o (here, in mm), the principal point $(\,c_o,\; r_o\,)^{\mathrm{T}}$ (provided by perpendicular projection of the optical center $\vec{o}$), the pixels width w and pixels height h (also in mm) – the local 3D coordinates $\vec{f}_l$ of a pixel $\vec{p} = (\,c_p,\; r_p\,)^{\mathrm{T}} \in \mathbb{R}^2$ can be computed by:

$$\vec{p} = \begin{pmatrix} c_p \\ r_p \end{pmatrix} \in \mathbb{R}^2 \;\rightarrow\; \vec{f}_l = \begin{pmatrix} x_l \\ y_l \\ z_l \end{pmatrix} \in \mathbb{R}^3 \quad \text{with} \quad \begin{aligned} x_l &= (c_p - c_o)\cdot w \\ y_l &= (r_o - r_p)\cdot h \\ z_l &= d_o. \end{aligned} \tag{2.8}$$

The corresponding world point $\vec{f}_w$ is computed by scaling the local coordinates x_l, y_l, and z_l by the factor θ_p that is determined using the distance value d_w measured in the pixel $\vec{p} = (\,c_p,\; r_p\,)^{\mathrm{T}}$ and the ray relations in the corresponding triangle ($\rightarrow$ Equation 2.9). As shown in Figure 2.3 the triangle is spanned by the optical axis and the ray with length d_w through the pixel $(\,c_p,\; r_p\,)^{\mathrm{T}}$.

$$\vec{f}_w = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \in \mathbb{R}^3 \quad \text{with} \quad \begin{aligned} x_w &= \theta_p \cdot x_l \\ y_w &= \theta_p \cdot y_l \\ z_w &= \theta_p \cdot z_l \end{aligned} \tag{2.9}$$

$$\text{and} \quad \theta_p = \frac{d_w}{d_l} \quad \text{where} \quad d_l = \sqrt{x_l^2 + y_l^2 + z_l^2}.$$

Sensors like the SwissRanger camera give in a nice way a 2D arrangement for a 3D point cloud. Throughout this thesis it can be seen that this combination of 3D and 2D information reduces computational costs as standard 2D techniques can be utilized. For example, adjacent points can be determined much easier on the 2D image plane than in the complete 3D space. In Figure 2.6(a) the resulting 3D point cloud can be seen when applying back-projection to the original distance image given in Figure 2.4(b).

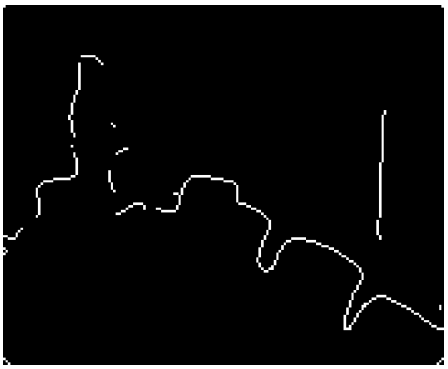
DISTANCE-ADAPTIVE MEDIAN FILTERING. Figure 2.4(b) shows a raw distance image with some noise from bad reflecting areas like the floor. As argued above the 3D point could can be smoothed efficiently by applying 2D filter techniques to the distance image. A standard approach is to use a median filter which smooths homogeneous regions and preserves edges. Choosing the adequate filter size is the crucial point as on the one hand noise should be removed and on the other hand details should be preserved. Taking into account the different distance measurements a proper size can be identified in a nice way. Due to projection properties an object is going to cover an increasingly smaller area on the image plane if it is moving away from the camera. Therefore, large distance measurements have to be smoothed with a small filter in order to keep details while small depth values can be convolved with a bigger filter. In [Swa07], I have introduced a *distance-adaptive* median filter which takes into account this projection property. The size of the filter applied to a distance value depends on the assignment to one of the three intervals: $[\,0,\ \frac{1}{3}d_{\max}\,]$, $]\,\frac{1}{3}d_{\max},\ \frac{2}{3}d_{\max}\,]$, and $]\,\frac{2}{3}d_{\max},\ d_{\max}\,]$ with $d_{\max} = 7.5$m. Possible filter sizes are 7×7, 5×5, and 3×3 (the order here determines the association of a filter size with an interval mentioned before). Applying back-projection to the smoothed depth map leads to a smooth 3D point cloud. Figure 2.5(a) shows the smoothed distance image. Hedge and Ye [Hed08] have proposed a Singular Value Decomposition (SVD) based filtering method of SwissRanger data. They convert a conventional range image into an enhanced range image where each pixel's intensity embodies the surface normal and the depth information of the corresponding pixel in the original image. This enhanced image is decomposed via SVD into the matrices $\mathbf{U}$, $\mathbf{V}$, and the diagonal matrix $\mathbf{D}$. The smoothed depth image is reconstructed from multiplying the three matrices after setting small diagonal values in $\mathbf{D}$ to 0. Even though the smoothed results are quite convincing this approach has the drawback of high computational costs as for each point a surface normal has to be estimated, for example, using the method presented in Section 2.4.1. This results in a reduction of the output frame rate thus the SVD based filtering might not be applicable in dynamic scenes like encountered in Chapter 5 where a reasonable frame rate is required for robust entity tracking. As a consequence, I continued to use my distance-adaptive median filtering throughout this thesis.

(a) amplitude image

(b) distance image

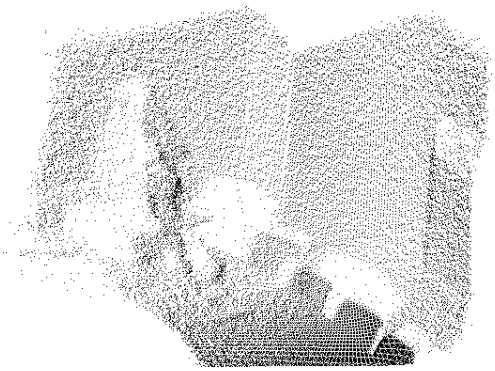
Figure 2.4: This figure shows an example amplitude and distance image delivered by a SwissRanger SR3100. In the amplitude image white pixels denote a huge amount of reflected infrared light while black pixels denote bad reflection properties. Black pixels in the distance image refer to small distance values and white pixels to large distance values.

(a) smoothed distance image

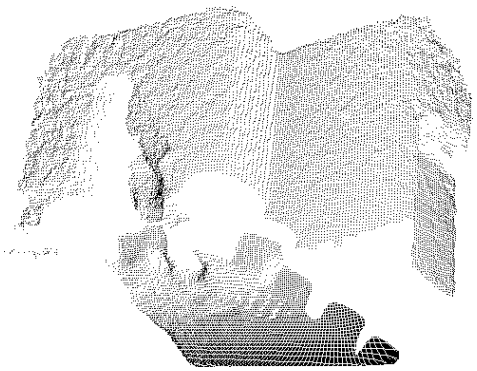


(b) depth edges

Figure 2.5: (a) This distance image has been smoothed using the distance-adaptive median filter. (b) Depth edges can be computed on the smoothed distance image. Here, the Sobel filter has been applied.



(a) raw 3D point cloud



(b) smoothed and valid 3D point cloud

Figure 2.6: This figure shows in (a) the raw 3D point cloud computed via back-projection from the original distance image. (b) displays the 3D point cloud after smoothing the distance image with the distance-adaptive median filter and removing invalid points via amplitude thresholding and depth edge detection.

AMPLITUDE FILTERING. As outlined in Section 2.3.1 the amplitude image (→ Figure 2.4(a)) provided by the SwissRanger camera holds for each pixel the amount of reflected infrared light. Surfaces which reflect infrared light well will have a large amplitude value and will appear in light colors in the amplitude image. The more light is reflected the more information is accumulate by the pixel sensor for measuring the phase shift in the light signal. Hence, amplitude values indicate indirectly the reliability of the distance measurements. A depth value with an amplitude value below a certain threshold θ_a is declared to be an invalid or noisy measurement. For an arbitrary amplitude image $\mathcal{A}$ the corresponding threshold θ_a is a fraction of the mean amplitude value:

$$\begin{aligned} \theta_a &= \frac{1}{3}\bar{A} \quad \text{with } \bar{A} = \frac{1}{n}\sum_{i=1}^{n} A_i \quad \text{and} \\ \mathcal{A} &= \left\{ A_i \right\}_{i=1\,\ldots,\,n} \quad (\text{here, } n = 176 \cdot 144). \end{aligned} \tag{2.10}$$

EDGE FILTERING. A common problem in actively sensing systems are the so-called "flying pixels". Especially at edges, a pixel sensor may collect simultaneously light reflected from a foreground object and a background surface. Both signals cannot be distinguished which results in a hallucination of a 3D point somewhere "flying" between the foreground and the background. In [Swa07], I have proposed to determine for each 3D point the amount of near neighboring points. Points are declared as valid if there are enough near points in their 3×3 neighborhood. Even though, this method removes some flying pixels the definitions for "near point" and "enough points" have to be estimated empirically. This parameter tuning can be skipped by computing edges in the distance image using, e. g., a Sobel filter (→ Figure 2.5(b)). Points located on these *depth edges* are removed from the valid point cloud. This approach removes successfully noisy points where the previous method has failed since a set of scattered points appearing along an edge leads to the effect that flying pixels have enough close points supporting each other.

Figure 2.6(b) shows the final 3D point cloud after all preprocessing steps have been applied. Compared to the raw point cloud in Figure 2.6(a) the 3D points are smoothed convincingly and scattered points along depth edges and from badly reflecting surfaces like the floor are removed reliably.

2.4 BASIC PROCESSING OF A SINGLE PERCEPT

This section presents some extraction of basic information from one scene snapshot. Section 2.4.1 describes how 3D points can be enhanced with local surface information. Section 2.4.2 discusses the extraction of geometric primitives suitable as first abstraction from raw data.

2.4.1 *Computing Oriented Particles*

As the SwissRanger camera samples 3D points from surfaces in the scene, estimating the orientation of the 3D points will allow to infer characteristics of the scanned surfaces. Fua has proposed to define a point $\vec{f}_i$ as an *oriented particle* $\mathcal{P}_i$ which means that a 3D point is enhanced with a normal vector $\vec{n}_i$ encoding the point's orientation [Fua97]:

$$\mathcal{P}_i : \ \vec{n}_i \cdot \vec{x} - d_i = 0. \tag{2.11}$$

Equation 2.11 describes the oriented particle in Hessian Normal form. d_i is the Euclidean distance of the particle's centroid to the origin of the world coordinate system. Assuming piecewise planarity of surfaces, points in the neighborhood of a point can be used to estimate the normal of this point. In principle, one can think of using k-nearest neighbors or points in a fixed distance [Rab06]. Independent from the chosen approach, searching for neighboring points directly in the 3D space has an increased complexity and requires some effort for efficient space representation. Representations can range from a decomposition of the 3D space into regular cubes [Wei03] to a data-driven one into octrees [Sam02] or kD-trees [Lee77]. Alternatively, approximate nearest neighbor methods like "best bin first" [Bei97] can increase the search efficiency. In my case, the SwissRanger

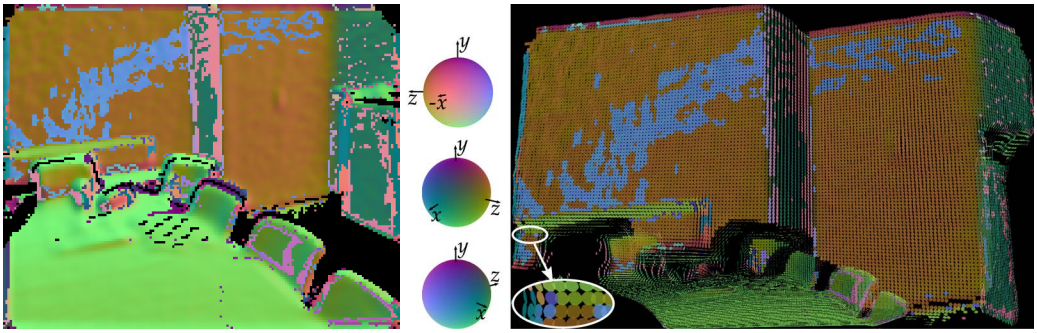


Figure 2.7: The right image shows an example frame of the scene shown in Figure 2.4(a) where the 3D points $\{ \vec{f}_i \}$ are replaced by small planar patches representing the computed oriented particles $\{ \mathcal{P}_i \mid \vec{n}_i, \ d_i \}$ (see zoom in the left bottom edge of the right image). The patches are colored according to their orientations to the axes of the coordinate system. Parallel planes like the wall in the back and the cupboard doors are colored equally (e. g., orange) or with a complementary color (e. g., blue) depending on the sign of the normal. The left image shows the 2D image where each pixel is colored according to the orientation of the corresponding 3D point. (best viewed in color)

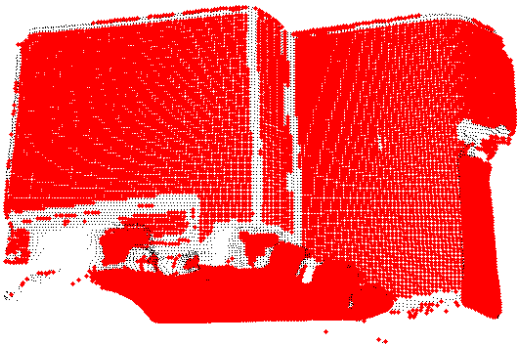


Figure 2.8: This figure shows the points of an example frame which have been declared as planar when defining oriented particles. Points at sharp edges are not planar. This is an advantage for the planar surface extraction because these points stop growing of regions at edges.

characteristics provide a by-pass to the nearest neighbor computation in 3D space. As illustrated in Figure 2.3, each 3D point of a SwissRanger frame has a corresponding 2D pixel in the camera's image plane. Points are neighbors in 3D if their corresponding 2D pixels are neighbors on the image plane. Therefore, the normal vector $\vec{n}_i$ of a point $\vec{f}_i$ is computed from a set of neighboring points $\{ \vec{f}_j \}$ selected through determining the 8-neighborhood $\mathcal{N}_{3\times3}$ of point $\vec{f}_i$ on the image plane:

$$\left\{ \vec{f}_j \mid \vec{f}_j \in \mathcal{N}_{3\times3} \text{ of } \vec{f}_i \right\}. \tag{2.12}$$

Applying Principal Component Analysis (PCA) to this set of points estimates the normal vector $\vec{n}_i$ by choosing the eigenvector with the smallest eigenvalue. The centroid point computed from the point set $\{ \vec{f}_j \} \cup \vec{f}_i$ is used to compute d_i in Equation 2.11. Figure 2.7 shows for an example point cloud the computed oriented particles. A particle $\mathcal{P}_i$ is colored according to its orientation to the axis $\vec{x}$, $\vec{y}$, and $\vec{z}$ of the world coordinate system. The color $[\, R_i,\ G_i,\ B_i \,]$ is determined by:

$$\begin{aligned} &[\, R_i,\ G_i,\ B_i \,] \ = \ \left[\, \frac{\alpha_x}{\pi},\ \frac{\alpha_y}{\pi},\ \frac{\alpha_z}{\pi} \,\right] \quad \text{with} \\ &\alpha_x \ = \ \arccos(\vec{x} \cdot \vec{n}_i), \quad \alpha_y \ = \ \arccos(\vec{y} \cdot \vec{n}_i), \quad \alpha_z \ = \ \arccos(\vec{z} \cdot \vec{n}_i) \\ &\text{“}\cdot\text{”}: \text{ scalar product} \end{aligned} \tag{2.13}$$

It can be seen that parallel planes like the wall and the cupboard doors contain points with normals of similar orientation (here, orange). Particles colored with the complementary color (here, blue) have the same orientation but the normal is reflected along the plane resulting flipped signs. I am going to tackle the problem of different normal signs by just considering the acute angles between normals when extracting planar patches.

According to [Sta02], the deviation σ_i of the point $\vec{f}_i$ to the fitted plane $\mathcal{P}_i$ can be used as a measurement to judge the quality of the plane fitting. Points with a deviation below a threshold θ_σ are classified as *locally planar* otherwise as *non-planar*. Points which are not part of locally planar surfaces are detected in this way. Figure 2.8 highlights in the example point cloud the planar points in red. As expected, points at sharp edges are labeled as non-planar.

2.4.2 *Extracting Planar Surfaces*

While the normal vectors just hold local characteristics of the 3D points, extraction of large continuous surfaces can be used to represent the scene on a more abstract level. If you look around in a standard indoor room most surfaces are planar. Hence, focusing on the extraction of planar surfaces as geometric primitives is an acceptable restriction. First, some related work is presented and their advantages and disadvantages are discussed. Second, the implemented algorithm is outlined.

RELATED WORK. In principle there are three main algorithms for extracting planar surfaces in 3D data. Expectation Maximization (EM) and Region Growing (RG) are the most used techniques while pure RANdom SAmple Consensus (RANSAC) [Fis81] is less frequently used.

The advantage of the RANSAC algorithm is that planes can be fitted robustly into data while omitting outliers. In general, three points are chosen randomly determining a plane and the remaining points are added if they fulfill the plane equation. This scheme is rerun several times and the most supported plane is selected. Nüchter [Nüc08] runs RANSAC several times to find iteratively the main planes by choosing randomly a point, fitting a plane to this point by using its neighboring points and adding points to the current set that fulfill the plane equation. The resulting set of points is refined by the Iterative Closest Points (ICP) algorithm. Lee [Lee05] enhances Scale-Invariant Feature Transform (SIFT) features with 3D positions and extracts planes via RANSAC from these stereo-sis SIFT features. The extracted planes are priors for the subsequent object detection. The main drawback of pure RANSAC-based methods is the fact that a boundary constraint cannot be simply integrated. In cluttered scenes, planes can be extracted which consist of two or more unconnected patches or contain points from the intersection of planes with other planes. As shown by Nüchter, RANSAC performs well in *convex* scenes like a floor but cannot be applied directly to crowded rooms of the living space.

The second method, Expectation Maximization (EM), estimates plane models and main directions for a given number of planes such that the likelihood of the data is maximized. During expectation (E-step), for each point probabilities are estimated that encode the belonging of the point to planes estimated during the previous run using its distances to the planes. During maximization (M-step), the new positions of the planes are computed using a regression weighted with the probabilities of the E-step. Afterwards, the number of planes is optimized. The EM optimization is normally point-based however Andreasson et al. [And05] propose also to incorporate color information. The purpose is to support plane estimation in noisy data but has the drawback to produce a huge amount of patches in textured scenes. As EM needs to know the number of planes beforehand this number has to be estimated outside the EM computation. In literature, different methods are reported for estimating the correct number of planes. For example, the Bayesian Information Criterion (BIC) [Sch78] can be minimized by dropping planes as a high BIC value denotes redundancy in the

model [Tri05, And05]. Lakaemper [Lak06] divides patches into tiles which drive splitting and merging of patches if they are unsupported. Martin [Mar02] uses a straightforward Bayesian prior to penalize complex maps. The complexity penalization is combined with the data likelihood through a maximum posterior probability estimator.

The last set of methods are based on the Region Growing (RG) technique. If using RG, the number of patches needs not to be adjusted beforehand. The algorithm starts with a seed point and extends the current set with points from the neighborhood if a homogeneity criterion (mostly, planarity criterion) is fulfilled. After the growing has stopped, a new plane is initialized by selecting randomly a point from the set of remaining points [Dor07]. Resulting patches are often noisier compared to those produced by EM and requires therefore subsequent smoothing such as restarting region growing several times in a RANSAC-like manner [Häh03], refining seed regions using graph-cuts where additional edge information is incorporated [Käh08], or introducing intensity similarity as an additional homogeneity criterion [Cob01]. Alternatively, enhancing 3D points with information from its local neighborhood, in particular surface orientation through normal vectors, has shown a positive effect on planar surface extraction using RG [Hoi08, Rab06, Wei03, Sta02].

An interesting combination of Region Growing (RG), RANSAC, EM, and 3D point enhancement with orientations is proposed by Murray and Little [Mur04]. They assume the number of planar regions via RG over patch-lets (oriented 3D points) and rerun it several times to avoid the problem of explicitly estimating the planar boundaries. Subsequently, the found planar surfaces are refined using the EM paradigm.

Recent approaches also have started to recover the spatial scene layout from 2D images. The spatial layout of the scene is mainly inferred from extracted line segments. For example, Yu et al. [Yu08] cluster lines to obtain depth-ordered planes, while Lee et al. [Lee09] analyze sets of lines with rules describing geometric constraints. Using the *Manhattan World* assumption saying that indoor rooms mainly consist of orthogonal planes, a geometric reasoning delivers the most plausible physical interpretation for a set of lines. Alternatively, a Markov Random Field (MRF) model can be used to identify the different planes and edges in the scene, as well as their orientations [Del05]. Hedau et al. [Hed09] and Wang et al. [Wan10] jointly estimate a coarse space model for an indoor room by fitting a parametric 3D box to the extracted lines and locate walls, the floor, the ceiling, and objects via surface labels of pixels [Sax08, Hoi07]. The strength of their approach is the ability to deal with clutter in the scene disturbing the visibility of the room frame.

IMPLEMENTATION. For extracting planar patches in SwissRanger data with the aim to use them for 3D scene analysis the algorithm should be able to extract an unknown number of bounded connected patches in data from a cluttered scene. My algorithm for decomposing a point cloud into connected planar regions is based on Region Growing (RG) [Swa08c]. Further, surface properties like a normal orientation are incorporated which decreases, in contrast to edge-based methods, the sensitive of the segmentation to noise [Rab06]. An additional dimensionality is provided that reduces ambiguity of segmentation-by-clustering [Mur04]. A final refinement of the extracted regions is done via RANSAC. Iteratively, a point is selected as seed of a new region and extended with points of the 8-neighborhood $\mathcal{N}_{3\times3}$ if four criteria are fulfilled. A neighboring point can be added to the region if it is *valid* (→ Section 2.3.2) and *planar* (→ Section 2.4.1). Further, the *conormality* and *coplanarity* criterion has to apply between the seed point and the neighboring point [Sta02]. Two points $\vec{f}_1$ and $\vec{f}_2$ are conormal if for the acute angle α between their normals $\vec{n}_1$ and $\vec{n}_2$ the following statement holds:

$$\alpha = \begin{cases} \arccos(\vec{n}_1 \cdot \vec{n}_2) & :<= \frac{\pi}{2} \\ \pi - \arccos(\vec{n}_1 \cdot \vec{n}_2) & : \text{else} \end{cases} \tag{2.14}$$

$$\alpha < \theta_\alpha \quad \Leftrightarrow \quad \text{conormal}\Big(\vec{f}_1,\ \vec{f}_2 \Big). \tag{2.15}$$

Taking into account the noise level of the camera the threshold θ_α is set to $\theta_\alpha = 10°$. Two points $\vec{f}_1$ and $\vec{f}_2$ are coplanar if the distance d is smaller than a threshold θ_d:

$$d = \max\Big(|\vec{r}_{12} \cdot \vec{n}_1|,\ |\vec{r}_{12} \cdot \vec{n}_2| \Big), \tag{2.16}$$

$$\vec{r}_{12} = \vec{f}_1 - \vec{f}_2$$

$$d < \theta_d \quad \Leftrightarrow \quad \text{coplanar}\Big(\vec{f}_1,\ \vec{f}_2 \Big). \tag{2.17}$$

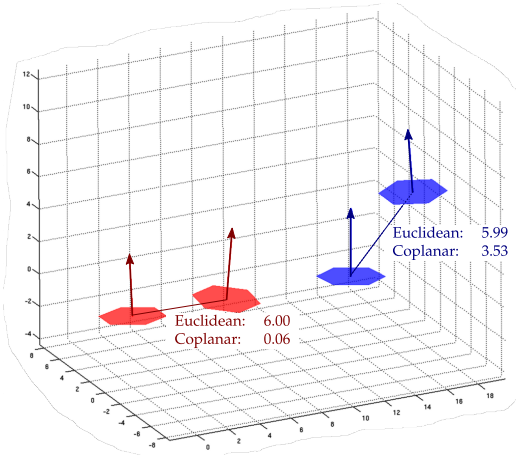
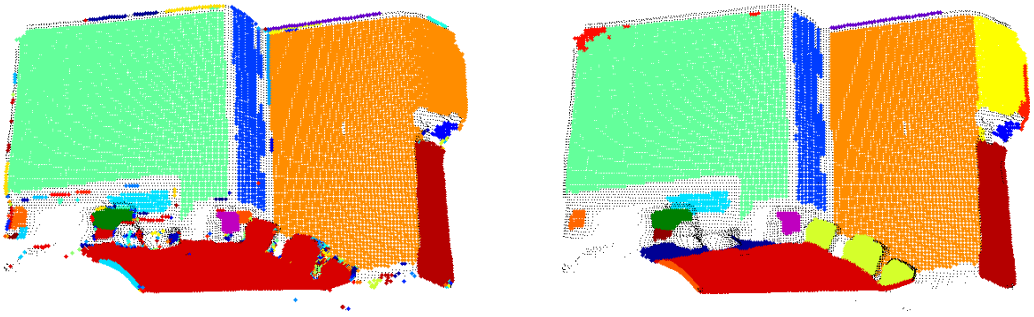


Figure 2.9: The red colored patch pair is conormal and coplanar, while the blue colored one is conormal but not coplanar.



(a) Patches provides by Region Growing.

(b) Planar patches $\{\mathcal{P}_k\}_{k=1,\,...,\,m}$ after RANSAC post-processing.

Figure 2.10: (a) shows the initially extracted bounded patches applying Region Growing (RG) on normal vectors. Points are added to a region if they fulfill the validation, planarity, coplanarity, and conormality measurement. Due infrared light reflections in corners normals change at the transition between different oriented walls less strong (e. g., cupboard doors to wall, table to chairs). As a consequence, they are merged during RG. (b) Resulting planar surfaces after RANSAC post-processing. Problems with smooth transitions between neighboring surfaces are solved.

The distance d is computed with respect to the orientation and distance of the oriented particles. For every newly added point the described selection of neighboring points is repeated. The growing stops if no point can be added to the region. From the remaining points a new point is selected to initialize a new region. The result is a set of nearly planar patches where Region Growing over the 8-neighborhood and the coplanarity measurement ensures their connectivity and compactness because it deals especially with the situation at depth jumps. Figure 2.9 shows in blue the case where the conormality measurement would fail to assign the two particles to different planar regions. While both pairs have nearly the same Euclidean distance the coplanarity measurement differs significantly. Figure 2.10(a) shows the initial Region Growing result. Due to infrared light reflections in inward-looking corners the normals at the transition between two neighboring scene surfaces differ less strong. As a consequence, such surfaces fall into one smooth patch like the table and the chairs or the cupboard doors and the right wall. Nevertheless, Region Growing separates a cluttered scene into a set of convex subparts. These subparts can be further decomposed using some RANSAC iterations extracting the largest planes while omitting outliers [Nüco8]. Per region, a point is chosen randomly which determines through its normal vector a candidate plane. For all remaining points of the region their distance to this plane is used to decide whether the point is an inlier or an outlier. An inlier is found if the distance is smaller than 100mm. A minimum patch size ensures noise reduction by dropping patch candidates with less than 30 points. This procedure is rerun several times with different points determining the reference plane. The largest set of inlier points determines the largest planar patch $\mathcal{P}_k$. The plane parameters of $(\,\mathcal{P}_k \mid \vec{n}_k,\ d_k\,)$ are computed using PCA. RANSAC is applied on the remaining outlier points to find the second, third, ... largest planar patch. Figure 2.10(b) shows the final patches after RANSAC post-processing: cupboard doors and wall as well as table and chairs are separated into own planar patches. Further, patches that are too small have been removed.

2.5 BASIC PROCESSING OF CONSECUTIVE PERCEPTS

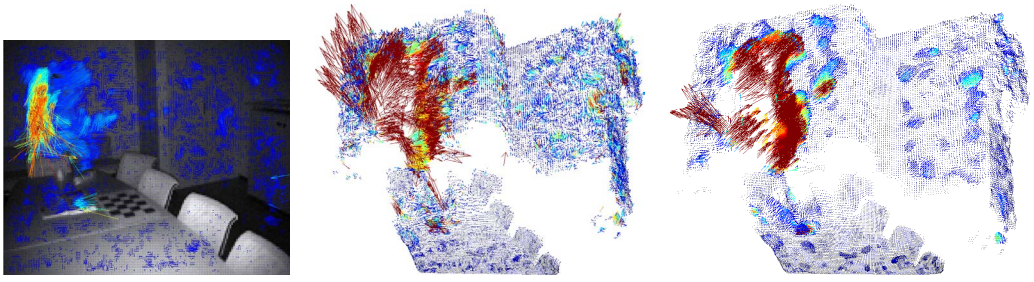
A sequence of SwissRanger frames is required for analyzing dynamic scenes (→ Chapter 5) or for perceiving larger regions than provided by one SwissRanger frame. Considering two consecutive frames, Section 2.5.1 shows how 3D data can be enhanced with 3D velocity vectors. Section 2.5.2 sketches how information from several frames can be fused to one consistent 3D point cloud.

2.5.1 *Extending 3D Data with Velocities*

The analysis of dynamic scenes often requires knowledge about motion present in the scene. If the SwissRanger camera observes the scene from a static view point the high frame rate of the camera results in small changes between two consecutive frames. This allows to use local techniques like dense optical flow computation for estimating motion in the scene. The *optical flow* of a 2D image pixel is the distribution of apparent velocity of moving brightness pattern in an image which can arise both from the relative objects' and the viewer's motion [Gib50]. The flow of a constant brightness profile $I(c, r)$ is described by a constant velocity vector $\vec{v}^{2D} = (\, v_c,\, v_r \,)^{\mathrm{T}}$:

$$\begin{aligned} I(\, c,\, r,\, t\,) &= I(\, c + \mathrm{d}c,\, r + \mathrm{d}r,\, t + \mathrm{d}t\,) && (2.18)\\ &= I(\, c + v_c \cdot \mathrm{d}t,\, r + v_r \cdot \mathrm{d}t,\, t + \mathrm{d}t\,) \\ \Rightarrow \quad -\frac{\partial I}{\partial t} &= \frac{\partial I}{\partial c} \cdot v_c + \frac{\partial I}{\partial r} \cdot v_r && (2.19) \end{aligned}$$

where Equation 2.19 is the *optical flow constraint* which has to be solved. Usually, differential methods estimate the optical flow. They optimize either a global energy functional like proposed by Horn and Schunck [Hor81] or a local energy like expression like proposed by Lucas and Kanade [Luc81]. Lucas and Kanade find for each pixel $\vec{p}_i$ in the image $\mathcal{I}_1$ a good match in the image $\mathcal{I}_2$ using a



(a) (b) (c)

Figure 2.11: (a) shows the 2D optical flow field computed using two consecutive amplitude images. (b) shows the 3D velocity vectors resulting from point correspondences in 3D established by the underlying 2D pixel correspondences given through the 2D optical flow field. (c) shows the final 3D velocity vectors smoothed with median filters. For clarity each 3^{th} velocity vector is displayed and colored with respect to its length (red = large and blue = small velocity vectors).

type of Newton-Raphson iteration. They assume the optical flow to be constant within a certain neighborhood $\mathcal{N}$ which allows to solve Equation 2.19 via least square minimization.

For SwissRanger data the optical flow computation can be performed on the amplitude images of two consecutive frames $(\mathcal{F}_t, \mathcal{F}_{t-1})$. I use the hierarchical implementation of Sohaib Khan [12] [13]. As can be seen in Figure 2.11(a) each pixel of frame $\mathcal{F}_t$ is annotated with a 2D velocity vector providing a corresponding pixel in frame $\mathcal{F}_{t-1}$. As each 2D pixel $\vec{p}_i$ is associated with a 3D point $\vec{f}_i$ a pixel correspondence $(\vec{p}_k^t, \vec{p}_l^{t-1})$ can be transformed directly to a correspondence of 3D points $(\vec{f}_k^t, \vec{f}_l^{t-1})$. The 3D velocity vector $\vec{v}_k^{3D}$ which enhances the 3D point $\vec{f}_k$ in frame $\mathcal{F}_t$ is computed through:

$$\vec{v}_k^{3D} = \begin{pmatrix} v_x^k \\ v_y^k \\ v_z^k \end{pmatrix} = \vec{f}_k^t - \vec{f}_l^{t-1}. \tag{2.20}$$

Figure 2.11(b) shows for an example frame $\mathcal{F}$ the estimated 3D velocity field $\mathcal{V} = \{ \vec{v}_i^{3D} \}$. Each 3D point $\vec{f}_i$ has a velocity vector $\vec{v}_i^{3D}$. The vectors are colored according to their length. In contrast to stereo cameras, Time-of-Flight (ToF) cameras have a good depth resolution so that the challenging estimation of a reliable z component of a velocity vector can be done in a satisfying way. Erroneous velocity vectors are estimated only at depth edges due to noise and inaccuracies of the optical flow computation. One can get rid of these outliers by applying a 5×5 median filter to $\{ v_x^i \}$, $\{ v_y^i \}$, and $\{ v_z^i \}$ separately as the velocity components can also be arranged in a 2D matrix. Figure 2.11(c) shows the smoothed 3D velocity field. The computation of 3D velocity vectors is utilized in Section 5.3.1 to track persons moving through the observed scene.

2.5.2 *Fusing Sets of Point Clouds*

So far, data has been recorded with a static camera. Due to limitations in the field of view of an arbitrary sensory system, agents will also try to acquire more data from the environment by moving the head simulating an eye saccade. The challenge is to fuse the acquired data to a consistent representation of the scene. In the case of 3D point clouds provided by the SwissRanger camera, this means that they have to be registered into one global world coordinate system. In [Swa07], I present a registration approach which fuses a sequence of frames acquired during an arbitrary tilting of the camera. The goal is to computed the rotation $\mathbf{R}_t \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t}_t \in \mathbb{R}^3$ of the camera made between two

12 http://www.cs.ucf.edu/~khan

13 http://server.cs.ucf.edu/~vision/source.html

consecutive frames $\mathcal{F}_t$ and $\mathcal{F}_{t-1}$. Referring to $\mathcal{F}_{t-1}$ as the *model* point set $\mathcal{B}$ and to $\mathcal{F}_t$ as the *data* point set $\mathcal{A}$ a set of point correspondences $\mathcal{C}$ has to be found

$$\mathcal{C} = \left\{ \left(k,\, l \right) \mid \vec{a}_k \in \mathcal{A},\, \vec{b}_l \in \mathcal{B} \right\}. \tag{2.21}$$

Ideally, these correspondences correlate with distinct 3D world points which belong to static scene parts and are located in the area covered by both views. The minimization of the mean square objective function $f_{\min}$ provides the optimal transformation $\left(\mathbf{R},\, \mathbf{t} \right)$ of the local coordinate system of $\mathcal{A}$ to the coordinate system of $\mathcal{B}$:

$$f_{\min}\left(\mathbf{R},\, \mathbf{t} \right) = \frac{1}{|\mathcal{C}|} \sum_{(k,l)\in\mathcal{C}} \left\| \vec{b}_l - \mathbf{R} \cdot \vec{a}_k - \mathbf{t} \right\|^2. \tag{2.22}$$

According to Schönemann [Sch66], the cross-covariance matrix $\mathbf{K}$ contains all necessary information to find the motion solution. Whereas, Lorusso [Lor97] has pointed out that a Singular Value Decomposition (SVD) of $\mathbf{K}$ solves the optimization problem with biggest accuracy and stability:

$$\begin{aligned} \mathbf{K} &= \sum_{(k,l)\in\mathcal{C}} \left(\vec{b}_l - \vec{b} \right) \cdot \left(\vec{a}_k - \vec{a} \right)^{\mathrm{T}} = \mathbf{V}\mathbf{D}\mathbf{U}^{\mathrm{T}} \quad \text{with} \\ &\quad \vec{a} = \frac{1}{|\mathcal{C}|} \sum_{(k,l)\in\mathcal{C}} \vec{a}_k, \quad \vec{b} = \frac{1}{|\mathcal{C}|} \sum_{(k,l)\in\mathcal{C}} \vec{b}_l, \end{aligned} \tag{2.23}$$

$$\mathbf{R} = \begin{cases} \mathbf{V}\mathbf{U}^{\mathrm{T}} & : \det(\mathbf{R}) = 1 \\ \mathbf{V}'\mathbf{U}^{\mathrm{T}} & : \det(\mathbf{R}) = -1 \end{cases} \quad \text{with} \quad \begin{aligned} \mathbf{V} &= \left(\vec{v}_1,\, \vec{v}_2,\, \vec{v}_3 \right) \\ \mathbf{V}' &= \left(\vec{v}_1,\, \vec{v}_2,\, -\vec{v}_3 \right) \end{aligned} \tag{2.24}$$

$$\mathbf{t} = \vec{b} - \mathbf{R} \cdot \vec{a} \tag{2.25}$$

My registration system consists of two steps: a coarse registration providing an initial transformation between two consecutive frames and a fine registration refining this initial guess. During coarse registration significant structures like, e. g., edges and corners, are extracted in the amplitude image of $\mathcal{B}$ using the structure tensor operator [För87]. The corresponding pixels in $\mathcal{A}$ are determined by applying the optical flow computation of Section 2.5.1 to the outstanding pixels. The resulting pixel correspondences can be transformed directly to the required 3D point correspondences $\mathcal{C}$ from which an initial transformation from $\mathcal{B}$ to $\mathcal{A}$ is computed using Equations 2.23 to 2.25. This initial guess is refined using the Picky Iterative Closest Points (ICP) approach proposed by Zinßer [Zino3]. Here, corresponding points are determined by estimating in $\mathcal{A}$ nearest neighbors for points in $\mathcal{B}$. This system has accomplished successful registration of test sequences with an average reconstruction error between 9mm and 86mm. As the development of this reconstruction system has been the topic of my Diploma thesis I refer for more details to my Diploma thesis [Swa06].

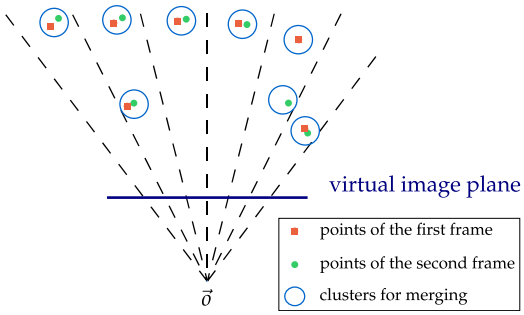


Figure 2.12: This figure shows a 2D visualization of a clustering of redundant points using a virtual plane. The dashed lines indicate the projection pyramids of the pixels on the image plane.

Instead, I am going to discuss a possible post-processing of the registered 3D point cloud. If perfect data is available and a optimal registration is possible the same world points of two different scans will have equal 3D coordinates. Due to sensor noise the mean square minimization runs into local minima resulting in noisy and thickened surfaces in the registered point cloud (→ Figure 2.13(b) and 2.13(g)). These surfaces contain redundant information. The goal is to thin out the point cloud while preserving structural information in the data and accumulating redundant information to increase the reliability of the final 3D coordinates. In [Swa08b], I present a *Virtual Image Plane Projection* (VIPP) for fusing several registered scans acquired while observing a vista space scene (→ Section 2.2.3). This means that the camera is tilted at most by 180°. My approach profits from the projection properties of range cameras and is inspired by reverse calibration [Bla95]. The general idea is to project the registered point cloud on a plane and to extract from discretization on the plane candidates for fusion. The registration system outlined above integrates all frames into the coordinate system of the first frame which forms the global coordinate system. Further, the intrinsic parameters of the camera are considered, either extracted from a parameter sheet or obtained by calibration: the focal length d_o, pixel width w, pixel height h, and principal point $(\, c_o,\, r_o \,)^{\mathrm{T}}$ (compare Figure 2.3). The image plane of the first frame is extended to an infinite *virtual plane* with pixel size and projection properties being equal to the original bounded image plane. The registered point cloud is positioned parallel to this virtual image plane in order to evenly distribute the points into the pixels. This is done by computing the two orthogonal principal axes via PCA. The axes span a plane which fits the point cloud with smallest least square error. The point cloud is rotated around its barycenter so that this major plane is located parallel to the virtual image plane. Now, each point $\vec{p} = (\, x,\, y,\, z \,)^{\mathrm{T}}$ is projected on the virtual plane by connecting the point with the optical center $\vec{o}$ of the virtual image plane. The corresponding the pixel $(\, c,\, r \,)^{\mathrm{T}}$ is located where this ray intersects the plane:

$$c \;=\; c_o + \frac{x}{\theta \cdot w}, \quad r \;=\; r_o + \frac{y}{\theta \cdot h}, \qquad \text{with } \theta \;=\; \frac{z}{d_o} \tag{2.26}$$

As can be seen in Figure 2.12 the projection pyramid of each pixel collects like a container points which form a set of candidates for fusion. The points have to be clustered locally to keep the additional information provided from multiple scans. Among the candidates of one container, clusters are found by means of Region Growing over the Euclidean distance [Ada94].

Finally, each cluster is replaced by its centroid. This approach keeps the property that objects near to the camera are sampled at a higher resolution than objects which are further away.

The thinning using Virtual Image Plane Projection (VIPP) is evaluated on two test sequences (→ Figure 2.13(a), Figure 2.13(f)). Both sequences consist of 11 frames recorded while turning the camera by 70°. For comparison, two other fusion techniques are implemented. Voxel Sampling (VS) discretizes the 3D space into voxels with size $20 \times 20 \times 20(\text{mm}^3)$. Points within one voxel are replaced by their centroid point. Fua's Patch Merging (FPM) uses the particle-based representation introduced in [Fua97]. The 3D points are binned into voxels. For each of these voxels a plane is estimated using PCA. The center of each voxel is projected onto the plane and the voxel is rejected if the projected center lies outside the voxel. Otherwise, the points in the voxel are replaced by the projected voxel center. In both test point clouds ground truth planes are extracted manually. A point assigned to a ground truth plane is colored according to its distance to this plane. The colors range from blue meaning small deviation (less than 10mm) to red meaning large deviation (more than 40mm). Figure 2.13 presents the ground truth planes in the original point clouds, (b) and (g), and in the thinned point clouds using VIPP, (c) and (h), FPM, (d) and (i), and VS, (e) and (j). The more points of a plane are colored in blue the smoother is the planar surface. Visually, VIPP seems to improve the planes in the sense of computing 3D points that form a smoother surfaces than those points computed by FPM and VS. For a quantitative analysis, the percentage of points per plane with a deviation smaller than 10mm is compared. A good thinning method will increase the percentage of points with a small deviation, the so-called smooth points. In Figure 2.14 for each plane four bars are plotted which describe per merging method the relative amount of smooth points (blue: no merging, orange: VIPP, yellow: FPM, green: VS). In most planes, VIPP achieves an increasing of the percentage of smooth points independent from the original smoothness of a plane. In average, the percentage of smooth points is increased by **5**% while the amount of points is reduced by ca. **64**%. The reduction rate of VIPP is implicitly given by the setup of the camera towards the scene and the amount of frames registered. Whereas, FPM and VS are less suited for fusing redundant data while improving or at least preserving the smoothness of surfaces. Their reduction rate is directly influenced by the voxel size which has to be specified explicitly. Also, they do not consider the different sampling rates of objects which dependent on their distance to the camera. To conclude, it can be stated that VIPP successfully merges redundant points in registered range data while preserving the objects' sampling rate and smoothing planar structures.

Even though, I have shown here that fusion of frames is possible, I have decided to examine in the following chapters the performance of frame-based modeling. If necessary, fusion is performed on a higher level by, e. g., classifier fusion (→ Section 3.3.3).

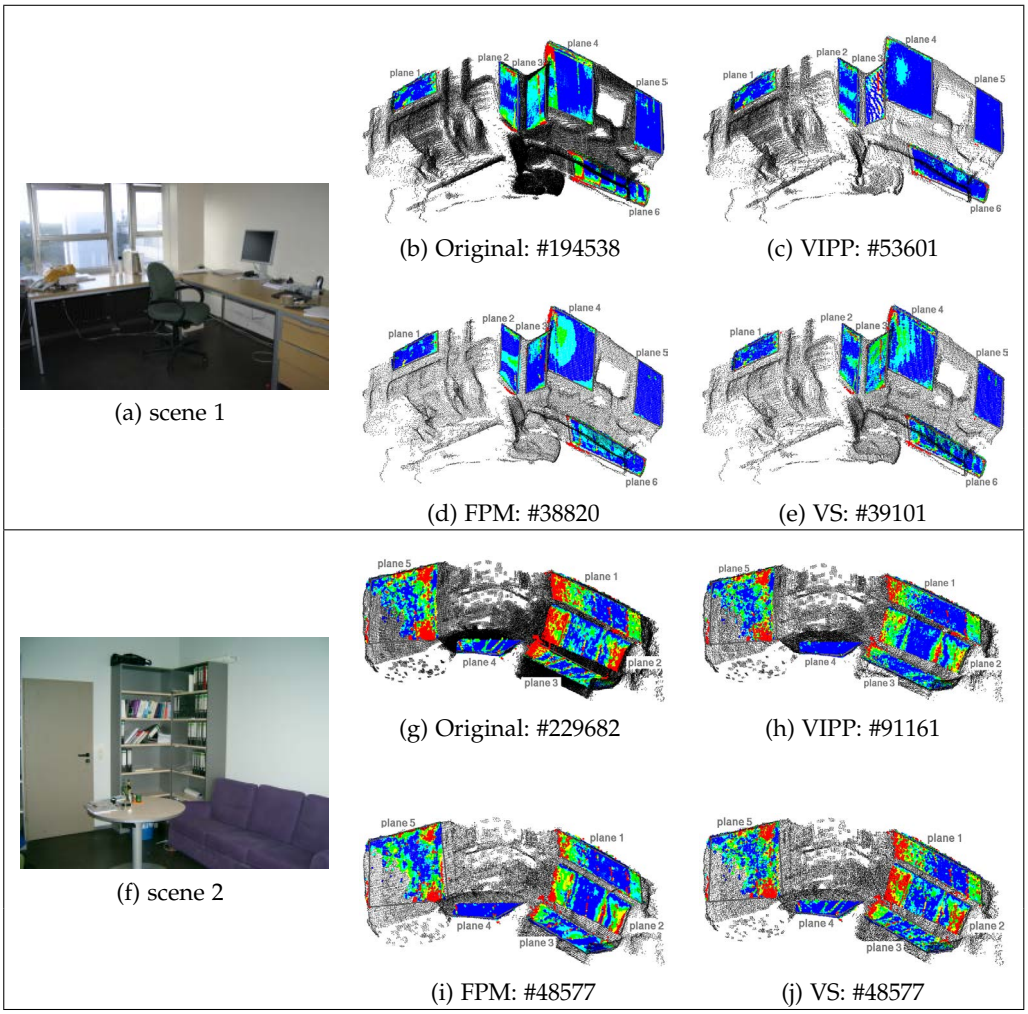


Figure 2.13: Subplots (a) – (e) belong to test scene 1 and (f) – (j) to test scene 2. Below each 3D plot the number of contained points is listed. The points are colored according to their deviation of the chosen ground truth planes. (a) and (f) visualize the test scenes and (b) and (g) show the original point clouds after registration. Thinned out point clouds using VIPP are given in (c) and (h), using FPM in (d) and (i), and using VS in (e) and (j).

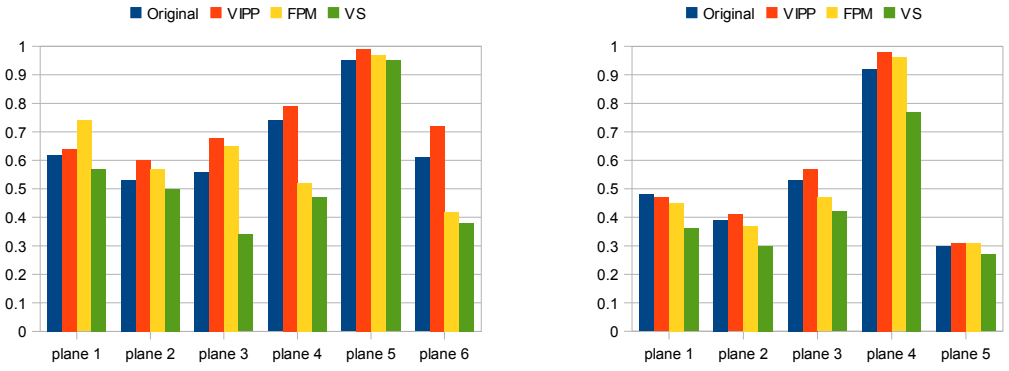
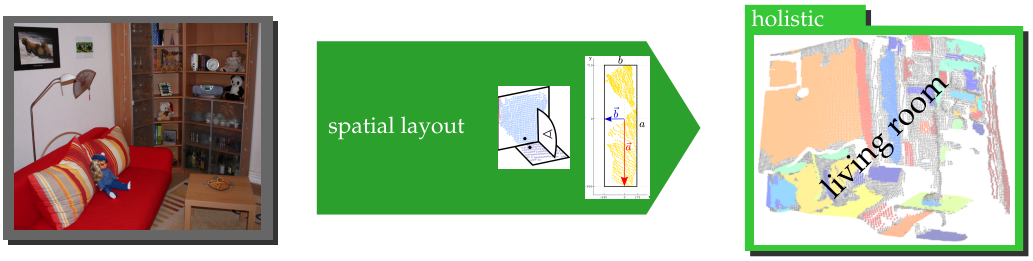


Figure 2.14: The bars plot for each test plane and each thinning method the percentage of smooth points. They are characterized by a deviation smaller than 10mm.

3

LEARNING HOLISTIC SCENE MODELS FROM SPATIAL LAYOUTS



Recognizing the type of an indoor room, e. g., "living room", is a basic spatial ability. The aim of this chapter is to realize indoor scene classification using the 3D spatial layout of rooms. In contrast to classical object based classification approaches, the focus on spatial layout provides a modeling of rooms that is independent from the specific design of rooms. I am going to introduce a 3D feature capturing characteristics of spatial room structures in a holistic way. Classifiers trained on these 3D features extracted from example rooms form the room models, the so-called *Holistic Scene Models*. This holistic scene models give a robot like BIRON the ability to categorize rooms perceived with a 3D sensor during a "home tour".

Section 3.1 motivates the use of spatial structures for solving the categorization task in domestic rooms. In Section 3.2 related approaches to scene classification of indoor rooms and scenes in general are presented. The 3D spatial feature defined on extracted planar surfaces capturing the spatial layout of a room is introduced in Section 3.3. Additionally, the learning stage, fusion of classification results over time, and possible combination schemes with the popular 2D Gist feature are presented. Section 3.4 evaluates diverse aspects of the indoor categorization system utilizing a special 3D database recorded in a regular IKEA home store and test sequences acquired in two real apartments. Section 3.5 summarizes the contributions and results of this chapter.

3.1 MOTIVATION

Any kind of holistic high-level concept of the surrounding is important for activating top-down knowledge that can guide the visual analysis in further tasks like, for example, enhancing object detection by context [Div09, Kim06, Tor03a]. A nowadays popular approach is the so-called *Gist* feature vector developed by Oliva and Torralba [Oli01]. It represents the *spatial envelope* of a 2D scene image and models the human ability of providing quickly a scene impression before recognizing any object in the scene. Evidence for this ability is given by several studies discussed in [Oli01]. Based on this general idea a variety of scene recognition approaches have been proposed that work well for outdoor scenes like "city", "street", "landscape", "mountains", etc. but, as reported by Quattoni and Torralba [Qua09], break down for indoor scenes like "kitchen", "living room", and so on. Therefore, Quattoni and Torralba have tackled the indoor problem by combining global information with local object information to achieve a better performance on different indoor categories. Although, they have achieved significant improvement the drawback of their method is that the training relies on a previous hand labeling of relevant regions of interests.

In general, object-based approaches need knowledge about interdependencies between objects and places. These interdependencies can be learned from training data [Vis09, Vas07c] or can be given as predefined ontologies [Zen08, Gal05]. Vasudevan et al. [Vas07b] have shown in their user study that an object based representation seems to be used by humans and might be useful for a robots in order to develop a human compatible representation of space. But the main problem is to define objects that are relevant for a certain room type. Some objects, e. g., a coffee machine in a kitchen, may be typical for a certain place but can be removed without changing the room type. While other objects like furniture, e. g., a sofa or a bed, form the functional and spatial layout of places but are hard to detect with conventional object detectors. Furthermore, psychological studies have shown that perceptual mechanisms of humans which are optimized for the room schemata rely on the spatial layout in general and not on the detection of specific objects. For example, Brewer and Treyen [Bre81] have found that with the perception of the room type, objects (e. g., books) are memorized to be in the experimental room (here, office) even though they have not been present and thus not perceivable. Nevertheless, the subject is able to determine the room type. Also, there exists a brain area, called Parahippocampal Place Area (PPA), that shows strong response to stimuli with spatial layout but does not respond to arrays of objects without three-dimensional spatial context [Eps98]. Henderson et al. [Hen08] have refined this finding in their fMRI studies showing that close-up views of scene-relevant objects (e. g., kitchen oven) produce less activation in this area than full-view indoor scenes, and that full-view indoor scenes produce more activation than outdoor scenes [Hen07]. This emphasizes the special nature of the indoor scene categorization problem and has motivated me to examine the contribution of the scene geometry to this problem.

As man-made environments mainly consist of planar surfaces and it is often assumed that objects lie on these surfaces, e. g., [Lee05]. I will investigate throughout this chapter whether generating holistic scene models from planar patches is a suitable approach. In contrast to my patch-based approach, point-based 3D features which are mostly used to classify single 3D points [Mun09a, Rus08, Tri07, Joh99] only encode local information. They are not applicable for classifying a point sets as a whole. A prove of concept for my 3D spatial features is given in [Swa08c]. Here, I have examined the performance in categorizing and recognizing 3 room types in a university, namely "office", "seminar room", and "corridor", using statistics defined on extracted planes [14]. Per category 2 different rooms have been selected with 300 frames acquired per room. One room per category has been used for training while the remaining rooms have formed the test set. A categorization rate of up to **0.81** is achieved for unknown rooms and a recognition rate of up to **0.99** for the known rooms. This performance has motivated me to extend the idea to broader classes, e. g., those of a flat (here: "bathroom", "bedroom", "eating place", "kitchen", "living room", and "office"). The remaining chapter focuses on the design of the global 3D feature and its combination with a local 2D feature. The goal is to acquire holistic scene models optimized for the indoor classification problem of percepts from the vista space.

14 The computed values and statistics slightly differ from those introduced later in this chapter. Only three characteristics have been considered: the patch size, the size ratio and the angles between patches. The size of a patch is estimated by the number of points establishing the patch. The patch shape is ignored. More details on the feature definition, the training and test rooms, and the classification results can be found in Appendix A.1.

3.2 RELATED WORK

This section reports on relevant work in the field of robotics and computer vision. Approaches from robotics mostly concentrate on recognizing and categorizing indoor scenes (→ Section 3.2.1). These approaches utilize data acquired with robot platforms driving around in an apartment or laboratory. Approaches from computer vision categorize 2D images showing any scene (→ Section 3.2.2). They rely on databases collecting images from the web. This images mostly show outdoor scenes like "building", "coast", or "mountains" but have been recently extended to indoor scenes like "store", "living room", "gym". Section 3.2.3 describes in more details two approaches which have been chosen for comparison in this chapter.

3.2.1 *From Robotics Perspective*

In the field of robotics, literature about scene classification has to main directions. One focuses on recognizing unknown rooms often with the purpose to enhance navigation allowing the robot to understand and execute commands like "go into the kitchen". The other direction aims at concept knowledge about indoor environments. Most approaches categorize room percepts based on the contained objects where the interdependencies between objects and rooms are either given top-down or learned in a bottom-up manner.

PLACE RECOGNITION. Early spatial abilities of robots have been developed in the context of determining drivable areas, e. g., encoded in a navigation map [Yua09]. It has been followed by decomposing such maps into places in general and providing labels to these places by recognizing known rooms. Places are mostly defined as some continuous area that is extracted by detecting transitions like doorways between two places [Zen08, Bee07] or by segmenting the open space into connected room-like places by, e. g., watersheding [Bus02]. The set of places and doorways can then be arranged to topological maps. The re-detection of known rooms is often realized by comparing 2D features in the current camera image to those in saved views [Ull08, Spe06] or by matching invariant features in laser scans [Top08]. Pronobis and colleagues provide indoor databases recorded from the view point of moving robots: INDoor Environment under Changing conditionS (INDECS) [15], Image Database for rObot Localization (IDOL) [16] [Pro10a], and COsy Localization Database (COLD) [17] [Ull08]. The databases contain 2D images of different rooms like "kitchen", "printer area", and "office" acquired by different robot platforms driving around in three different laboratories. The purpose of the databases is to capture rooms under varying illumination and whether conditions and changing environments. Using these databases room recognition experiments were performed using Composed

15 http://cogvis.nada.kth.se/INDECS
16 http://cogvis.nada.kth.se/IDOL
17 http://cogvis.nada.kth.se/COLD

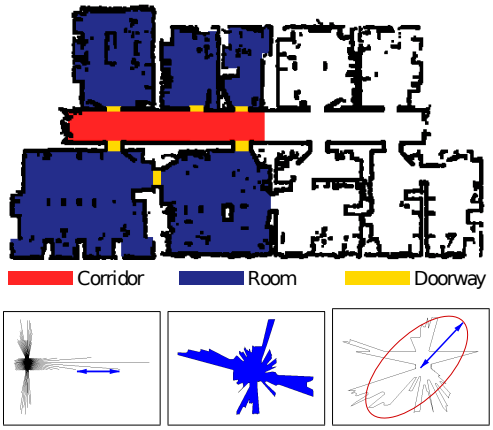


Figure 3.1: Mozos et al. has developed simple geometric features defined on laser scans (bottom row) to classify these scans as "corridor", "room", or "doorway" [Moz05].

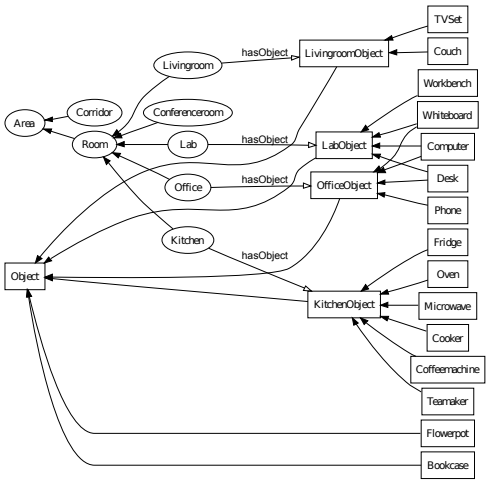


Figure 3.2: Zender et al. use the commonsense ontology of interdependencies between objects and room concepts for classifying a room as "kitchen", "lab", or "office" [Zen08].

Receptive Field Histograms (CRFH) as global features [Pro06] and Harris-Laplace detectors and Scale-Invariant Feature Transform (SIFT) descriptors as local features [Pro10a]. Further, global, local, and laser features have been fused to realize a multi-modal place classification [Pro10b].

PLACE CATEGORIZATION. A basic place categorization has been developed by Zender and colleagues [Zen08]. 360° Laser scans are categorized as "corridor", "hallway", "room" or "doorway" using simple geometrical features [Moz05]. Figure 3.1 shows some standard single-value geometrical features like average difference between the length of consecutive beams or area covered by the polygonal approximation of the beams. In a further step, they distinguish places recognized as rooms into finer concepts like "kitchen", "lab", or "office" using detected objects [Zen08]. They use the commonsense ontology of office environments displayed in Figure 3.2 which gives interdependencies between objects and room concepts. Noise in determining the room concept is treated by introducing consistency within a place using HMMs [Moz07] or Markov networks [Tri07]. A similar idea is proposed by Galindo et al. [Gal05]. They determine the semantic label (e. g., "bedroom") of an extracted place by inferring the room concept from detected objects via anchoring the objects in the conceptual hierarchy of an indoor ontology.

Instead of encoding the interdependency between object and room type in a top-down way through ontologies, it can be also learned from trainings data. Often graphical models are utilized for representing place through local object graphs [Vas07a, Kim06] or constellation models incorporating objects and their 3D positions [Ran07]. Figure 3.3 shows such hierarchical graphical model proposed by Kim and Kweon [Kim06]. From detected objects room labels are inferred using well established techniques like belief propagation or computation of joint posterior distributions [Vis09]. An important information for object

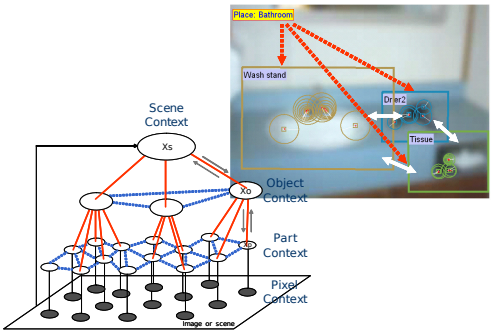


Figure 3.3: The hierarchical graphical model used by Kim et al. for representing visual scene context, objects, and parts [Kim06] is shown schematically and as an example.

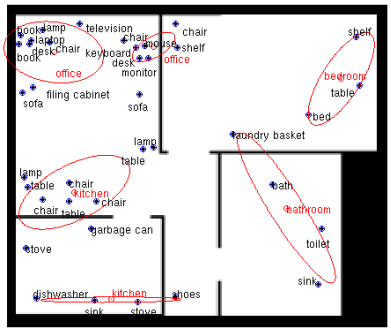


Figure 3.4: The spatial-semantic modeling system of Viswanathan and colleagues can cluster objects maps and label these clusters with according place names [Vis09].

based approaches is the occurrence of objects [Vas07d]. Some objects may be less informative as they appear across different room types like shown in Figure 3.4 or the absence of some objects is the significant information [Vas07c].

3.2.2 *From Vision Perspective*

A famous approach for real world scene recognition that bypasses segmentation of individual objects is proposed by Oliva and Torralba [Oli01]. Their procedure is based on a low dimensional representation, the *Spatial Envelope*. A set of perceptual dimensions represent the dominant spatial structure of a scene. These dimensions are naturalness, openness, roughness, expansion, and ruggedness. Figure 3.5 shows three dimensions on which man-made and natural scenes have differences. This modeling is referred to as Gist feature of a scene and allows a reliable categorization of outdoor scenes but has problems with indoor scenes. To tackle this problem, Quattoni and Torralba recently have extended the global Gist vector with local information prototypes [Qua09]. For each scene category prototype images are segmented into candidate regions for which histograms are computed. During categorization the candidate regions are allowed to move within a small window and the similarity between two regions is determined from the distance between the region histograms.

The Gist descriptor encodes properties of a scene on a global level in one feature vector. In contrast to that, a whole bunch of codebook based approaches is using visual words that encode local properties [Bos08, Laz06, Pos06, FF05, Pir04, Vog04]. Such local information are, for example, textures or intermediate themes as shown in Figure 3.6. Visual words are obtained by clustering of local features. As shown in Figure 3.7 the association between words and room types can be done by assigning probabilities which describe how likely a word can be found within a certain concept. Alternatively, the occurrence of words can be utilized [Bos08, Pos06, Vog04] similar to object occurrences described in Section 3.2.1.

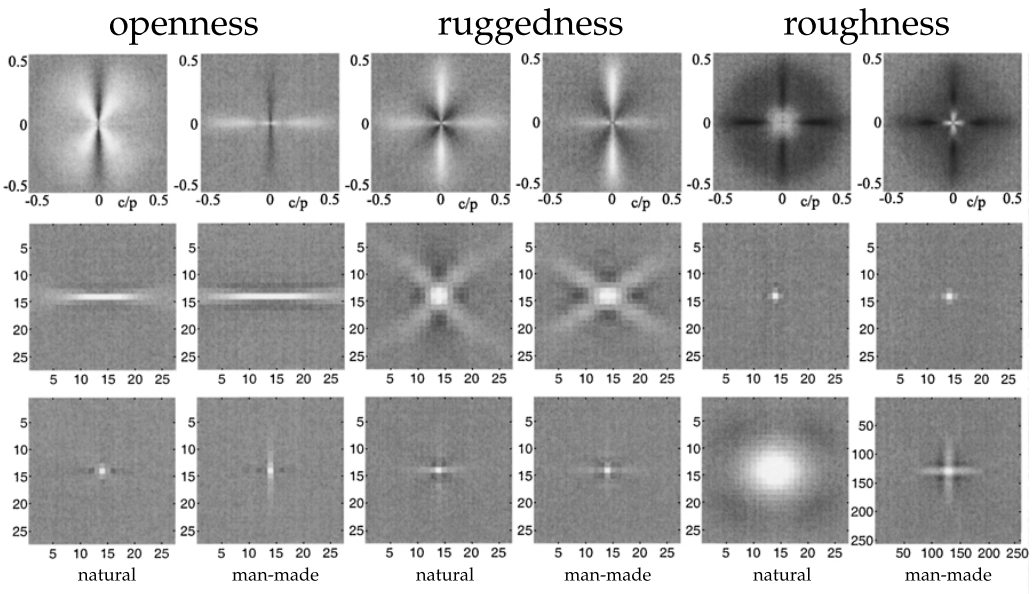


Figure 3.5: Dimensions of the spatial envelope approach proposed by [Olio1] are displayed. Differences between natural and man-made scenes can be observed.

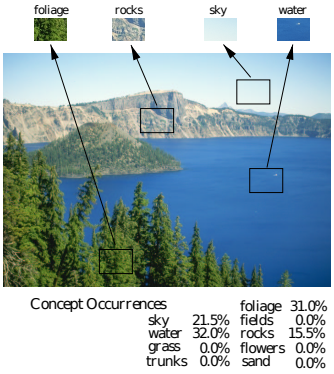


Figure 3.6: This example visualize some intermediate themes used by [Vog04] for natural scene categorization.

Texture	Furniture			Wall			Pavement	...
	Closet	Bed	Bookcase	Curtain	Wallpaper	Carpet	Brick	Tile
	0.02	0.13	0	0.01	0.15	0.12	0	0.04
	0	0.1	0	0.2	0.1	0.1	0	0
	0	0.02	0.1	0.1	0	0.01	0.02	0.27
	0.14	0.02	0	0	0	0	0.01	0
	0	0	0	0	0.12	0.1	0.37	0
	0.01	0	0	0.01	0.01	0	0.12	0.23
	0	0.01	0.17	0.02	0	0.01	0.01	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 3.7: This table shows an excerpt of Pirri's texture data where confidence vectors are kept with one element for each indoor object [Pir04]. These values are interpreted as an object belonging probability distribution.

3.2.3 *Approaches Chosen for Comparison*

From the previous sections I have chosen two approaches for comparison which are described here in detail. Originally, the COLD database presented in Section 3.2.1 has been designed to test recognition of known place under different illumination conditions. Place recognition is done on local features provided by a Harris-Laplace detector and a SIFT descriptor for which SVM models are trained using the match kernel [Wal03]. As the same rooms have been recorded in different laboratories Ullah and colleagues [Ull08] also tested whether their models can be used for categorizing unknown rooms. The models are trained on data from two laboratories and test on data from the third laboratory. Figure 3.8 shows the achieved performances. The rates result from classifying single images.

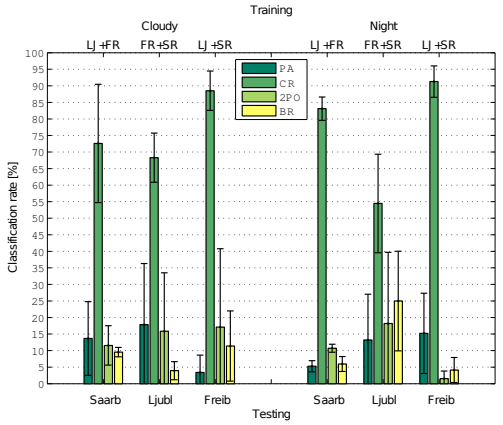


Figure 3.8: This diagram is taken from [Ull08]. It shows the classification rates of the room types "printer area" (PA), "corridor" (CR), "two-person office" (2PO), and "bathroom" (BR) from the COLD database. Single frames are categorized based on visual similarity. Frames from two sub-databases are used for training and from the remaining database for testing. The sub-databases are: 'SR' for Saarbrücken, 'FR' for Freiburg, and 'LJ' for Ljubljana.

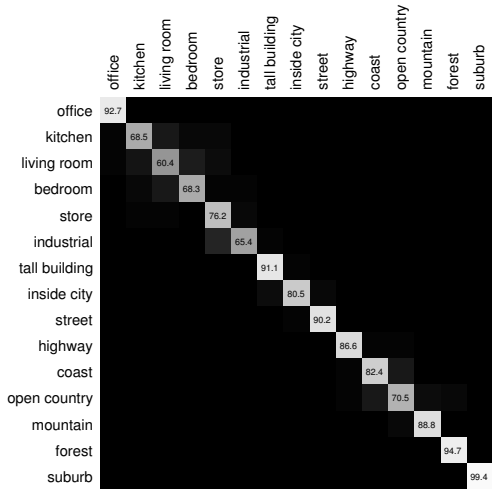


Figure 3.9: This confusion table shows on the diagonal the performance of the approach proposed by Lazebnik [Laz06]. A scene category database is tested which is mainly based on the database provided by [FF05]. It contains 15 categories with 200 to 400 images per category. The vocabulary size has been $M = 200$.

The "corridor" can be categorized with a rate of 0.76 while the rates for the other rooms are quite low (PA: 0.12, 2PO: 0.13, BR: 0.10).

As state-of-the-art computer vision approach that is based on local features I have chosen Lazebnik's approach [Laz06] which is on-line available [18]. In this work edge points at two scales and eight orientations are extracted as local features. Further, SIFT descriptors of 16×16 pixel patches are computed over a grid with spacing of 8 pixels. Then, the image is partitioned into increasingly finer sub-regions and histograms of local features found inside each sub-region are computed. The resulting *spatial pyramid* introduces an order to the so far orderless bag-of-features image representation. The histograms are computed based on a visual vocabulary which is formed by k-means clustering of features extracted in the training set. The set of histograms is concatenated to one vector on which they train a SVM. Figure 3.9 visualizes the performance of this approach as confusion table. They utilize a scene category database with 15 classes provided by [FF05] and originally collected by [Oli01]. Confusion occurs between the indoor classes (kitchen, bedroom, living room), and between some natural classes, such as coast and open country. The drawback of Lazebnik's approach is the high dimensionality of the final image descriptors. The vectors are even at coarser resolution still 8500-dimensional. Learning with such large vectors theoretically requires an enormous amount of training examples due to the curse of dimensionality and the empty space phenomena [Sco83, Bel61]. Also, classifying with a vector that large is time consuming. This limits an application of these features on a robot platform with limited resources and time constraints.

18 http://www.cs.unc.edu/~lazebnik/research/spatial_pyramid_code.zip

3.2.4 *Contribution of the Holistic Scene Model*

The holistic scene model provides 3D features capturing holistically the spatial layout of indoor rooms. This approach is independent from detection of specific objects and knowledge about interdependencies between objects and room types. The features are optimized for 3D data collected in indoor rooms of conventional apartments. The scene classification focuses on data that is typical for a robot's view. Each scene view is encoded by a light-weighted 3D feature vector that nicely complements the scene encoding with a classical Gist feature vector. A 3D indoor database is put together from 3D data collected in a main furniture store. This database is utilized to learn room type models on the basis of 3D features.

3.3 THE HOLISTIC SCENE REPRESENTATION

Figure 3.10 visualizes the main phases of my indoor scene classification problem. First, suitable features for given percepts have to be extracted. Second, for each class a classifier ($\rightarrow g_i$) is trained based on these features. Here, the set of classifiers $\{ g_i \}_{i=1}^{n}$ forms the holistic scene model. In the recognition phase the classification responses of each classifier are fused with a proper combination scheme ($\rightarrow E(\vec{d})$) to provide for a percept its class label (here, label of room). Section 3.3.1 presents the computation of a novel 3D *spatial* scene descriptor capturing the spatial layout of the 3D point cloud given by a SwissRanger frame. In Section 3.3.2 it is shown how the well known 2D *Gist* scene descriptor can be computed for a SwissRanger frame. As the goal is to achieve a robust scene categorization of data from a so far unseen room, Section 3.3.3 gives details on the learning of room models from these features. Further, a decision function optimized for the room type categorization problem is presented allowing a combination of single 3D spatial and 2D Gist feature responses and responses of these features over a couple of consecutive frames. I have presented this work at the Asian Conference on Computer Vision in 2010 [Swa10b].

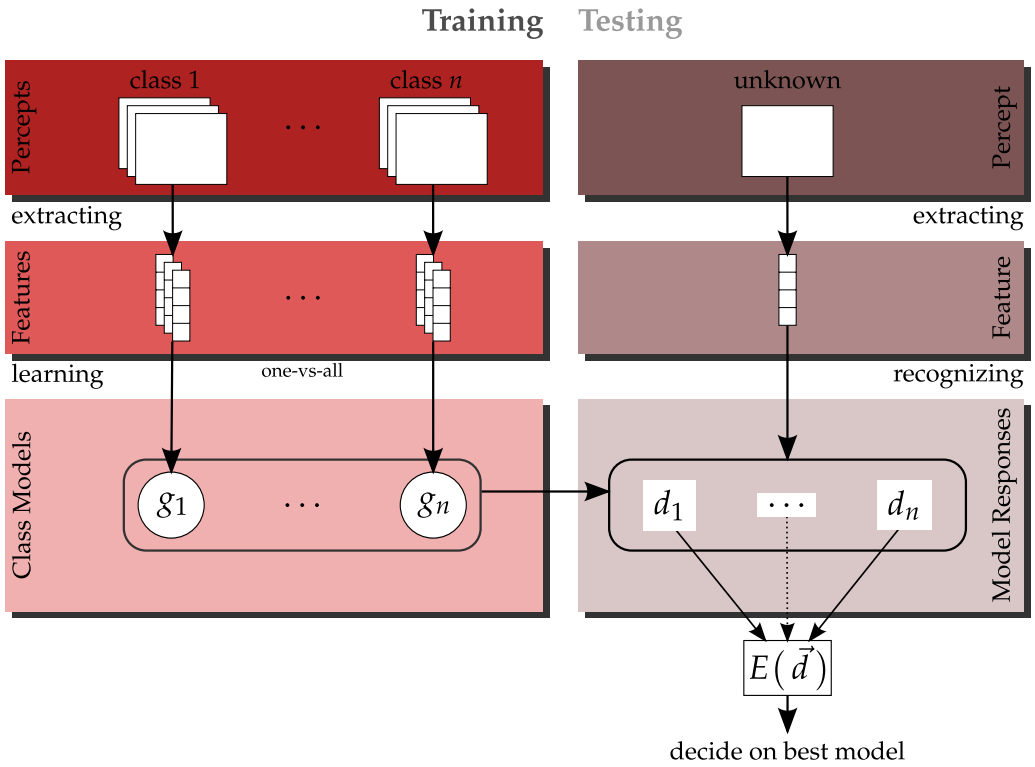


Figure 3.10: This flow chart describes schematically the important steps necessary during training classifiers and testing classification. For each percept a feature is extracted. A model per class is learned based on these features. For recognizing an unknown percept the extracted feature vector is fed into the class models and a decision is done upon the model responses.

3.3.1 *The Scene Descriptor from 3D Data*

For transforming a SwissRanger frame into an appropriate *3D spatial scene descriptor* the challenge is to find adequate statistics capturing the spatial layout. Thereby, certain requirements have to be considered like:

- *robustness to dynamic changes*: robustness against changes in the particular arrangement of furniture [Ull08],
- *robustness to non-categorical changes*: place models have to be independent from specific layouts, furniture's colors and textures, and
- *robustness to view changes*: robustness to view changes on the scene.

For encoding 3D scene geometry in a simple way independent from scene colors and textures, a set of planar patches is extracted from a 3D point cloud. Figure 3.11 shows a SwissRanger frame on which a set of m planar patches is extracted by decomposing the 3D point cloud into connected planar regions using the approach presented in Section 2.4.2. Defining appropriate features for a set of 3D planar patches could be inspired by shape analysis of 2D patches and feature definition based on 3D points. For example, Mozos [Moz05] has transformed 360° laser scans to 2D patches and has analyzed their shape in order to assign them to one of the three classes: "corridor", "room", or "doorway". The area covered by a patch and the shape of a patch are two basic properties which can be also computed for 3D planar patches. View changes and changes of their arrangement do not influence their size and shape. In the context of analyzing 3D data, point-based features have been used to recognize objects given as 3D scan [Hub04, Csá03, Het01, Joh99]. Local features defined for each 3D point have also been used to assign each point of a 3D scene to a predefined class like "vegetation", "facade", ... [Mun09a], "chair", "table", ... [Tri07], or "plane", "sphere", "cylinder", ... [Rus08]. As most of the cited work use surface normals, this information is also incorporated into my 3D scene descriptor. This is done by computing angles between patch normals. Patch normals are more robust to noise than normals computed for single 3D points. Further, histograms over angles between normals are invariant to view changes which is not the case for histograms over angles computed in relation to a vertical and a horizontal reference plane [Mun09b]. The angle between two patches captures the orientation of the patches to each other. While size and shape are properties of the patch itself, features capturing the relation between patches can be less found in literature. Therefore, the ratio of patch sizes is introduced as a further feature capturing the relationship between patches.

Abstractly spoken, the 3D feature vector

$$\vec{x}^{\,3D} \in \mathbb{R}^{25} \tag{3.1}$$

is computed from patch characteristics of a set of m planar patches $\{ \mathcal{P}_j \}_{j=1}^{m}$ which have been extracted from the 3D point cloud of frame $\mathcal{F}$.

Definition. ***Patch characteristics.***

Each planar patch element in $\left\{ \mathcal{P}_j \right\}_{j=1}^{m}$ *is characterized by:*

c_j^{s}, *the shape characteristic and* c_j^{A}, *the size characteristic.*

Each pair of planar patches in $\left\{ (\mathcal{P}_k, \mathcal{P}_l) \right\}_{(k,l)=(1,2)}^{(m-1,m)}$ *is characterized by:*

$c_{kl}^{\sphericalangle}$, *the angle characteristic and* $c_{kl}^{\div}$, *the size ratio characteristic.*

For each of the four sets of values

$$\mathcal{C}^{\mathrm{s}} = \left\{ c_j^{\mathrm{s}} \right\}_{j=1}^{m}, \qquad \mathcal{C}^{\mathrm{A}} = \left\{ c_j^{\mathrm{A}} \right\}_{j=1}^{m}, \tag{3.2}$$

$$\mathcal{C}^{\sphericalangle} = \left\{ c_{kl}^{\sphericalangle} \right\}_{(k,l)=(1,2)}^{(m-1,m)}, \qquad \mathcal{C}^{\div} = \left\{ c_{kl}^{\div} \right\}_{(k,l)=(1,2)}^{(m-1,m)}$$

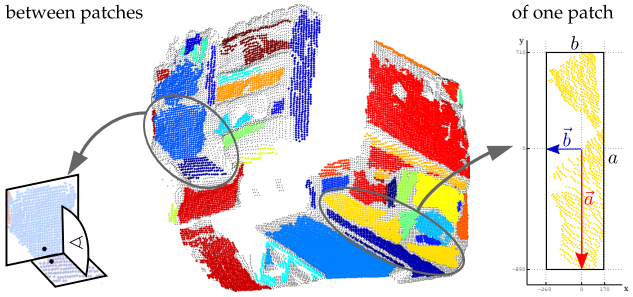
a histogram $\mathrm{H}(.)$ encoding the distribution of the values is computed and normalized to length 1. The four histogram vectors are concatenated to one feature vector $\vec{x}^{\,3\mathrm{D}}$ defining the *3D spatial scene descriptor*. Using plane pairs ensures independence from a specific spatial arrangement and from the orientation of the camera.

The computation of the wanted patch characteristics requires to known for each patch its orientation, the area covered, and the patch outline. The orientation of a patch is given through a normal vector $\vec{n}$ that is provided during the planar surface extraction presented in Section 2.4.2. The computation of the patch size and patch outline can be approximated by area and outline of the minimum bounding box enclosing the patch points. The box approximation has the advantage that area and outline of a box can be computed easily. For calculating the minimum bounding box of a patch, it is necessary to compute the direction along which the points have the largest variance. This direction determines the orientation of the bounding box with respect to the points that have to be enclosed. A suitable algorithm is the Principal Component Analysis (PCA) which provides three vectors: $\vec{a}$, $\vec{b}$, and $\vec{n}$ with

- $\vec{a}$ indicating the direction of the largest variance in the data,
- $\vec{b}$ the direction orthogonal to $\vec{a}$ with the second largest variance, and
- $\vec{n}$ the normal vector of the planar patch.

The right close-up of Figure 3.11(b) shows a planar patch $\mathcal{P}$ transformed so that $\vec{a}$ and $\vec{b}$ are parallel to the coordinate axes x and y. The edges of the resulting minimum box are parallel to the coordinate axis denoting the length of the edge parallel to vector $\vec{a}$ by a and of the other edge by b. The following paragraphs describe in more details how to compute the plane characteristics mentioned above and how to transform them to a feature vector.

(a) SR amplitude image



(b) extracted 3D planes and their properties

(c) computed 3D spatial feature $\vec{x}^{3D}$

Figure 3.11: This figure shows an example output of the SwissRanger camera (frame 162 of living room `liv.5`) – (a): the amplitude image, (b): the 3D point cloud with points highlighted by color according to their patch membership; the left close-up shows the acute angle between two patches; the right close-up shows a plane transformed so that the vectors indicating the two largest variance directions in the data are parallel to the coordinate axis. (c) visualizes the 3D spatial feature vector $\vec{x}^{3D}$ computed based on the set of planes presented in (b).

THE SHAPE CHARACTERISTIC c^s. To compute the shape of a patch $\mathcal{P}$ the estimated minimum 2D bounding box is utilized. Considering the length of the edges which are denoted by a and b, the shape characteristic computes from

$$c^s = s\Big(\mathcal{P}\Big) = \frac{\min(a,b)}{\max(a,b)}. \tag{3.3}$$

The possible values for c^s are lying between 0 and 1. The smaller c^s is the more elongated is the shape, a value near 1 denotes a quadratic shape. For a set of patches a set of shape values is computed:

$$\Big\{\mathcal{P}_j\Big\} \xrightarrow{s(.)} \mathcal{C}^s = \Big\{c_j^s\Big\}, \quad j = 1, \ldots, m. \tag{3.4}$$

For the histogram H_s over these values the range $[0,1]$ is divided into five bins $h_1, \ldots, h_5$ with bin width $\Delta h = 0.2$. The histogram vector is then transformed to a feature vector by normalizing its length to 1:

$$\vec{x}^s = \frac{1}{m} \cdot H_s\big(\mathcal{C}^s\big), \quad \text{with} \tag{3.5}$$

$$H_s\big(\mathcal{C}^s\big) = \begin{pmatrix} \sum_{j=1}^{m} \delta(c_j^s, h_1) \\ \vdots \\ \sum_{j=1}^{m} \delta(c_j^s, h_5) \end{pmatrix}, \quad \delta(c_j^s, h_i) = \begin{cases} 1 & : c_j^s \in h_i \\ 0 & : \text{else} \end{cases}. \tag{3.6}$$

This feature vector encodes for a room whether there are a lot of elongated structures or more quadratic like or a mixture of both. The introduced encoding of the shape slightly differs from the standard encoding of the shape of an arbitrary 2D patch which is $s = \frac{U^2}{4\pi \cdot A}$ [ASMoo] where U is the outline of the region and A its area. In the case of 2D boxes, the content of both values c^{S} and s is equivalent as there exists a strictly monotonic function $f(x) = \frac{1}{\pi} \cdot (2 + x + \frac{1}{x})$ that allows bijective mapping between them. The advantage of the shape factor c^{S} is as mentioned above the built-in normalization because c^{S} can only take values in the interval of $]0, 1]$. The derivation of the function $f(x)$ can be looked up in Appendix A.2.

THE SIZE CHARACTERISTIC c^{A}. The area covered by a patch $\mathcal{P}$ is computed using its minimum bounding box:

$$c^{\mathrm{A}} = \mathrm{A}\Big(\mathcal{P}\Big) = a \cdot b, \tag{3.7}$$

$$\Big\{ \mathcal{P}_j \Big\} \xrightarrow{\mathrm{A}(.)} \mathcal{C}^{\mathrm{A}} = \Big\{ c_j^{\mathrm{A}} \Big\}, \quad j = 1, \ldots, m. \tag{3.8}$$

This estimation of the patch size is more reliable compared to our initial approach used in [Swao8c] where the number of points assembling the patch has been used. Assuming two patches containing the same amount of points, the patch localized nearer to the camera will cover due to general projection properties a smaller area in 3D. As the *size values* $\{c_j^{\mathrm{A}}\}$ are not normalized the histogram $\mathrm{H_A}$ consists of $h = 6$ bins $h_1, \ldots, h_6$ with the following boundaries in cm^2: $[0, 25^2[$, $[25^2, 50^2[$, $[50^2, 100^2[$, $[100^2, 200^2[$, $[200^2, 300^2[$, and $[300^2, \infty[$. The first interval captures small objects located on/in the furniture, the second interval bigger objects and small furniture or furniture parts, and the third interval mid-size furniture. The remaining intervals aim for the large spatial structures. A further distinction on patches larger than $(300\mathrm{cm})^2$ is not necessary as such big patches rarely occur in the case of perceiving indoor environments. The corresponding feature vector computes as follows:

$$\vec{x}^{\mathrm{A}} = \frac{1}{m} \cdot \mathrm{H_A}\big(\mathcal{C}^{\mathrm{A}}\big), \quad \text{with} \tag{3.9}$$

$$\mathrm{H_A}\big(\mathcal{C}^{\mathrm{A}}\big) = \begin{pmatrix} \sum_{j=1}^{m} \delta(\, c_j^{\mathrm{A}}, h_1 \,) \\ \vdots \\ \sum_{j=1}^{m} \delta(\, c_j^{\mathrm{A}}, h_6 \,) \end{pmatrix}. \tag{3.10}$$

It captures the occurrence of small, medium-size, and large patches.

THE SIZE RATIO CHARACTERISTIC $c^{\div}$. Considering the computed sizes ($\rightarrow$ Equation 3.7) of two patches $(\mathcal{P}_1, \mathcal{P}_2)$ the size ratio is computed by:

$$c_{12}^{\div} = \mathrm{R}\Big(\mathcal{P}_1, \mathcal{P}_2\Big) = \frac{\min(c_1^{\mathrm{A}}, c_2^{\mathrm{A}})}{\max(c_1^{\mathrm{A}}, c_2^{\mathrm{A}})} = c_{21}^{\div}, \tag{3.11}$$

$$\Big\{ \big(\mathcal{P}_k, \mathcal{P}_l\big) \Big\} \xrightarrow{\mathrm{R}(.)} \mathcal{C}^{\div} = \Big\{ c_{kl}^{\div} \Big\}, \quad (k,l) = (1,2),\ \ldots,\ (m-1,m). \tag{3.12}$$

A value near 0 means that two planes significantly differ in their size whereas two planes cover the same area if the value is near 1. The histogram $\mathrm{H}_{\div}$ for the *size ratio values* $\{c_{kl}^{\div}\}$ is designed equally to histogram $\mathrm{H_s}$ (dividing the range $[\,0,1\,]$ in five bins $h_1, \ldots, h_5$ of width $\Delta h = 0.2$). Given m planes, $\frac{m\ (m\ -\ 1)}{2}$ number of plane pairs can be chosen. The feature vector is calculated through:

$$\vec{x}^{\div} = \frac{2}{m(m-1)} \cdot \mathrm{H}_{\div}\big(\mathcal{C}^{\div}\big), \quad \text{with} \tag{3.13}$$

$$\mathrm{H}_{\div}\big(\mathcal{C}^{\div}\big) = \begin{pmatrix} \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} \delta(\, c_{kl}^{\div},\, h_1\,) \\ \vdots \\ \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} \delta(\, c_{kl}^{\div},\, h_5\,) \end{pmatrix}. \tag{3.14}$$

This feature vector focuses on the question whether a room contains a lot of similar sized patches or not. As can be seen in Figure 3.12 the only size ratio can separate the "eating place" class from the other room classes. This feature is unique for this class because an eating place can be thought of a set of equally sized small patches which are the chairs and one big patch which is the table.

THE ANGLE CHARACTERISTIC $c^{\sphericalangle}$. A feature vector based on angles between planar patches encodes the geometric configuration of the presented scene, e. g.whether there are a lot of patches parallel or orthogonal to each other. For two planes $(\mathcal{P}_1, \mathcal{P}_2)$ the angle characteristic $c_{12}^{\sphericalangle}$ holds the acute angle between them. It is computed by applying Equation 3.15 to their normals $\vec{n}_1$ and $\vec{n}_2$. A scalar product of these two vectors is computed and the arc-cosine of the result delivers the corresponding angle. The left close-up of Figure 3.11(b) shows an example of an angle between two patches.

$$c_{12}^{\sphericalangle} = \sphericalangle\Big(\mathcal{P}_1, \mathcal{P}_2\Big) = \begin{cases} \arccos(\vec{n}_1 \cdot \vec{n}_2) & :<= \frac{\pi}{2} \\ \pi - \arccos(\vec{n}_1 \cdot \vec{n}_2) & : \text{else} \end{cases} \tag{3.15}$$

$$\Big\{ \big(\mathcal{P}_k, \mathcal{P}_l\big) \Big\} \xrightarrow{\sphericalangle(.)} \mathcal{C}^{\sphericalangle} = \Big\{ c_{kl}^{\sphericalangle} \Big\}, \quad (k,l) = (1,2),\ \ldots,\ (m-1,m). \tag{3.16}$$

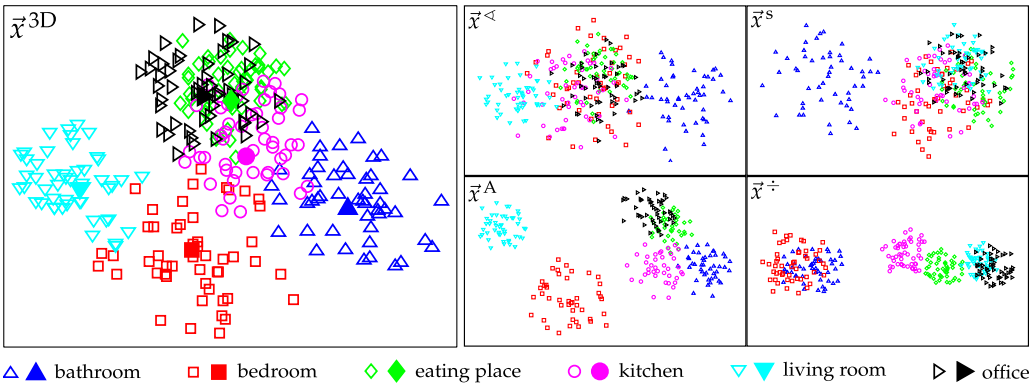


Figure 3.12: (left) The 25-dimensional 3D spatial features $\vec{x}^{3D}$ extracted on frames of the 3D database are plotted in 2D using the multidimensional scaling method. Per category the mean feature vector is computed and those 50 feature vectors are extracted which are nearest to this mean vector. The mean feature vectors are labeled by ▲, ■, ♦, ●, ▼, ▶. The Euclidean distance is used as inter-point distance. Four of six categories (bathroom, bedroom, kitchen, living room) are already in 2D nicely clustered. (right) shows plots of the sub-feature vectors $\vec{x}^{\triangleleft}$, $\vec{x}^{s}$, $\vec{x}^{A}$, and $\vec{x}^{\div}$. They give an impression by which plane property a class separation may be caused.

The corresponding histogram $H_{\triangleleft}$ over the set of *angle values* $\{c_{kl}^{\triangleleft}\}$ consists of nine bins $h_1, \ldots, h_9$ between 0 and $\frac{\pi}{2}$ with a bin width of $\Delta h = \frac{\pi}{18}$. The feature vector results from:

$$\vec{x}^{\triangleleft} = \frac{2}{m(m-1)} \cdot H_{\triangleleft}(\mathcal{C}^{\triangleleft}), \quad \text{with} \tag{3.17}$$

$$H_{\triangleleft}(\mathcal{C}^{\triangleleft}) = \begin{pmatrix} \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} \delta(c_{kl}^{\triangleleft}, h_1) \\ \vdots \\ \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} \delta(c_{kl}^{\triangleleft}, h_9) \end{pmatrix}. \tag{3.18}$$

This feature is best suited to separate between cluttered rooms like a bedroom and less cluttered rooms like a bathroom or a corridor.

CONCATENATION. The above partial feature vectors are concatenated to a 25-dimensional feature vector, the so-called *3D-based spatial feature vector*,

$$\vec{x}^{3D} = \begin{pmatrix} \vec{x}^{\triangleleft} \\ \vec{x}^{s} \\ \vec{x}^{A} \\ \vec{x}^{\div} \end{pmatrix}. \tag{3.19}$$

It captures the spatial properties of the planar patches in a scene, like the orientation of the patches to each other, their shapes, and their size characteristics. Figure 3.11(c) visualizes the 3D feature vector computed from the set of planes

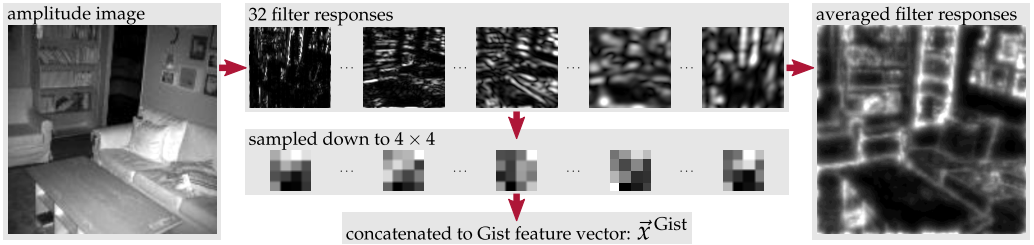


Figure 3.13: This figure shows 32 response images acquired by applying 32 Gabor filters to the input image, a SwissRanger amplitude image. The input image is a 144×144 clipping of the original 176×144 amplitude image resized to the size of 256×256. The response images are sampled down to 4×4 and concatenated to one 512-dimensional Gist feature vector $\vec{x}^{\text{Gist}}$. Averaging over the response images gives an impression of the information encoded.

shown in Figure 3.11(b). Further, the quality of the separation of different room categories in the feature space is shown on features extracted from the IKEA database (→ Section 3.4.1). For each frame in the IKEA database the 3D feature vector is extracted. For each room category a mean feature vector is computed and 50 feature vectors are selected which are closest to the mean. Figure 3.12 visualizes this subset of feature vectors in 2D by a classical multidimensional scaling [Kru78] which computes a projection of features in, e. g., 2D, while trying to preserve the original inter-point distances. There are four clusters clearly visible which corresponds with the room categories bathroom, bedroom, kitchen, and living room. Only office and eating place are not clearly distinguishable in the 2D plot. Recalling the spatial layouts of the two room types they share some similarities which are that at least a table and a chair are contained in a room being an office or an eating place. In contrast to the 2D plot of the Gist features (see Section 3.3.2 and Figure 3.14) the living room features form a compact cluster clearly separated from the other clusters. This explains the substantial contribution of the 3D features in categorizing percepts from living rooms as observed during evaluation (see Section 3.4). In Figure 3.12, the plots of the sub-features $\vec{x}^{\sphericalangle}$, $\vec{x}^{\text{s}}$, $\vec{x}^{\text{A}}$, and $\vec{x}^{\div}$ give an impression which plane property describes a room class best. For example, angle, shape, and size characteristic separate "bathroom" percepts from other percepts. "Bedroom" percepts only differ clearly from other rooms when the size characteristic is observed and "eating place" percepts when the size ratio characteristic is observed. In both characteristics also a definite "kitchen" cluster is visible. A clear "office" cluster is not noticeable in the defined patch characteristics. A further analysis of the correlations between the sub-vectors is given in Section 3.4.7.

3.3.2 *The Scene Descriptor from 2D Data*

Torralba [Toro3b] has developed a powerful feature for scene classification in 2D. The *Gist feature vector* is a further approach to encode holistically spatial characteristics of a scene. Therefore, it is worth investigating whether it can extend the information captured by the previously defined 3D spatial feature vector. The Gist computation relies on a wavelet image decomposition. Each image

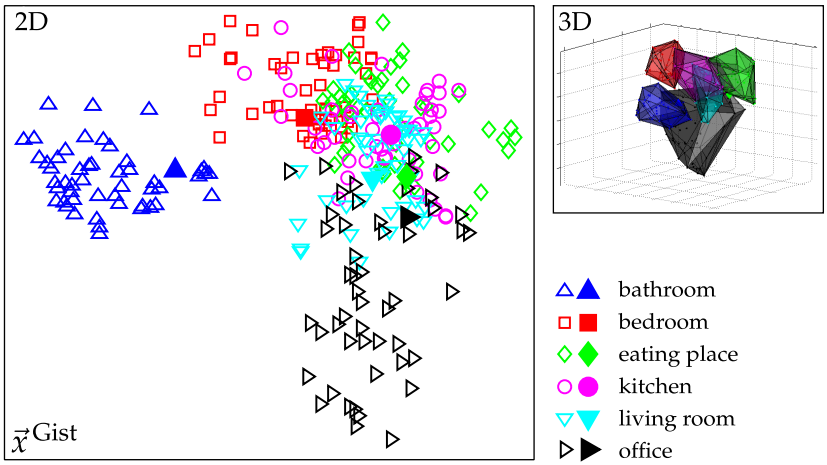


Figure 3.14: Using multidimensional scaling the 512-dimensional Gist feature vectors is plotted in 2D. Per category the mean feature vector (labeled by ▲, ■, ♦, ●, ▼, ►) and the 50 closest vectors are displayed. The clusters of bathroom, bedroom, and a sub-cluster of the office category are already nicely separated in 2D even though plotting 512-dimensional feature vectors under preserving their inter-point distances is a hard problem. The sub-window shows the 3D plot of the features with convex hulls enclosing features of one category. Here, the features of the eating place category occurred to be linear separable from the other features while living room and kitchen features are still mixed.

location is represented by the output of filters tuned to different orientations and scales. As the SwissRanger ToF camera delivers besides a 3D point cloud an amplitude image generated from the amount of infra-red light reflected, the implementation of the Gist features [19] can be applied directly to the amplitude image. Here, I have used 8 orientations and 4 scales of the Gabor filters applied to a 144×144 clipping of the 176×144 amplitude image resized bilinearly to 256×256. The clipping is anchored in the center of the original image. The resulting representation is sampled down to 4×4 pixels resulting in a dimensionality of $8 \times 4 \times 16 = 512$ for $\vec{x}^{\text{Gist}}$. Figure 3.13 shows some filter responses resulting from convolving the input image with the 32 different Gabor filters. While the 3D feature vector $\vec{x}^{\text{3D}}$ captures information about the arrangement of planar patches in the scene, $\vec{x}^{\text{Gist}}$ encodes additional global scene information on the level of edges. The bright pixels in the averaged response image visualize the peaks of the wavelet responses.

Figure 3.14 shows a 2D plot of some Gist features computed on amplitude images in the 3D IKEA database. Per category the mean feature vector and the 50 closest vectors are displayed. The clusters for bathroom, bedroom, and a subpart of the office category are already in the 2D nicely separated. As plotting 512-dimensional feature vectors in 2D under preserving their inter-point distances is a hard problem I have also examined the 3D feature plot. It turns out that the eating place forms a compact cluster in 3D while kitchen, a subpart of the office cluster, and especially the living room cluster have still intersections and are not linearly separable. As mentioned above, this impression explains to a certain point the poor categorization performance of Gist features for "living room"-percepts.

19 http://people.csail.mit.edu/torralba/code/spatialenvelope/

3.3.3 *Training Room Models and Combining Single Classifications*

The next step after computing suitable features is to train classifiers that estimate the boundaries between the different classes. Theoretically, for each class a discriminant model has to be learned which can be used to compute the probability that a feature belongs to a class. For example, this probability could dependent on the distance of a feature vector to the class boundary. This set of classifiers form the *holistic scene model*. Formally written, for a set of 6 classes

$$\Omega = \{\omega_i\}_{i=1,\ldots,6} \quad \text{with} \tag{3.20}$$

$$\omega_1 \mathrel{\widehat{=}} \texttt{bath.},\ \omega_2 \mathrel{\widehat{=}} \texttt{bed.},\ \omega_3 \mathrel{\widehat{=}} \texttt{eat.},$$
$$\omega_4 \mathrel{\widehat{=}} \texttt{kit.},\ \omega_5 \mathrel{\widehat{=}} \texttt{liv.},\ \omega_6 \mathrel{\widehat{=}} \texttt{off.}$$

a vector of discriminant functions $G(\vec{x})$ is learned where each function $g_i(\vec{x})$ maps the n-dimensional feature vector $\vec{x}$ on a scalar value d_i encoding how likely $\vec{x}$ lies in class ω_i [Kuno4]:

$$\begin{aligned} G &: \mathbb{R}^n \to \mathbb{R}^{|\Omega|} \\ g_i &: \mathbb{R}^n \to \mathbb{R}, \quad i = 1, \ldots, |\Omega| = 6 \\ G(\vec{x}) &= \begin{pmatrix} g_1(\vec{x}) \\ \vdots \\ g_6(\vec{x}) \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_6 \end{pmatrix} \quad \text{with, e. g., } d_i = \begin{cases} > 0 & : \vec{x} \in \omega_i \\ < 0 & : \vec{x} \notin \omega_i \end{cases}. \end{aligned} \tag{3.21}$$

The result for an input feature vector $\vec{x}$ is a $|\Omega|$-dimensional vector $\vec{d}$ (here: 6-dimensional) which holds for each class the likelihood that $\vec{x}$ is in this class. Finally, classification of a feature vector $\vec{x}$ means to perform a decision based on the likelihood stored in $\vec{d}$. In the simplest case, the maximum value is determined and the corresponding class is selected to be the classification answer. Mathematically, this can be formulated by a decision function $E(.)$ which maps the vector $\vec{d}$ on a binary vector $\vec{e}$ with the i_1-th component equal to 1 if $i_1 = \arg\max_i d_i$ is the maximum value in $\vec{d}$. All other components are set to 0:

$$E(\vec{d}) = \begin{pmatrix} e_1 \\ \vdots \\ e_6 \end{pmatrix}, \quad e_i = \begin{cases} 1 & : i = i_1 \\ 0 & : \text{else} \end{cases}, \quad i_1 = \arg\max_i d_i. \tag{3.22}$$

A classification result based on a single feature vector might not be very reliable because a SwissRanger frame contains only a small part of the recorded room due to the limited view field of the camera. Fortunately, our envisioned scenario allows for a stabilization of the room categorization over several consecutive

frames as the robot would record with its camera a sequence of frames from the unknown room while tilting and panning the camera. By considering the classification results of a sequence of consecutive frames

$$\{ \mathcal{F}_{t-i} \}_{i=0}^{\Delta t-1} \tag{3.23}$$

in a time window $[\, t - \Delta t,\, t \,]$ it is assumed that the decision becomes more reliable and less vulnerable to uninformative views on the scene like, e. g., walls, ceiling, floor or furniture that is spread over all room types. Seeing the set of classifications $\{ \vec{d}_{t-i} \}_{i=0,\ldots,\Delta t-1}$ as results of independent classifiers simple classifier combination schemes are an obvious choice [Kuno2]. According to Kittler's theoretical framework [Kit98] the fusion can then be done through the product, sum, max, min, median, or majority vote rule. Among those listed, majority voting is very popular due to its simplicity and has demonstrated experimentally and theoretically its effectiveness in combining individual classifiers [Naro5, Lam97]. In my case, fusing by majority voting means that classification results $\{ \vec{e}_{t-i} \}_{i=0,\ldots,\Delta t-1}$ achieved during a time window $[\, t - \Delta t,\, t \,]$ are summed to a new voting vector $\vec{d}^{*}$. The decision function $E(.)$ is then applied a second time on this accumulated distance vector:

$$\begin{aligned} \vec{e}^{*} &= E\left(\vec{d}^{*}\right), \\ \vec{d}^{*} &= \sum_{i=0}^{\Delta t-1} \vec{e}_{t-i}, \quad \vec{e}_{t-i} = E\left(\vec{d}_{t-i}\right), \quad \vec{d}_{t-i} = G\left(\vec{x}_{t-i}\right). \end{aligned} \tag{3.24}$$

$\vec{x}_{t-i}$ is the feature vector encoding characteristics of frame $\mathcal{F}_{t-i}$.

In the following, I am going to present two different fusion schemes, V^S and V^M optimized for the indoor scene classification problem during a "home tour" of a robot. V^M slightly extends the classical majority voting by introducing weights for the single classifiers (here, classification answers over time). In an experimental comparison, Kittler and colleagues [Kit98] have shown that the sum rule outperforms the other classifier combination schemes. Therefore, voting scheme V^S is based on the sum rule where the class hard decision on the frame-level is skipped. The class decision is just taken on the sum of normalized model answers. This allows to pass the support for all classes to the final classification. The benefit of this scheme arises in cases where some frames give more or less equal support for two or more classes. The idea is to lower the risk of making a wrong decision by relying on frames in Δt which enable a clearer decision.

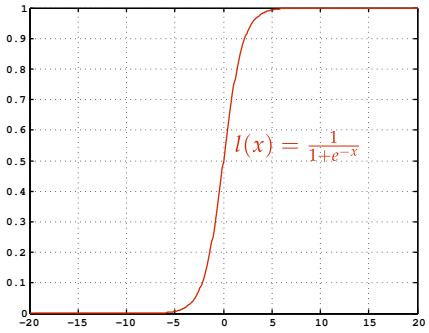


Figure 3.15: The logistic function $l(x) = \frac{1}{1+e^{-x}}$ is plotted for visualization on the range $[-20, 20]$.

THE VOTING SCHEME V^M. In some cases, several classes are similar likely as $\vec{d}$ hold similar values. In other cases, clear class decision can be taken as one value in $\vec{d}$ is clearly outstanding. This means that there are views on a scene which are more or less informative which means that the quality of the decision varies. This can be incorporated in the classical majority voting scheme by weighting each decision vector $\vec{e}_{t-i}$ with a factor α_{t-i} while summing over the window $[t - \Delta t, t]$:

$$\begin{aligned} \vec{e}^M &= E\left(\vec{d}^M\right), \\ \vec{d}^M &= \sum_{i=0}^{\Delta t-1} \alpha_{t-i} \cdot \vec{e}_{t-i}, \quad \alpha_{t-i} = A\left(\vec{d}_{t-i}\right), \quad \vec{e}_{t-i} = E\left(\vec{d}_{t-i}\right). \end{aligned} \tag{3.25}$$

An intuitive definition for the reliability of a classification decision $\vec{e}$ is given for a distance vector $\vec{d}$ by the difference between the distance value of the most supported class $d_{i_1} = \max_i d_i$ and the distance value of the second best class $d_{i_2} = \max_{i \setminus i_1} d_i$. The bigger the difference the more reliable is the decision. For comparison reasons, a normalization function $l(.)$ is applied to the distance values. As can be seen in Figure 3.15 the logistic function $l(x)$ is a strictly monotonic function mapping values of the interval $]-\infty, \infty[$ to values out of $]0, 1[$:

$$l(x) = \frac{1}{1+e^{-x}}, \qquad \lim_{x \to -\infty} l(x) = 0, \quad \lim_{x \to \infty} l(x) = 1. \tag{3.26}$$

Due to the steepest gradient around 0 constellations like $d_{i_1} > 0$ and $d_{i_2} < 0$ are higher weighted than those with $d_{i_1}, d_{i_2} > 0$ or $d_{i_1}, d_{i_2} < 0$. This means that a case where $\vec{x}$ is found to be in class ω_{i_1} and not in class ω_{i_2} is weighted higher than a case where the likelihood for being in class ω_{i_1} is just higher than the likelihood for being in class ω_{i_2}. Function $A(.)$ shows how the weight is computed which encodes the reliability of a classification $\vec{d}$:

$$\begin{aligned} A\left(\vec{d}\right) &= l\left(d_{i_1}\right) - l\left(d_{i_2}\right), \\ d_{i_1} &= \max_i d_i, \quad d_{i_2} = \max_{i \setminus i_1} d_i. \end{aligned} \tag{3.27}$$

THE VOTING SCHEME V^S. Making a winner-takes-all class decision based on one frame might be vulnerable to noise. Therefore, an alternative approach is the sum rule which skips the decision on the frame-level. Instead, the responses of each model g_i are collect over time resulting in some kind of accumulated distance vector $\vec{d}^S$ on which the decision function $E(.)$ is applied. In this case, a classification is performed on a bigger amount of data compared to the classification on frame-level which hopefully leads to a more stable result. As outlined above the output distances of the discriminant functions have to be normalized. This can be realized by extending the logistic function of Equation 3.26 to vectors by applying $l(.)$ to each entry of the input vector:

$$L(\vec{d}) = \begin{pmatrix} l(d_1) \\ \vdots \\ l(d_6) \end{pmatrix} \tag{3.28}$$

The final decision vector $\vec{e}^S$ is defined as follows:

$$\begin{aligned} \vec{e}^S &= E\Big(\vec{d}^S\Big), \\ \vec{d}^S &= \sum_{i=0}^{\Delta t-1} L(\vec{d}_{t-i}), \quad \vec{d}_{t-i} = G(\vec{x}_{t-i}). \end{aligned} \tag{3.29}$$

COMBINING DIFFERENT FEATURE TYPES. As described above, the fusion of feature vectors over time is realized by summing up weighted decision vectors or normalized distance vectors to a voting vector on which the final classification decision is taken. This voting technique can be easily extended to fuse responses of different features. First, a set of discriminant functions per feature type has to be learned. Second, responses over time are combined to different voting vectors, one per feature type. Third, the fusion of feature types is then realized by simply summing these voting vectors to one final voting vector on which the classification function $E(.)$ is applied to get a final class decision. The following formulas give details on the fusion of 3D features and Gist features $\{ (\vec{x}^{3D}_{t-i}, \vec{x}^{Gist}_{t-i}) \}_{i=0}^{\Delta t}$ extracted from a sequence $\{ \mathcal{F}_{t-i} \}_{i=0}^{\Delta t}$. The discriminant functions learned from 3D features are denoted by $G^{3D}(.)$ and those learned from Gist features by $G^{Gist}(.)$. Using voting scheme V^M ($\rightarrow$ Equation 3.25) the decision vector $\vec{e}^M_C$ is computed by:

$$\begin{aligned} \vec{e}^M_C &= E\Big(\vec{d}^M_{3D} + \vec{d}^M_{Gist}\Big), \\ \vec{d}^M_{3D} &= \sum_{i=0}^{\Delta t-1} A(\vec{d}^{3D}_{t-i}) E(\vec{d}^{3D}_{t-i}), \quad \vec{d}^{3D}_{t-i} = G^{3D}(\vec{x}^{3D}_{t-i}), \\ \vec{d}^M_{Gist} &= \sum_{i=0}^{\Delta t-1} A(\vec{d}^{Gist}_{t-i}) E(\vec{d}^{Gist}_{t-i}), \quad \vec{d}^{Gist}_{t-i} = G^{Gist}(\vec{x}^{Gist}_{t-i}). \end{aligned} \tag{3.30}$$

For voting scheme V^S (Equation 3.29) the final classification $\vec{e}_C^S$ results from:

$$\begin{aligned} \vec{e}_C^S &= E\Big(\vec{d}_{3D}^S + \vec{d}_{Gist}^S\Big), \\ \vec{d}_{3D}^S &= \sum_{i=0}^{\Delta t-1} L\big(\vec{d}_{t-i}^{3D}\big), \quad \vec{d}_{t-i}^{3D} = G^{3D}\big(\vec{x}_{t-i}^{3D}\big), \\ \vec{d}_{Gist}^S &= \sum_{i=0}^{\Delta t-1} L\big(\vec{d}_{t-i}^{Gist}\big), \quad \vec{d}_{t-i}^{Gist} = G^{Gist}\big(\vec{x}_{t-i}^{Gist}\big). \end{aligned} \tag{3.31}$$

REJECTION. Depending on the selected window size, the speed of the camera drive, and the current frame rate of the camera the actually acquired frames might only show uninformative scene views or may be disturbed by persons moving in front of the camera. It is clear that classifications results on frames acquired in such situations will barely be correct. As in some human-robot scenarios a robust scene classification will be requested the robot should rather reject some classification results than provide uncertain class labels. Rejection can be introduces by modifying the decision function $E(.)$ of Equation 3.22 in the following way

$$\begin{aligned} E_{rej}\big(\vec{d}\big) &= \begin{pmatrix} e_1 \\ \vdots \\ e_6 \end{pmatrix}, \quad \text{where } e_i = \begin{cases} 1 & : i = i_1 \wedge \frac{d_{i_1} - d_{i_2}}{d_{i_1}} > \theta_{rej} \\ 0 & : \text{else.} \end{cases} \\ i_1 &= \arg\max_i d_i, \quad i_2 = \arg\max_{i \setminus i_1} d_i \end{aligned} \tag{3.32}$$

It means that the maximum value determining the resulting class must significantly differ from the second best value. Otherwise, the classification cannot be conducted reliably enough and is therefore rejected. During evaluation $\theta_{rej} = 0.05$ has turned out to be best suited as not more than 20% of the test frames have been rejected. $E_{rej}(.)$ replaces $E(.)$ only at the final decision stage where the summed voting vector $\vec{d}^*$ is mapped on the final decision vector

$$\begin{aligned} \vec{e}_{rej} &= E_{rej}\big(\vec{d}^*\big), \\ \vec{d}^* &= \vec{d}^S \quad \text{or} \quad \vec{d}^* = \vec{d}^M. \end{aligned} \tag{3.33}$$

3.4 EVALUATION

This section is going to evaluate the 3D features and voting schemes proposed in Section 3.3. This is done on the basis of a newly recorded 3D indoor database introduced in Section 3.4.1 and on test sequences acquired in real homes. Section 3.4.2 presents the specific classifiers and training strategy used to learn room models from the database. The following evaluation focuses on different aspects of the 3D indoor categorization problem. Section 3.4.3 analyzes the performance of both voting schemes V^M and V^S for integrating classification responses over time and different feature types proposed in Section 3.3.3. Additionally, the influence of rejecting unstable classification results on the error rate is examined. In Section 3.4.4 the best combination of voting scheme and rejection is picked for a detailed analysis of the appearing inter-class confusions using different features. Section 3.4.5 deals with the question how to combine features best. It is possible to either concatenate feature vectors or to fuse classifier outputs. Section 3.4.6 investigates the room label distribution and frame rejection along selected test sequences. And finally, the correlations between sub-vectors of the 3D feature vector are investigated in Section 3.4.7.

3.4.1 *The 3D Indoor Database*

For studying the classification performance of my 3D features, a suitable database is required containing snapshots of diverse indoor rooms from a robot perspective consisting of dense 3D point clouds. In the area of scene classification based on 2D images some databases are available on the web [20] [21] [22] [23]. These databases mainly contain pictures of natural outdoor scenes like "forest", "mountain", "coast" and pictures of man-made outdoor scenes like "building", "highway", "suburb". Sometimes the databases also include man-made indoor scenes like "store", "living room", "kitchen". An interesting benchmark for visual indoor place recognition from a robot's point of view is provided by Pronobis and colleagues [24]. The COLD database contains 2D images of a one-person office, a two-person office, a kitchen, a corridor and a printer area acquired at three different laboratory environments under various weather and illumination conditions.

Even though, the rough 3D spatial layout can be estimated from a single 2D image – e. g.via estimating the surface orientation [Sax08, Hoi07], the 3D room frame from line segments [Lee09, Hed09], or depth-ordered planes [Yu08] – the resulting 3D layout is not detailed enough for capturing all spatial structures given by scene-typical furniture. Further, the existing databases fail to capture indoor rooms from a robot perspective. Therefore, I have recorded an own indoor

20 `http://web.mit.edu/torralba/www/indoor.html` [Qua09]
21 `http://people.csail.mit.edu/torralba/code/spatialenvelope` [Oli01]
22 `http://www.emt.tugraz.at/~pinz/data/tinygraz03` [Wen07]
23 `http://vision.stanford.edu/Datasets/SceneClass13.rar` [FF05]
24 `http://cogvis.nada.kth.se/COLD` [Ull08]

Figure 3.16: This figure shows photos of 28 different rooms that have been scanned with the SwissRanger SR3100 in an IKEA home-center. The images have been taken from the view the 3D camera. The database contains 3 bathrooms, 4 bedrooms, 6 eating places, 7 kitchens, 5 living rooms, and 3 offices.

database [25] capturing 3D point clouds from a sufficient number of different rooms. As outlined in Chapter 2 3D ToF sensors are best suited for capturing quickly dense 3D point clouds from indoor rooms, especially, from homogeneous furniture areas. For acquiring many different arranged rooms per category we have taken the SwissRanger SR3100 camera to a regular IKEA home-center [26]. The exhibition is ideally organized for our purpose, because it is assembled by opened 3D boxes showing example rooms. Therefore, it is possible to acquire per room category a proper number of differently arranged rooms in a short time. To simulate a robot-like view on the scene the 3D camera is placed at an arbitrary position of the open box side at a height of 140cm scanning the room for 20 to 30 seconds. Round about 300 to 400 frames are acquired while continuously moving the camera by ca. 40° left/right and ca. 10° up/down. This simulates a robot moving its head around for perceiving the entire scene. We have acquired data from 3 bathrooms, 4 bedrooms, 6 eating places, 7 kitchens, 5 living rooms, and 3 offices. Figure 3.16 shows digital photos of the scanned rooms taken at the positions of the 3D camera. As IKEA has stores all over the world a database on IKEA data can be easily extended and holds furniture arrangements available in real rooms all over the world.

3.4.2 *Classifier Selection and Training*

The 3D feature plot in Figure 3.12 suggests that the room classification using these features can be solved by using linear classifiers. Support Vector Machines are such widely used classifiers [Vap95]. Also, in my exploration phase (→ Section A.1), SVMs have turned out to work well for the indoor classification problem. Therefore, I have selected the SVM approach to learn room models on features extracted from SwissRanger frames. I utilize the SVM^{light} library [27] [Joa99]. It comes with four built-in kernels which are the linear kernel, the polynomial kernel, a kernel based on the radial basis function, and a kernel based on the sigmoid function. Due to impressions gained in an empirical testing, the Radial Basis Function (RBF) is used as SVM kernel here:

$$K(\vec{x},\vec{y}) = e^{-\gamma\cdot\|\vec{x}-\vec{y}\|_2}. \tag{3.34}$$

The parameter γ of the RBF kernel and the regularization parameter c (trade-off between training error and margin) are optimized in a 10-fold-cross-validation scheme choosing a model with a small number of support vectors while reducing the classification error. Here, $c = 900$ and $\gamma = 2$ turned out to produce models which perform well. These parameter values are fixed over the following test runs and feature sets in order to get comparable results.

25 It can be downloaded from this website `http://www.techfak.uni-bielefeld.de/~aswadzba/3D-IKEA-database.tar.gz`

26 `http://www.ikea.com/us/en`

27 `http://svmlight.joachims.org`

The nature of the SVM approach is that it has been designed to solve a two-class problem. This means that one hyperplane is estimated through the Support Vectors defining the boundary between these two classes. One procedure to transfer the SVM approach to a multiple-class classification problem is to train for each class (here, room type) a SVM model in an one-vs-all way. All features belonging to one room type are the positive samples for the model. The negative examples are uniformly sampled from features of the remaining room classes. Per model the amount of positive and negative samples is kept equally. Finally, for m classes m models will be trained.

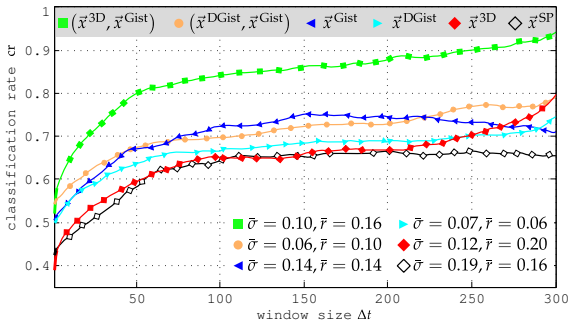
An analysis of the classification performance of features requires a separated training and test set. Here, the training and test sets are generated by choosing randomly from the 3D database one room per room type as test sequence. The remaining rooms of one type form the training set for this room class. This selection is repeated 10 times. Averaging the classification rates over these 10 runs should ensure comparability of the classification rates of different features and voting schemes as a bias arising from differences in the performance of individual test sequences is averaged out.

3.4.3 *Classification Performance for Different Window Sizes*

This section contrasts the performance of the features $\vec{x}^{\,3D}$, $\vec{x}^{\,Gist}$, $(\vec{x}^{\,3D}, \vec{x}^{\,Gist})$, $(\vec{x}^{\,DGist}, \vec{x}^{\,Gist})$, and $\vec{x}^{\,SP}$. The calculation of $\vec{x}^{\,3D}$ is given in Section 3.3.1 and of $\vec{x}^{\,Gist}$ in Section 3.3.2. Both features are fused to $(\vec{x}^{\,3D}, \vec{x}^{\,Gist})$ using the voting schemes presented in Section 3.3.3. $\vec{x}^{\,DGist}$ is a so-called *Depth-Gist* feature vector. It is computed in the same way as $\vec{x}^{\,Gist}$. But for the Gist computation the SwissRanger depth image instead of the SwissRanger amplitude image is used. $\vec{x}^{\,SP}$ is computed according to Lazebnik's approach presented in Section 3.2.3. The curves in Figure 3.17 show the classification rates where the window size Δt is varied from 1 to 300 frames. A curve is plotted per feature type. Given models trained with a certain feature type, the classification rate for a certain window size Δt is computed by counting the correct labels over the different test classes and averaging this overall performance over 10 runs.

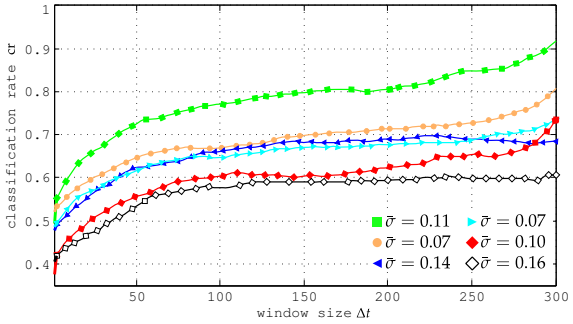
The adjustable parameter in the voting schemes presented in Section 3.3.3 is Δt the size of the window $[\,t - \Delta t,\, t\,]$ over which classification results are fused. Δt is expressed by the number of consecutive frames necessary for determining the correct class label. As our sequences are recorded with a camera roughly standing at one or two positions and being moved continuously simulating a robot's head looking around, the possible values could range from one frame to all frames of the sequence. Performing room category decision on one frame is expected to be very fast but vulnerable to noise. The more frames are integrated the more stable the classification result should become and the longer the robot has to collect data before a decision can done. There is a trade-off between getting a room type label quickly and reliably. The curves in Figure 3.17 show the development of the classification rates if the window size Δt is enlarged from 1 frame to 300 frames. It can be observed over all tested features that

(a) voting scheme: V^S; rejection: $\theta_{rej} = 0.05$



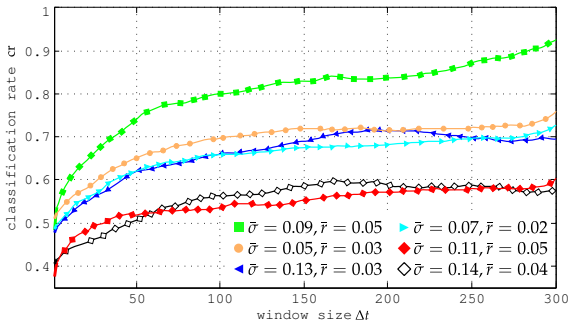
cr	$\Delta t = 1$	$\Delta t = 50$	$\Delta t = 300$	∅
■	0.52	0.80	0.94	0.84
●	0.54	0.67	0.79	0.71
◀	o.50	0.67	0.71	0.71
▶	0.50	0.64	0.75	0.67
◆	0.39	0.59	0.80	0.65
◇	0.42	0.58	0.66	0.63

(b) voting scheme: V^S; rejection: no



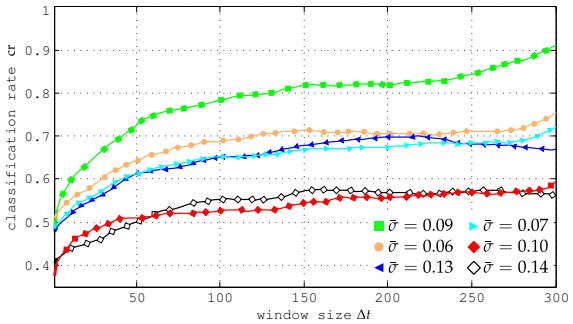
cr	$\Delta t = 1$	$\Delta t = 50$	$\Delta t = 300$	∅
■	0.49	0.72	0.92	0.78
●	0.52	0.65	0.81	0.69
◀	0.48	0.62	0.68	0.66
▶	0.49	0.62	0.74	0.65
◆	0.38	0.56	0.74	0.60
◇	0.41	0.53	0.61	0.57

(c) voting scheme: V^M; rejection: $\theta_{rej} = 0.05$



cr	$\Delta t = 1$	$\Delta t = 50$	$\Delta t = 300$	∅
■	0.48	0.74	0.93	0.80
●	0.51	0.65	0.76	0.69
◀	0.48	0.62	0.69	0.66
▶	0.49	0.62	0.73	0.66
◆	0.38	0.52	0.60	0.55
◇	0.41	0.51	0.58	0.56

(d) voting scheme: V^M; rejection: no



cr	$\Delta t = 1$	$\Delta t = 50$	$\Delta t = 300$	∅
■	0.48	0.73	0.91	0.79
●	0.51	0.64	0.75	0.68
◀	0.48	0.61	0.67	0.65
▶	0.49	0.61	0.72	0.65
◆	0.38	0.51	0.59	0.54
◇	0.41	0.50	0.56	0.54

Figure 3.17: This figure shows the development of the classification results while enlarging the window Δt from 1 to 300 frames ($\rightarrow$ x-axis). The voting schemes V^M and V^S are evaluated as introduced in Equation 3.25 and 3.29. Additionally, rejection using $\theta_{rej} = 0.05$ is contrasted with no rejection. The tested feature types and combinations are: ■ $(\vec{x}^{3D}, \vec{x}^{Gist})$, ● $(\vec{x}^{DGist}, \vec{x}^{Gist})$, ◀ $\vec{x}^{Gist}$, ▶ $\vec{x}^{DGist}$, ◆ $\vec{x}^{3D}$, and ◇ $\vec{x}^{SP}$. $\bar{\sigma}$ denotes the mean standard deviation of the corresponding classification curve. $\bar{r}$ denotes the mean rejection rate. $cr_{\Delta t=1}$, $cr_{\Delta t=50}$, and $cr_{\Delta t=300}$ in the tables are the classification rates at the corresponding window sizes while $cr_{\varnothing}$ is the classification rate averaged over the different window sizes. (best viewed in color)

the classification rates increase if the Δt becomes larger. The steepest ascent is located at the first part of the curves for $\Delta t = 1, \ldots, 50$ followed by a smoother increasing over the rest of the curve. The gradient $\frac{\partial \, \mathrm{cr}(\Delta t)}{\partial \, \Delta t}$ of a curve $\mathrm{cr}(\Delta t)$ can be computed by dividing the difference of two classification rates in percent by the difference of the corresponding window sizes. For example, the green curve in Figure 3.17(a) has between the classification rates of $\mathrm{cr}_{\Delta t=1} = 0.52$ and $\mathrm{cr}_{\Delta t=50} = 0.80$ a much higher gradient (here, 0.56) than between $\mathrm{cr}_{\Delta t=50}$ and $\mathrm{cr}_{\Delta t=300} = 0.94$ (which is 0.056).

Taking a deeper look on the curves in Figure 3.17 and the classification rates in the tables it can be seen that our proposed combination $(\vec{x}^{\,\mathrm{3D}}, \vec{x}^{\,\mathrm{Gist}})$ of spatial and gist features (green curve) clearly outperforms the other features under all voting and rejection conditions. The classification rates (e. g., $\mathrm{cr}_{\varnothing} = \mathbf{0.84}$, $\mathrm{cr}_{\Delta t=300} = \mathbf{0.94}$) are significantly higher than the corresponding rates of the second best feature combination $(\vec{x}^{\,\mathrm{DGist}}, \vec{x}^{\,\mathrm{Gist}})$ (orange curve, e. g., $\mathrm{cr}_{\varnothing} = \mathbf{0.71}$, $\mathrm{cr}_{\Delta t=300} = \mathbf{0.79}$). The corresponding error reduction of, e. g., **45**% and **71**% is quite impressive. In general, the error reduction lies between 29% and 71%. Contrasting the two curves reveals insights in the nature of information that is encoded by the feature vectors $\vec{x}^{\,\mathrm{3D}}$ and $\vec{x}^{\,\mathrm{DGist}}$, respectively. Though the performance of both features used alone is comparable only $\vec{x}^{\,\mathrm{3D}}$ encodes information of room types that is complementary to the information encoded by $\vec{x}^{\,\mathrm{Gist}}$. Consequently, using $\vec{x}^{\,\mathrm{3D}}$ in combination with $\vec{x}^{\,\mathrm{Gist}}$ boosts the room type categorization performance. This points out that a 25-dimensional feature vector $\vec{x}^{\,\mathrm{3D}}$ carefully defined on 3D data captures spatial structures in a sufficiently generalized way compared to a 512-dimensional Depth-Gist feature vector $\vec{x}^{\,\mathrm{DGist}}$. Of further interest, is a comparison between $\vec{x}^{\,\mathrm{3D}}$ (red curves) and the 8500-dimensional feature vector $\vec{x}^{\,\mathrm{SP}}$ (black curves). The curves of both features show a similar performance which is positively remarkable for the $\vec{x}^{\,\mathrm{3D}}$ as it shows that meaningful information is encoded in order to be able to recognize the room type of a newly presented room at an acceptable performance level ($\mathrm{cr}_{\varnothing} = 0.65$). But it is also visible that the codebook based approach producing 8500-dimensional feature vectors $\vec{x}^{\,\mathrm{SP}}$ is far too costly for not improving classification. Especially, the large mean standard deviation of about $\bar{\sigma} = 0.14$ to $\bar{\sigma} = 0.19$ is due to the fact that some room types are well recognized (like bathroom and kitchen) while others are not. As computation and training of such big features is quite time consuming the application of these features on a mobile platform with limited computational power is currently not realistic and not necessary since as shown other well performing features are available.

Figure 3.17 contrasts the different voting and rejection schemes. Figure 3.17(b) shows the results acquired by integrating over time using Equation 3.29 and 3.31 for combining features of different types. In Figure 3.17(a) also rejection is considered by replacing the final decision function $E(.)$ with $E_{\mathrm{rej}}(.)$, introduced in Equation 3.32. Figure 3.17(c) and 3.17(d) show results using the voting scheme V^{M} (Equation 3.25 and 3.30). The results in Figure 3.17(c) are achieved by rejecting in the final classification step unclear classification decisions. Considering all feature curves the rejection of undecidable frames influences mostly voting scheme V^{S}. The mean improvement is about 0.04 while V^{M} is only improved

by 0.01. That is because taking decisions on the frame level leads to clearer decisions on the window-level which leads to less influence of the rejection. It comes with the drawback of being more vulnerable to noise. If rejection is enabled the rate of dismissed frames range from $\bar{r} = 3\%$ to $\bar{r} = 20\%$ leaving a fair amount of classifiable frames. Skipping the decision on the frame-level (sum rule) improves classification results by round about 0.03. Especially, the classification based on the 3D spatial features $\vec{x}^{\,3D}$ shows a considerable improvement by 0.10 and 0.06 if the voting scheme V^S is used. 3D features profit at most from keeping the normalized support for all classes when fusing over several consecutive frames. The mean difference between results based on V^M without rejection (see Figure 3.17(d)) and V^S (Figure 3.17(a)) with rejection is 0.06. The following evaluations are based on results achieved by using V^S as voting scheme and rejection with $\theta_{\text{rej}} = 0.05$.

3.4.4 *Classification Performance per Class*

This section examines the classification performance for each class by analyzing confusion tables. As I aim for a realistic robotic scenario, integration times of up to 60 frames are of interest as there will be an initial delay of 2 to 6 seconds before the robot would deliver class labels when continuously scanning a room. Therefore, classification results achieved by a window size of $\Delta t = 60$ are presented in Figure 3.18 for a detailed analysis.

Contrasting Figure 3.18(a) and Figure 3.18(c) with Figure 3.18(d) and Figure 3.18(b) it can be seen that the Gist features and the 3D spatial features provide complementary information because the Gist features perform well for the bathroom, the bedroom, the eating place and the kitchen while the 3D features work for the bathroom, the kitchen, the living room, and the office. So, combining both feature types covers all room types contained in the 3D database. The high classification rate of **0.97** of $\vec{x}^{\,3D}$ for the "living room"-class has to be emphasized especially in contrast to the low rate of **0.27** for $\vec{x}^{\,\text{Gist}}$. The 2D and 3D projection of both features (see Figure 3.12 and 3.14) visualize reasons for this behavior. Features of living rooms form a compact and well separable cluster in the feature space of the 3D spatial features which is not the case in the Gist feature space. The projections give also an explanation for the mix-up between "living room", "kitchen", and "bedroom" in the Gist space as the clusters for these room types are close to or even intersect each other. While "office"-percepts are often classified as "eating place", "kitchen", or "living room" and seldom as "bathroom" or "bedroom" as these two clusters are quite distant and separated from the other clusters. Figure 3.12 shows a proximity of features from the "office"-class and the "eating place"-class. This explains why "office"-percepts are often mistaken for "eating place"-percepts and vice versa in the confusion matrix. This proximity may arise for views on an eating place where only one chair is visible together with a table whereas normally eating places consist of a table and at least two chairs positioned around the table. Lazebnik's 8500-dimensional reference features (see Figure 3.18(f)) perform well

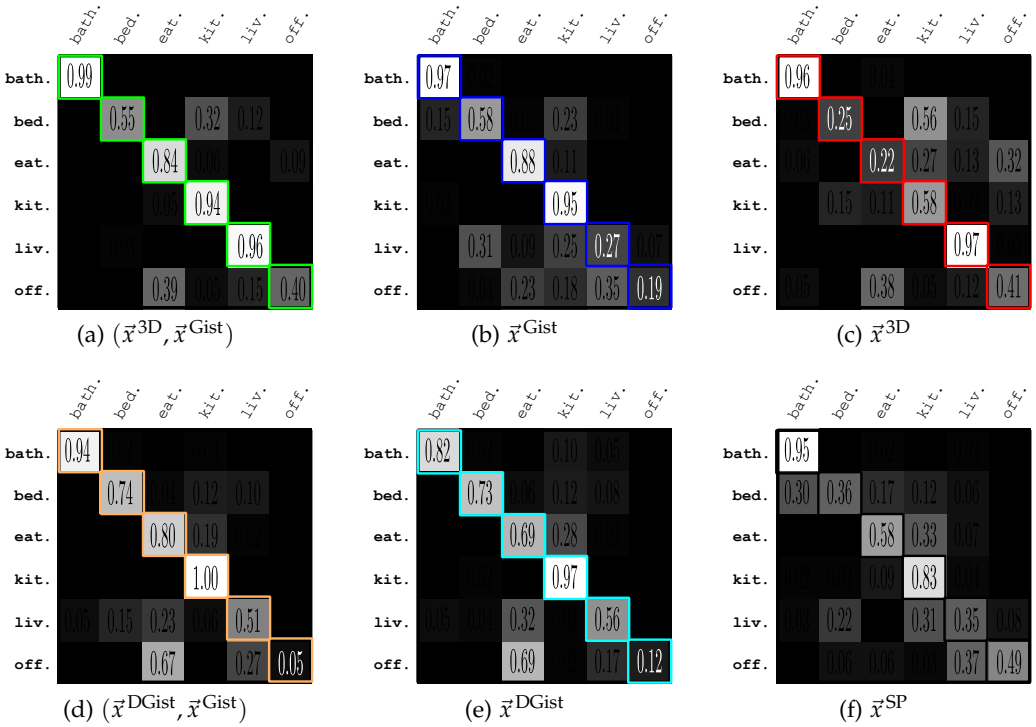


Figure 3.18: This figure shows the confusion matrices of all examined features and feature combinations: (a) $(\vec{x}^{3D}, \vec{x}^{Gist})$, (b) $\vec{x}^{Gist}$, (c) $\vec{x}^{3D}$, (d) $(\vec{x}^{DGist}, \vec{x}^{Gist})$, (e) $\vec{x}^{DGist}$, (f) $\vec{x}^{SP}$. Here, the window size over which consecutive frames are integrated using the V^S scheme is set to $\Delta t = 60$. The classification rates per room type are listed along the diagonal, all other entries show misclassifications. The **ground truth** labels are highlighted through bold letters.

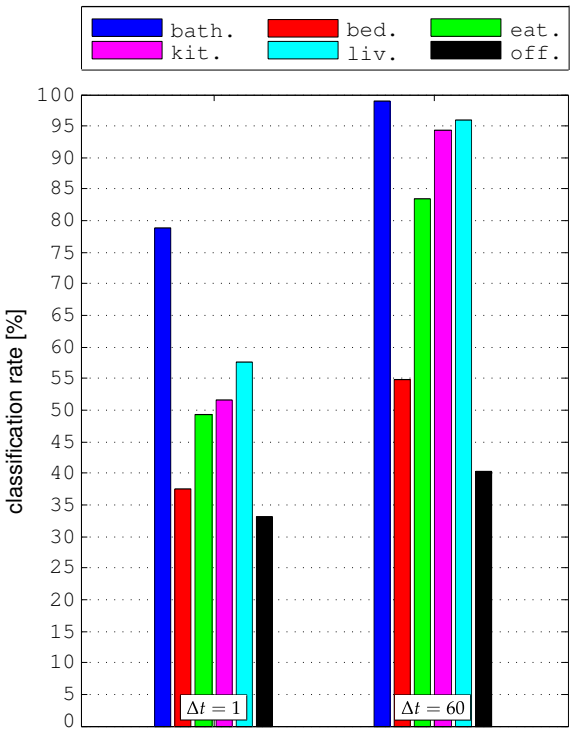


Figure 3.19: The bars in the diagram show the classification rates of the room types "bathroom", "bedroom", "eating place", "kitchen", "living room", and "office" from the 3D IKEA database. $\{ (\vec{x}^{3D}_{t-i}, \vec{x}^{Gist}_{t-i}) \}_{i=1,\ldots,\Delta t-1}$ has been used as features integrated over a window of size $\Delta t = 1$ and $\Delta t = 60$. Per room type a room is chosen randomly as test sequence while the remaining rooms form the training sequences. The selection is repeated 10 times. The classification rates are averaged over this runs.

for the "bathroom" and the "kitchen", acceptably for the "eating place" and the "office", and badly for "bedroom" and "living room".

Ullah and colleagues [Ullo8] have presented in their paper an an indoor categorization approach which is trained with 2D images of the COLD database collected in a robot-like fashion in different rooms of three universities. They assessed their database using a purely appearance-based method. Local 2D features were extracted from the training images using Harris-Laplace detectors and SIFT descriptors. For classification SVM models are trained. Even though the COLD database and our 3D IKEA database contain different room types both have been recorded from the perspective of robots. The COLD database consists of 2D images and the IKEA database of 3D point clouds. Ullah and me extract features optimized for the data, train SVM models for room types contained in the utilized database, and classify single images or frames. Comparing Figure 3.8 with Figure 3.19 showing the per-class classification rates it can be seen, that Ullah has only been able to train a good model for the "corridor" (`CR`) class (`CR`: 0.76, `PA`: 0.12, `2PO`: 0.13, `BR`: 0.10). We are able to provide several good holistic models, one for the "bath room", the "eating place", the "kitchen", and the "living room" (with `bath.`: 0.79, `bed.`: 0.38, `eat.`: 0.49, `kit.`: 0.52, `liv.`: 0.58, `off.`: 0.33 for $\Delta t = 1$ and even better if the window size Δt is enlarged).

3.4.5 *Feature Concatenation vs. Classifier Fusion*

Features of different type can be either combined by concatenating them to one vector or by fusing classifier outputs using a certain rule. Here, feature combination is necessary two times. First, $\vec{x}^{\triangleleft}$, $\vec{x}^{\div}$, $\vec{x}^{\mathrm{S}}$, and $\vec{x}^{\mathrm{A}}$ have to be combined. And second, $\vec{x}^{\mathrm{3D}}$ and $\vec{x}^{\mathrm{Gist}}$ have to be fused. Figure 3.20 compares the classification performances of the two different combination strategies, feature concatenation and classifier fusion. Voting scheme V^{S} presented in Equation 3.31 is used to fuse the different features. If features are concatenated to one vector, SVM models are trained with the same parameters used for the training of models for the individual features. The features capturing patch properties are combined best by concatenation as proposed in Section 3.3.1. A gain of 10% in average is achieved against classifier fusion of the spatial features. $\vec{x}^{\mathrm{3D}}$ and $\vec{x}^{\mathrm{Gist}}$ are slightly better combined by fusing their classifier outputs. The performance gain compared to concatenation is 5% at average.

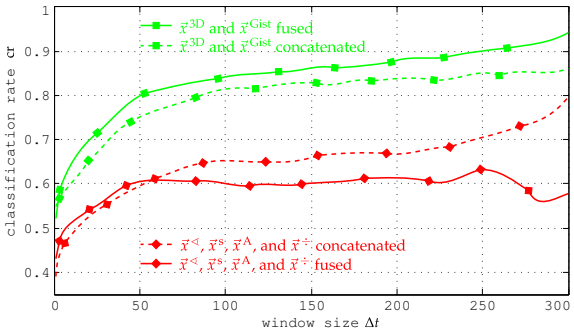


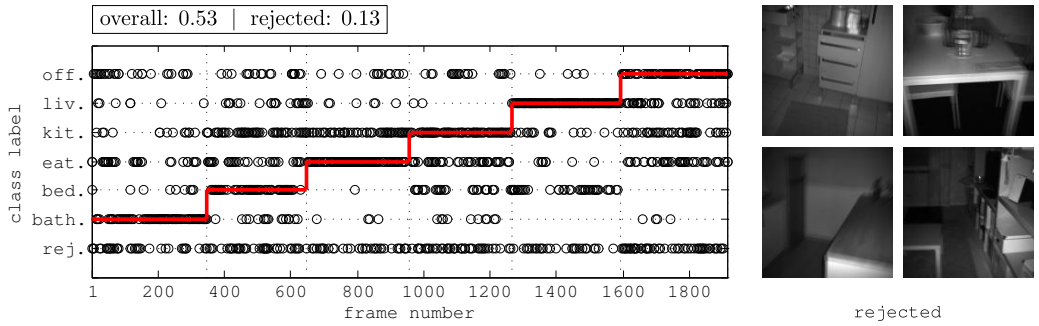
Figure 3.20: This figure shows the performance curves of different feature combinations. Different feature types are combined by either concatenating them to one vector or by fusing classifier outputs. Voting scheme V^{S} given in Equation 3.31 is used to fuse the different features.

3.4.6 *Room Label Distribution along Example Sequences*

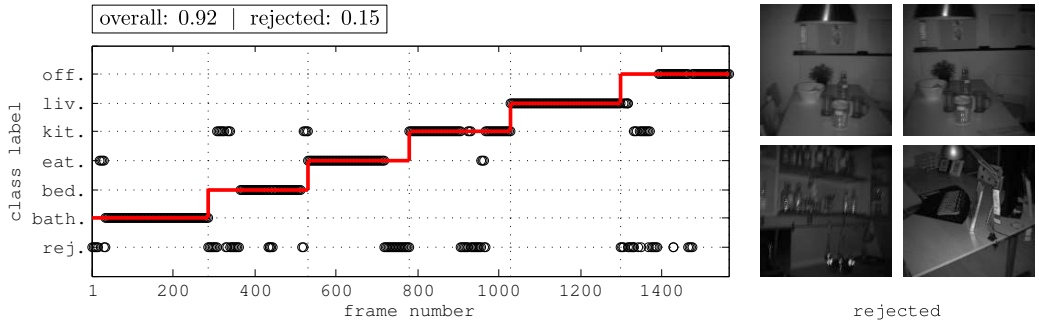
As I aim for a reliable classification of a continuously acquired sequence of a new room similar to the "looking around" or "eye saccade" paradigm, this section is going to analyze the label distribution along several test sequences. The test sequences come from the IKEA database and are recorded in two real apartments, my own one and a visitor apartment of the Applied Informatics Group at Bielefeld University. The purpose of this section is to give an impression of the label distribution utilizing different window sizes. Additionally, the performance of the holistic scene model learned from the IKEA database is tested if applied on data of real flats. 3D spatial features and Gist features are used together combined by voting scheme $\dot{\mathrm{V}}$ with the rejection parameter θ_{rej} set to 0.05.

First, a test run is analyzed where `bath.3`, `bed.3`, `eat.5`, `kit.1`, `liv.3`, and `off.3` have been chosen from the IKEA database as test sequences while the remaining sequences have been used for training the SVM models. Figure 3.21 gives an overview of the label distributions assigned in the classification process using different window sizes for voting. The sequences are concatenated in the figure but they are analyzed individually. The continuous red line denotes the ground truth labeling and the black circles mark the classification results. Figure 3.21(a) shows the label distribution when classifying each frame individually. As expected the distribution is quite noisy and only a fraction (**0.53**) of the frames meet the ground truth. In Figure 3.21(b) for each frame the preceding 59 frames contribute to the classification by voting. It can be seen that classification is much more stable and meets the ground truth quite well. Only few frames are misclassified, the majority of undecidable frames are rejected. Last, in Figure 3.21(c) a dynamic window expanding from 1 frame to the whole sequence is used for voting. For each frame all foregoing frames influence the classification. Especially, at the beginning of a sequence the classification could be unstable resulting in some rejections which is due to the fact that only a small part of the room is already known and only a small number of frames can be considered for room type categorization.

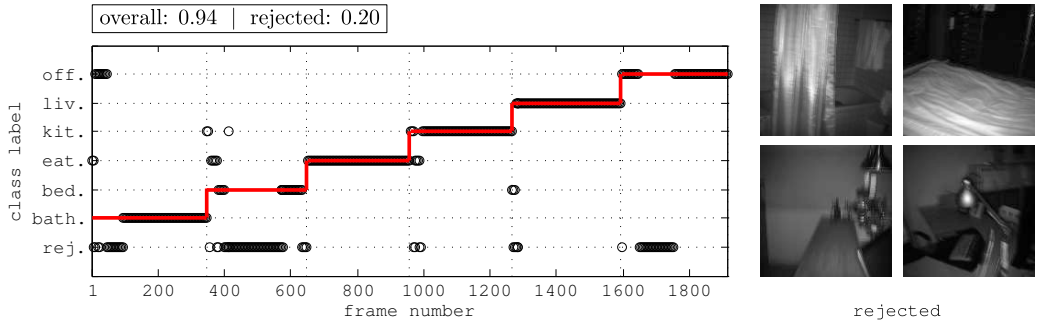
The right column of Figure 3.21 presents examples of rejected frames. Among them are a lot of close-up views on the current scene like views on diverse tables for which the spatial layout of the scene cannot be estimated and thus not decided to which class this view belongs. Other frames are rejected due to their noise in the raw data or because they belong to a sequence which contains views on furniture that can be found across different room types like, e. g., shelves or sideboards. Further, rejected frames cluster in time. This means that if consecutive frames are combined together for classification often sets of consecutive frames are rejected. This is due to the fact that the camera is moved continuously while recording so that a set of consecutive frames will contain close-up views or uninformative views on furniture.



(a) voting window size: $\Delta t = 1$



(b) voting window size: $\Delta t = 60$



(c) dynamic voting window

Figure 3.21: This figure shows the classification results of six test sequences (`bath.3`, `bed.3`, `eat.5`, `kit.1`, `liv.3`, and `off.3`). The red line refers to the ground truth and the black circles mark the classification results. The shown results are achieved by using 3D features and Gist features in combination and voting over a frame window (a) $\Delta t = 1$, (b) $\Delta t = 60$, and (c) a dynamic window which means that for all frames of a sequence all previous frames are considered during voting. The right column gives examples of rejected frames.

This paragraph is going to examine the generalizing abilities of the room models trained on the IKEA database. These models are applied to sequences acquired in real flats. Figure 3.22(a) and 3.22(b) show pictures of the rooms recorded in flat `F1` and flat `F2`. These pictures are roughly taken from the position where the SwissRanger camera recorded 300 frames per room. These sequences form unknown rooms that are tested against SVM models trained by utilizing the complete IKEA database. Here, also $\vec{x}_{t-i}^{\,3D}$ and $\vec{x}_{t-i}^{\,Gist}$ are integrated over a window $\Delta t = 60$ using V^S and a rejection with θ_{rej} set to 0.05. Figure 3.22(c) and 3.22(d) present the classification results for each sequence. The "kitchen" sequences,

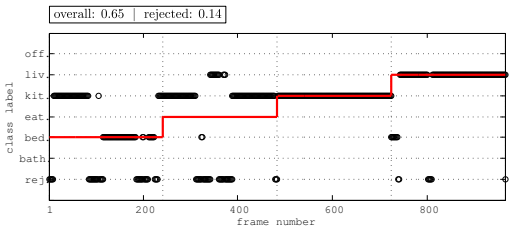
kit.F1 and kit.F2, the "living room" sequences, liv.F1 and liv.F2, and the "eating place" sequence, eat.F2, are most of the time correctly recognized. Only some frames in the middle of kit.F2 are rejected or wrong classified. This is because the camera has been directed towards the kitchen window. As the sun light contains infrared light the SwissRanger measurement principle gets confused (→ Section 2.3.1). This effect is also responsible for the misclassification of the complete eat.F1 sequence. Suppressing this effect is a subject to a technical solution since there already exists a suppression of background illumination for ToF sensors [Möl05]. Also, atypical missing or arrangement of furniture leads to misclassifications and rejections as happened for bed.F1 and bed.F2. Here, the bedroom of flat F1 contains only a bed and no further furniture that structures the room. In bed.F2, the bed is placed atypically in a corner of the room so that it is not visible in most views from the room entrance. It could be expected that classification rates will improve here when the room is entered to get a more typical view on the scene.

bed.F1 eat.F1 F1.kit liv.F1

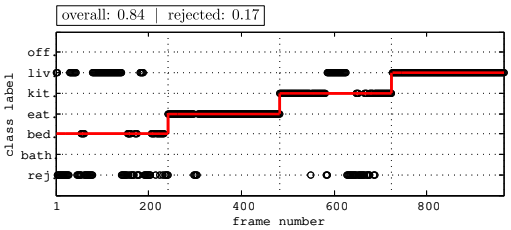
(a) Rooms of flat F1.

bed.F2 eat.F2 kit.F2 liv.F2

(b) Rooms of flat F2.



(c) Label distributions over room sequences of flat F1



(d) Label distributions over room sequences of flat F2

Figure 3.22: Here, photos show the rooms recorded from two flats F1 and F2. Per flat 4 rooms, namely bedroom, eating place, kitchen, and living room, have been recorded. For each room 300 frames are acquired with the SwissRanger camera. In (c) and (d), the red line refers to the ground truth and the black circles mark the classification results.

3.4.7 *Correlations between Sub-Vectors of the 3D Feature Vector*

The 3D feature vector $\vec{x}^{3D}$ consists of four sub-vectors capturing four properties of planar patches assembling a room (→ Section 3.3.1). This section explores the contribution of the sub-vectors on the classification performance of the 3D feature vector.

Identically to the evaluation setup presented in Section 3.4.2, test rooms are sampled 10 times randomly from the 3D IKEA database. Data of the remaining rooms is used to train Support Vector Machines (SVM) models based on the sub-vectors $\vec{x}^{\triangleleft}$, $\vec{x}^{\div}$, $\vec{x}^{A}$, and $\vec{x}^{s}$. Figure 3.23 displays for the different sub-vectors the classification performances averaged over the room types. The curves give the classification progress when the fusion window is increased from $\Delta t = 1$ to $\Delta t = 300$. The best performing sub-vector is $\vec{x}^{A}$ followed by $\vec{x}^{\triangleleft}$, $\vec{x}^{\div}$, and $\vec{x}^{s}$. For some window sizes the classification rate of $\vec{x}^{A}$ even exceeds the rate of the composed 3D vector $\vec{x}^{3D}$. However, if $\vec{x}^{A}$ or $\vec{x}^{3D}$ is fused with $\vec{x}^{Gist}$ nearly no difference in the classification power can be observed. The confusion tables in Figure 3.24 mediate contributions of the sub-vectors to the room class recognition problem. More than 50% of percepts captured in "bathrooms", "bedrooms", "living rooms", and "offices" are categorized correctly using $\vec{x}^{A}$. $\vec{x}^{\triangleleft}$ can reliably categorize percepts of "bathrooms" and "living rooms". $\vec{x}^{\div}$ is specialized on percepts of "bathrooms" and "eating places" and $\vec{x}^{s}$ on percepts of 'bathrooms" and "kitchen".

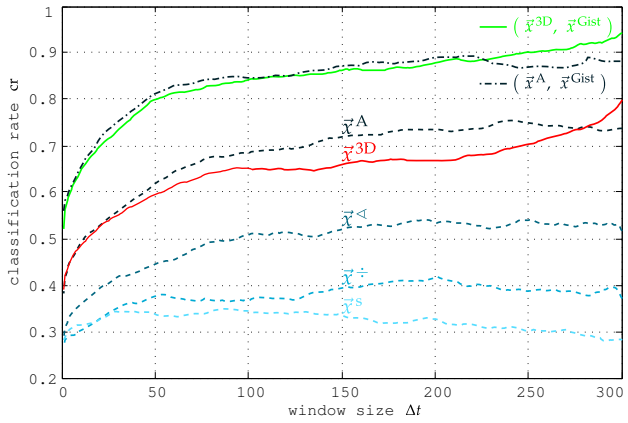
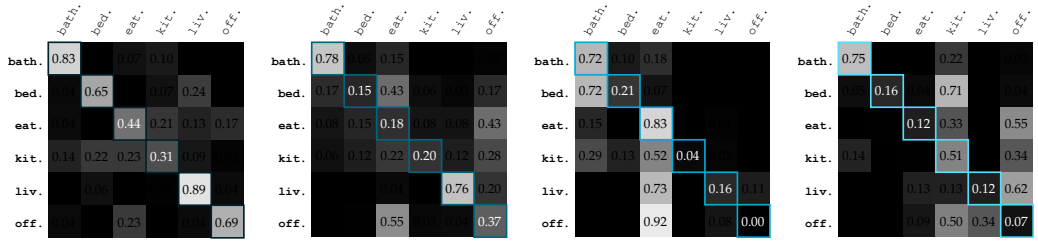


Figure 3.23: The curves show the classification power of the four sub-vectors, $\vec{x}^{A}$, $\vec{x}^{\triangleleft}$, $\vec{x}^{\div}$, and $\vec{x}^{s}$. For comparison reasons, the curve of the concatenated vector $\vec{x}^{3D}$ is displayed.

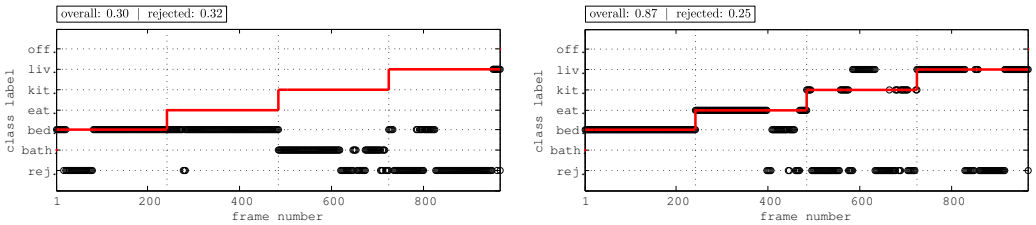


(a) $\vec{x}^{A}$ (b) $\vec{x}^{\triangleleft}$ (c) $\vec{x}^{\div}$ (d) $\vec{x}^{s}$

Figure 3.24: The confusion tables show the per class performances of the sub-vectors, $\vec{x}^{A}$, $\vec{x}^{\triangleleft}$, $\vec{x}^{\div}$, and $\vec{x}^{s}$. The **ground truth** is marked in bold letters.

# SV	models: $g_{\texttt{bath.}}$	$g_{\texttt{bed.}}$	$g_{\texttt{eat.}}$	$g_{\texttt{kit.}}$	$g_{\texttt{liv.}}$	$g_{\texttt{off.}}$	
features:	averaged over 10 models						
$\vec{x}^{3D}$	431	498	575	759	394	553	
$\vec{x}^{A}$	851	659	829	1058	609	752	← increase of ∅ 51%
	of models trained on the entire database						
$\vec{x}^{3D}$	685	712	1004	1165	725	880	
$\vec{x}^{A}$	1190	848	1324	1593	933	1134	← increase of ∅ 37%

Table 3.1: For each room model, $g_{\texttt{bath.}}$, $g_{\texttt{bed.}}$, $g_{\texttt{eat.}}$, $g_{\texttt{kit.}}$, $g_{\texttt{liv.}}$, and $g_{\texttt{off.}}$, the number of support vectors (# SV) is listed. The values are computed either by averaging the support vector counts over the 10 test runs or by counting the support vectors in the models trained on the entire database.



(a) $\left(\vec{x}^{A}, \vec{x}^{Gist}\right)$ applied on flat F1 (b) $\left(\vec{x}^{A}, \vec{x}^{Gist}\right)$ applied on flat F2

Figure 3.25: This figure illustrates the classification performance of the $\vec{x}^{A}$ feature on recordings from real apartments. The red line refers to the ground truth and the black circles mark the classification results. The distributions show the limited generalization performance of $\vec{x}^{A}$.

The evaluation of the sub-vectors on the 3D database gives the impression that the models learned on the basis of the sub-vector $\vec{x}^{A}$ encode the essential information for 3D indoor categorization. However, the question arises whether these models are sufficiently general. If the training parameters are fixed, the number of support vectors per model can serve as indicator for the generalization ability of the models. Table 3.1 lists the number of support vectors in each room model. Counts are either averaged of 10 models per room type or are given for models learned on the entire database. The models based on sub-vector $\vec{x}^{A}$ contain on average 50 or 37 percent more support vectors than the models based on $\vec{x}^{3D}$. It can be assumed, that the higher the number of support vectors the worse the models can be transferred to new rooms. Test sequences of real homes are classified with models based on $\vec{x}^{A}$ fused with models for $\vec{x}^{Gist}$ to test the hypothesis. Figure 3.25 shows the distribution of class labels along the test rooms of two real apartments. Comparing Figure 3.25 with Figure 3.22 a categorization utilizing the sub-vector $\vec{x}^{A}$ produces an exceptionally increased rejection of 0.25 and 0.32. This is probably because classification decisions are not as clear as when $\vec{x}^{3D}$ is used. Since the furnishing of the rooms in apartment F2 have many similarities with rooms in the database, the frames which are not rejected are mostly classified correctly. The total rate is 0.87. But, the models trained on the database cannot be transferred to the rooms of apartment F1. The overall classification rate is 0.30. It is quite small compared to 0.65 which is achieved when $\vec{x}^{3D}$ and $\vec{x}^{Gist}$ are calculated.

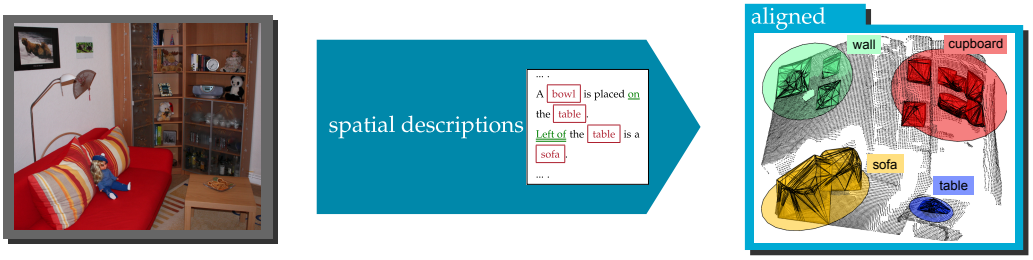
3.5 CONCLUSION AND OUTLOOK

In this chapter, I have presented a holistic scene model providing a rough scene impression of the robot's vista space. It relies on spatial information rather than on object information. 3D spatial features are defined encoding the 3D scene geometry given as a set of planar patches. Shape and size are a standard characteristic of patches, while angle and size ratio between two patches are a novel idea in such kind of particle analysis. I focus on analyzing the relation between patches for extracting information about the spatial layout. Evaluating the defined 25-dimensional 3D feature vector on the 3D IKEA database has shown a remarkable performance. It emphasizes the careful design of this new feature vector. As 3D and Gist feature vector capture complementary aspects of a spatial layout fusing both vectors leads to an error reduction of about 50%. Further, I have developed voting techniques for combining classification responses over different feature types and over time. Best results are achieved by keeping the support for all classes during summing over different responses from 3D and Gist features till a final decision has to be taken. The holistic scene model consisting of SVM models learned on the complete IKEA database have enough generalizing power for providing correct room labels for test sequences acquired in real flats. I have shown that rooms have room type specific arrangements which can be captured by my 3D feature vector.

Future steps will concentrate on defining further patch characteristics similar to those listed by Mozos [Moz05] and others capturing the relation between patches. I have shown that a fusion of classification results from consecutive frames provides a more stable scene categorization results because more information about the room is available than provided by one frame. Instead of classifier fusion, it would be interesting to investigate whether data fusion could be utilized for the indoor categorization problem. Registering several SwissRanger frames leads to a bigger 3D point cloud holding a larger part of the scene. The question is whether it is possible to learn better models from such point clouds enabling a more stable scene classification compared to the models learned from partial scene views. As 3D point cloud registration is a hard and resource consuming problem a significant improvement must be observed in order to recommend its application on a mobile robot. Additionally, the problem of computing the Gist feature vector for registered frames has to be solved as it captures an important part of the spatial layout. Due to these problems, it could be interesting to investigate to which extend the holistic approach would work on class hierarchies. On the top level of such hierarchies could be room types like "kitchen", "living room", etc. and on the lower levels functional subparts of a scene like "a wall with bookshelves" or "sideboard-like-furniture for placing things on it". Such scene subparts could be specific for a top level room type or could appear across different room types. Such a distinction could be used to introduce different weights for different scene subparts encoding their contribution to their parent scene. The weights could tackle the problem of the current classification scheme where views on furniture appearing across different room types (e. g., shelves) corrupt the room labeling procedure.

4

LEARNING ALIGNED SCENE MODELS FROM SPATIAL DESCRIPTIONS



The holistic scene model introduced in Chapter 3 provides a room label for a set of planar patches by analyzing the spatial layout of these patches. As the computed 3D features only capture global scene information intermediate scene structures like "table", "shelf", or "sofa" are not available. On the way towards a completely spatial aware robot such knowledge is important for understanding tasks like "Please, fetch the bowl on the table, which stands in front of the sofa!". The set of bottom-up extracted planar patches contains a mixture of meaningful and non-meaningful structures. The challenge is to find a representation which encodes only informative intermediate structures. For a smooth communication this representation and its level of details should be aligned to the one of the human tutor which means that similar structures should be represented with similar labels [Vaso7a, Pico4]. As human scene models differ over different humans, tasks, and situations, a generation of a universally valid model is not desirable or even possible. In principle, three strategies for communicating meaningful structures to a robot are thinkable. The robot could be taught explicitly important spatial structures by the human companion, it could take the initiative by iterating through perceived patches and asking for information, or it could be equipped with abilities for inferring meaningful structures during ongoing interaction. In a long-running interaction between human and robot all three strategies will be applied. During an initial introduction of a room, some important elements in the room will be presented roughly to the robot. As the introduced structures will not cover every spatial structure in the room, the robot could demand information about the missing ones by asking for additional information, e. g., [Pelo9]. The challenge for the robot is to guess when it is appropriate to take the initiative and ask questions. Also, the relevance of spatial structures changes with the tasks given to the robot. In the above example instruction the "sofa" and the "table" are important while the "cupboard" also contained in the room could be neglected. If the task would be to find something in the "cupboard", the relevance of the "cupboard" should increase. These relevance shifts have not be incorporated, so far. Therefore,

the third strategy is necessary. It allows the robot to infer the current relevant spatial structures. As within an instruction relevant structures are communicated by a spatial description, I focus here on the development of mechanisms for inferring scene models which hold spatial structures contained in general verbal descriptions of a room scene. The way verbal scene descriptions are constructed communicates which scene elements are currently relevant for the task or to the tutor (→ Section 4.1). This chapter is going to present how scene descriptions and bottom-up planar patches can be utilized to come up with an *Aligned Scene Model* providing meaningful structures with semantic labels. I suspect the envisioned descriptions to arise during a "home tour" where a human guides the robot into a room and describes what it can see in the room (→ Section 2.2.2 and 2.2.3).

This chapter is organized as follows: Section 4.1 provides insights from literature on the nature of spatial descriptions and shows why the descriptions can be utilized for building-up high-level scene models. Section 4.2 presents related work on models providing such semantic information. Section 4.3 gives an empiric analysis of spatial descriptions about two vista space scenarios. Section 4.4 explains the computational approach for generating the aligned scene model. It consists of rules for transforming a description into a set of trees, which encodes the given relations in a hierarchical way, and a grounding of the abstract trees to the visual perception of the according scene. This connection is established by utilizing 3D locations of detected objects and bottom-up extracted planar patches. In Section 4.5 the approach is applied to 30 descriptions from two different rooms. The resulting models are analyzed for consistency and recurring structures. Further, different errors in object detection are tested for their influence on the model generation process. Section 4.6 summarizes this chapter.

4.1 MOTIVATION

From psychological research we know, that the state of affairs is represented mentally by *situation models*. For example, Zwaan [Zwa99] has analyzed situation models arising during narration comprehension. Results from research on situation models in narrative comprehension suggest that comprehenders behave as though they are in the narrated situation rather than outside of it. The comprehension is influenced by the nature of the situation not by the structure of the text. In general, situation models arise as multi-dimensional representation of situations under discussion [Pic04] and encode space, time, causality, intentionality, and reference to main individuals [Zwa98]. People have for every situation their own representation of it, but, according to Pickering and Garrod [Pic04], these representations become *aligned* to the representation of the communication partner if the partners start a conversation. The term "alignment" originates from the research on comprehension in dialog situations. It defines a subconscious adaptation of representations at different levels. Such levels could be word choice, syntactic constructions of sentences, or interpretations of situations. Pickering and Garrod have presented as a catchy example a maze that can be represented as arrangement of patterns (like "right turn indicator, upside down T shape, or L on its side") or as a network of paths linking prominent points (e. g., "the bottom left corner"). Alignment means that both communication partners develop the same representation. Alignment is not necessary for a successful communication. But a dialog becomes more effective with alignment, as the partner's representation needs not to be modeled in addition to the own model. So far, only language has been examined as communication channel for alignment. But recently, other modalities like vision, gestures, or facial expressions [28] have raised research interest. The goal of this chapter is to develop a mechanism that enables the robot to infer a model of *space* present in its sensory input. It should be aligned to the tutor's situation model concerning *space*. The resulting aligned scene model will provide semantic structures the human has had in mind and will map them on perceived sensor data.

As I focus on modeling of space, especially vista space, *spatial descriptions* play an important role in exchanging concepts about the surrounding. Studying spatial descriptions can reveal insights on people's representations of space. Many linguistic experiments have already focused on analyzing spatial language [Sku04, Reg01, Tve98]. They show parallels in the way cognition and language schematize the spatial world [Tve98, Fre08]. According to Talmy [Tal83], language provides a systematic framework to describe space by selecting certain aspects of a referent scene while neglecting others. From this, Tversky and Lee [Tve98] have concluded that language will be successful in conveying space. For a robot this means, that it is reasonable to consider descriptions of space during the model building process. This follows also Waltz's premise where he has assumed that scene descriptions allow a hearer to build models similar to those the speaker has built via perceptual processing [Wal80]. Further, he has postu-

28 http://www.sfb673.org/. In the project A4 we contribute with our approach to the question of how to realize alignment based on linguistic and vision input.

lated that an entity would need a sensory system for comparing representations generated from the input data with those generated from verbal descriptions. Instead of just comparing both representations, my approach attempts to integrate both representations following the "alignment"-paradigm introduced in the previous paragraph. The generated high-level model meets the description content and the scene perception. As descriptions are underspecified [Wal80] visual input can help to solve ambiguities arising from descriptions.

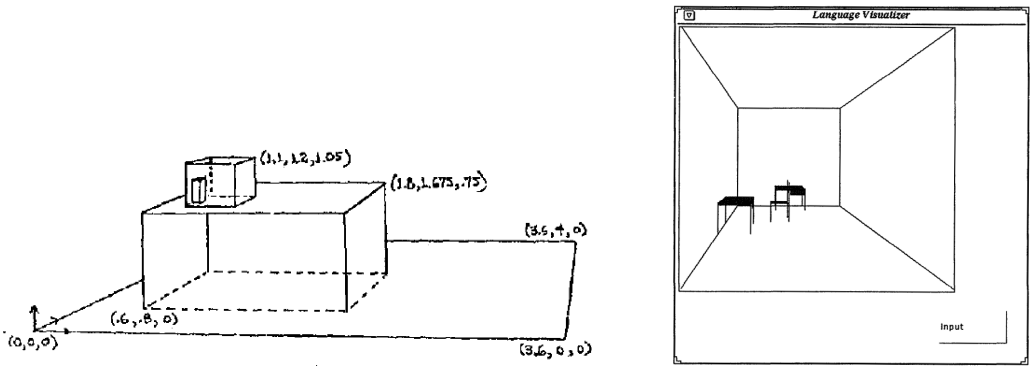
The nature of spatial descriptions and the corresponding cognition have been examined in more details by Hirtle and Jonides [Hir85]. They have found evidence for a hierarchical organization of spatial knowledge. To Tversky and Lee [Tve98], the hierarchical organization is visible through the decomposition of space into figures and spatial relations showing schematically their topological nature. This means that figures are located relatively to other figures or reference frames. As people's conception includes knowledge about gravity and mobility of objects [Tve99] their reference frames are often horizontal and vertical planes. Further, there is a tendency to use relatively large and fix objects as references [Her85]. The most frequently used spatial relations are prepositions like "at", "on", "in", "in front of", "on top of", or "parallel to". The empirical analysis in Section 4.3 shows that these findings are also visible in our scene descriptions collected for two home-tour scenarios. It also presents new insights for deriving models from spatial prepositions. Further work on understanding locative expressions will be discussed there [Reg01, Tom98, Log96, Gap95, RS88, Hut79]. To conclude, the essential step of the aligned scene model is a transformation of spatial object relations to a hierarchical representation of the described space. This representation estimates intermediate 3D scene structures fitting both the description schema and the perceptual reality.

4.2 RELATED WORK

As our computational model is combining verbal descriptions and bottom-up visual processing, related work about methods for deriving scene models from verbal descriptions only and from visual structures only are introduced. Further, work is discussed which deals with the integration of verbal and visual scene interpretations. Finally, the contribution of my aligned scene model to this research field is outlined.

4.2.1 *Scene Interpretation from Verbal Input*

First systems for depicting scene knowledge from descriptions have already been developed in the late 70's. Boggess [Bog79] has developed a program that accepts spatial prepositions like "the box is on the table, the table is on the floor, the floor is in the room" and creates a 3D box model only based on this sequence. The mentioned objects are modeled by open boxes of standard height and weight resulting in a model satisfying the gravity conditions. The model is called Spatial Analog Model and looks like displayed in Figure 4.1(a). Their aim was to develop a representation for linguistic scene descriptions compatible with representations generated with a vision system [Wal80]. Similar to this early approach, the Words-into-Pictures approach of Olivier and colleagues [Oli94] automatically generates 3D depictions from natural language descriptions like shown in Figure 4.1(b). They model objects qualitatively including the explicit representation of their deictic and intrinsic sides and quantitatively through constructing them from a finite set of geometric primitives. Unconstrained degrees of freedom are set to default values. The given spatial prepositions are modeled as potential fields incorporating constraints on orientation and position of an object located relatively to a reference object. Computations of the fields' minima provide acceptable interpretations of the given spatial predication.



(a) Spatial Analog Model of Boggess [Bog79]

(b) Words-into-Picture approach of Olivier et al. [Oli94]

Figure 4.1: Example results of two approaches for depicting scene knowledge from descriptions.

4.2.2 *Scene Interpretation from Visual Input*

The function of scene descriptions is to provide high-level knowledge that support the interpretation of a visually perceived scene. Such semantic analysis of 3D scenes is, for example, a meaningful labeling of compact sets of points or extracted planar patches. Such labeling can be achieved in a top-down manner using an ontology or in a bottom-up manner using proper classifiers. Nüchter and others [Nüc08, Can02, Gra97] have used semantic nets implementing general knowledge about the corresponding context, e. g., indoor environments. Planes are labeled according to their relative orientation or by inference on the given knowledge database. As example, Nüchter's semantic net and the resulting scene model is presented in Figure 4.2. Alternatively, a semantic labeling is also possible with bottom-up based approaches. They rely on training of appropriate classifiers for point-based classification providing a meaningful label for each data point. In general, each point is transformed into a feature vector encoding its special characteristic which could be neighborhood characteristic, orientation to a horizontal and vertical plane, or its normal. Based on such feature vectors each point is classified into given classes which could be, for example, {`wire`, `vegetation`, `tree trunk`, `facade`} [Mun09a], {`chair`, `table`, `screen`, `fan`, `trash can`} [Tri07] or {`plane`, `sphere`, `cylinder`, `cone`, `torus`, `edge`, `corner`} [Rus08]. Figure 4.3 shows Triebel's labeling of an indoor scene.

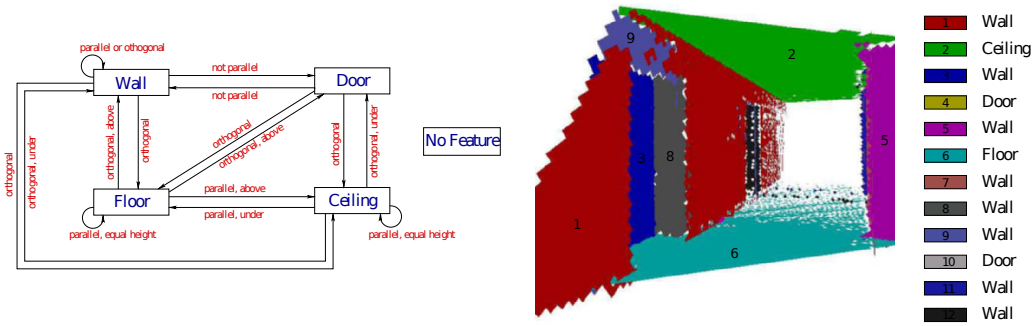
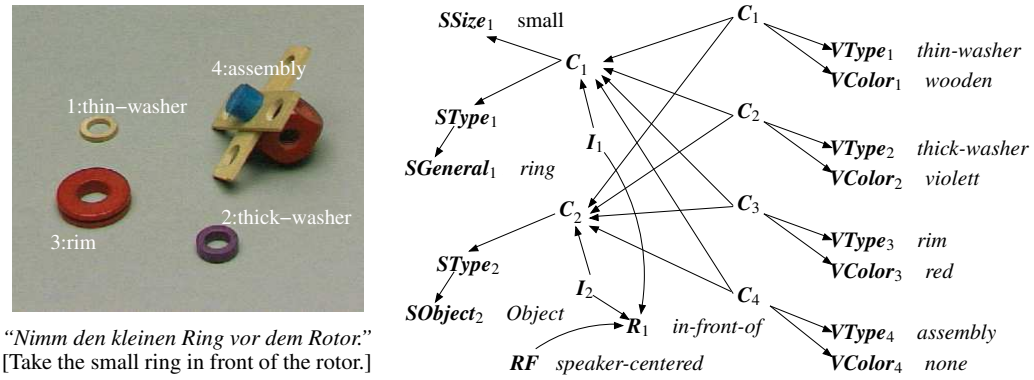


Figure 4.2: This figure shows the semantic net Nüchter applied to a set of extracted planar resulting in the semantic labeling of the patches displayed to the right [Nüc08].

Figure 4.3: This figure shows a semantic labeling of 3D points of an indoor scene using the point-based classification proposed by Triebel [Tri07].



"Nimm den kleinen Ring vor dem Rotor."
[Take the small ring in front of the rotor.]

Figure 4.4: Wachsmuth et al. [Waco2] show here their Bayesian network which is used for integrating speech and image interpretations given in the faced table-top scenario.

4.2.3 *Integration of Verbal and Visual Scene Interpretations*

According to Waltz [Wal80] scene descriptions are underspecified. This means that spatial descriptions has to be correlated with visual perceptions for a full spatial awareness. Achieving a suitable combination of both modalities is a challenging problem. Therefore, researchers have first started to find ways for matching simple projective spatial relations between objects like "above", "left", "right", or "below" on visual perceptions of the corresponding situation. Studying the generation of such relations can give insights for developing mechanisms for judging whether a given spatial relationship fits the perceived reality. Methods for generating relations are histograms of angles [Miy94], histograms of forces between objects [Mat99], acceptance volumes [Soco0, Vor97], and attentional vector-sum models [Reg01]. Acceptance volumes rely on inducing a binary acceptance relation that expresses whether an object intersects with the volume scoring it by calculating the corresponding degree of containment. Attentional vector-sum models combine orientational and height components of objects. Based on the gained insights robots have been equipped with modules for understanding commands like "go to the right of the object". For example, Skubic et al. [Sku04], are able to compute the correct target destination in unoccupied space for the four primary directions "left", "right", "front", and "rear" of an object with respect to the robot's perspective. They have utilized force histograms which provide confidence about how well a position meets the instruction. Moratz et al. [Mor01] have found in their experiments that humans mostly take the robot's perspective. Therefore, they have equipped their robot with an ego-centered 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.

The next challenge has been the integration of scene descriptions and visual perception where either the descriptions get more complex and the scenarios stay simple or vise versa. Simple scenarios are single table-top setups like a table-setting or a grasping scenario. Single caption words accompanying pictures of scenes are said to be simple descriptions.

For the first case, Wachsmuth and Sagerer [Waco2] have integrated verbal and visual descriptions in a probabilistic manner using Bayesian networks. Object descriptions from vision and language as well as relations between objects have been modeled as nodes in this network which is sketched in Figure 4.4. The system determines the desired object in instructions like "Take the small ring in front of the rotor" by a Bayesian inference process. The probabilistic approach is also followed by Mavridis and Roy [Mavo6]. They model their table-top scenario as a stochastic layer where vision percepts are integrated as well as "imagined" descriptions of unseen scene parts like "there is a blue object at the left". New information, either visual or linguistic, is continuously added by updating the probabilities in the stochastic layer. Another approach is followed by Brenner et al. [Breo7]. They have built a robot that can manage instructions like "put object1 to the left of object2". This qualitative description is first transformed to a potential field in the continuous space. It is then mapped to a geometric description like way-points which are passed together with symbolic representations of visible objects to a planner that generates acceptable actions. An integration of visual perception and high-level concepts through description logics is proposed by Neumann and Möller [Neuo8]. There, table-setting scenes are transformed into partial geometric scene descriptions. Symbolic constants from a given concept are then connected to individual entities in the scene using description logics resulting in an interpretation of the current scene. An approach combining the probabilistic and logical area in scene analysis is proposed by Hois et al. [Hoio8]. Their system analyzes a partial 3D scene, e. g., a table-top setting. First, objects are extracted as compact 3D point clouds located on planar patches. Second, objects are identified automatically. And third, assistance of a user is demanded to resolve unknown objects. The detected objects are integrated in a domain ontology which models the objects of the presented office scene and the spatial relations between them. The spatial relations are estimated from the arrangement of objects in the scene using heterogeneous non-overlapping acceptance areas of [Her94]. This allows the user to ask the system during the action phase questions about objects including their relations, e. g., "what are the objects to the right of the stapler?".

The previous work has in common that linguistic information is either used in or derived from simple scenes. In contrast to that, recent work from computer vision, e. g., [Jam10, Wano9], enhances visual interpretation like object classification in complex scenes by using words given in the caption accompanying an image. The idea is to learn strong correspondences between names and visual features during training of classifiers.

4.2.4 *Contribution of the Aligned Scene Model*

Summarizing this section it can be stated that bottom-up extraction of high-level scene descriptions from scene percepts, e. g., using semantic nets, leads to a predefined and static model (→ Section 4.2.2). It does not take into account the representation of the communication partner which changes depending on the situation. It cannot be ensured that mentioned scene parts are modeled. This could mean that the robot cannot solve the task. 3D modeling of the interlocutor's representation has only be done on an abstract level without grounding it into the visual reality (→ Section 4.2.1). So far, the link between descriptions and real percepts has only been established for complex descriptions and simple table-top scenarios which can be handled in 2D or simple caption words and complex environments (→ Section 4.2.3). The new computational model presented in this chapters attempts to close the observed gap in 3D scene analysis. The goal is to infer the partner's structural model from complex scene descriptions and to ground it in the perceived visual 3D data of a complex environment. In more details, the outcome is an aligned scene model that estimates the interlocutor's representation of space and grounds it to the perceived planar patches. The purpose is to give the robot knowledge about meaningful scene structures and their localization in the real world. The scene layout is concluded from objects and relations between them.

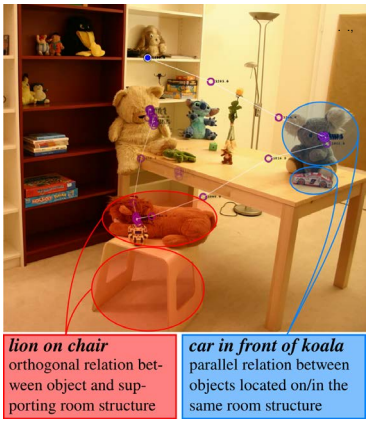
4.3 EMPIRICAL ANALYSIS OF SPATIAL SCENE DESCRIPTIONS

Here, the focused scenarios are vista space scenes appearing during a "home tour" where a human tutor guides the robot to a certain room or room part (→ Section 2.2.3). The robot stays still and records the scene with its 3D sensor while the tutor describes what can be seen. For analyzing these vista space descriptions, my colleague Constanze Vorwerg has designed a study to collect these descriptions in a controlled way [Swa09]. I have made photos of 2 scenes, a children's playroom and a living room. At the same position SwissRanger data is captured with the camera positioned on a tripod at a height and orientation comparable to the camera mounted on our BIelefeld Robot CompaniON (BIRON) platform (→ Section 2.2). Figure 4.5 shows the 2 photos and the corresponding 3D point clouds. Extracted planar patches as bottom-up scene representation are displayed on the right side (for the extraction algorithm see Section 2.4.2). In the pilot study the picture of scene $\mathcal{S}_1$ was presented as print-out to 10 students (all native speakers of German). Their task was to describe what they "see in the picture". In the second study 10 students (all native speakers of German) has been presented scene $\mathcal{S}_1$ and $\mathcal{S}_2$ on a computer screen. The task has been to describe what they "see in the room". These room depictions are *true scenes* [Hen99]. They are views of a natural environment that is semantically coherent and contains both background elements and genuinely spatially arranged objects.

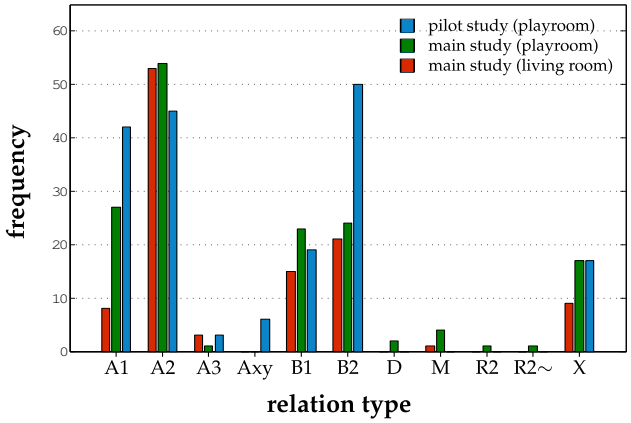
(a) scene $\mathcal{S}_1$: playroom

(b) scene $\mathcal{S}_2$: living room

Figure 4.5: These are the two scenes (a) $\mathcal{S}_1$ and (b) $\mathcal{S}_2$ which have been recorded for the study. The subjects have been told to freely describe the photographs. On the right, plots of the acquired 3D point clouds are displayed. The colors encode the extracted planar patches from the point cloud.



(a) Two relation types.



(b) Frequency of relation types.

Figure 4.6: (a) visualizes the two types of relations which are orthogonal relations between objects and its supporting structures and parallel relations between objects located on the same structure. (b) gives the distribution of relations appearing across all acquired scene descriptions [Vor09].

Alternatively, many psycholinguistic studies use *ersatz scenes* which are displays of arbitrary arrayed objects [Hen04]. Here, true scenes are used as it can be assumed that the realistic setup will lead to realistic descriptions. A robot like BIRON can then utilize the given descriptions directly as it can observe the same scenes with its sensors. The descriptions are analyzed for new insights about vista space descriptions. Furthermore, these descriptions can simulate descriptions given to a robot. Whereas, in a natural human-robot dialog a scene descriptions will not be given in total as a block of relations but sequence-wise providing one relation after the other. The robot is expected to give feedback from time to time showing that it still follows the human tutor. As my computational model processes descriptions sequentially, such a dialog design can be realized straight-forward.

My colleague Constanze Vorwerg, a researcher in psycholinguistics, has conducted the experiments and has analyzed the data [Vor09]. In the following, I am going to summarize the results and to describe consequences for the development of a computational model generating a scene model from spatial descriptions. Vorwerg's analysis shows that spatial room structures like pieces of furniture or other room parts serve as crystallization points for room descriptions. Typical relations are for example "the lion is on the chair" or "a toy car is in front of the koala" (see Figure 4.6(a)). Objects are put into relation with their supporting room structure or with other objects on the same room structure. Therefore, spatial aspects of visual scenes seem to be memorized hierarchically as already proposed by Hirtle and Jonides [Hir85].

Spatial relations like "in front of", "left", "right", "above", "below", "on", and "in" are an important part of spatial descriptions and have been examined exhaustively. Spatial templates [Log96] and angular deviations [Gap95] have been found to play an important role in understanding such relations. Traditionally, it is assumed that inferring the correct frame of reference is an essential task even though it is challenging [RS88]. Here we show that it is possible to model spatial

relations *without* knowing the used reference system by utilizing the observation that objects are related to their supporting structures or to other objects located on the same supporting structure. We assign all relations to a *super-ordinate* structure like "on the table" or "in the cupboard" to a so-called *orthogonal* relation type. *Co-ordinate* relations between elements located on the same super-ordinate structure like "on the right side of the sofa is a table" are referred to as *parallel* relation type. This view and reference frame independent modeling of spatial relations gives a nice methodology for extracting hierarchical representations from given spatial descriptions (→ Section 4.4.1). The importance of these orthogonal and parallel relations is also emphasized by a quantitative analysis of the descriptions gained in the studies. Table 4.1 lists all relationship types which can occur. Summed over all descriptions, Figure 4.6(b) shows the counts for each relation type. The majority of specified relations belongs to the parallel (**B**) or to the orthogonal relation type (**A**, nearly never **R**). Relations between same-level objects from different super-ordinate structures almost never occur (**D**). This leads to the assumption that the descriptions are organized hierarchically. Therefore, relations between objects can be used to derive knowledge about their supporting structures. This knowledge can facilitate the visual processing since objects occlude their supporting structures at least partially. Therefore, it is more robust to detect these objects and to infer the supporting structure from given spatial relations than to identify these structures in a bottom-up way in visual data. Our computational model supports the visual detection of spatial structures. While the visual processing that provides objects and their 3D positions simplifies the understanding of spatial depictions as only a distinction between parallel and orthogonal relations is necessary.

A detailed analysis of a playroom description is given in Figure 4.7. Each entry of the table is a relation given in the depiction. The last column holds the assigned relation type. The description is organized in a structured way. The participant follows the spatial layout by scanning one piece of furniture after another (red shelf, white shelf, table, stool). All relations to a specific structure are given as a block of successive relations. The connected blocks are highlighted in the figure by different colors. Relations introducing a new spatial structure mostly relate the new structure to an already mentioned one. The remaining relations in a block focus on objects located on/in the corresponding spatial structure. Here, a main pattern is that the first object is related orthogonally to the supporting structure while the other objects are related to each other through parallel relations. This strategy of describing a super-ordinate relatum with its objects followed by a related or next super-ordinate relatum is called *structure-detail strategy*. Vorwerg has also found other linearization patterns: *overview first strategy* first lists super-ordinate relata (chair, table, shelves) and then relates small objects to them and the *overview only strategy* simply itemizes objects and furniture with respect to the room ("in the room is a chair, ..., a koala, ..."). This strategies are more advanced compared to the basic strategy reported by Vasudevan et al. [Vas07b] where objects are described from one to the other.

A	super-ordinate relatum (orthogonal relation) A1 relatum is room or room part (wall, floor, ceiling) A2 relatum is furniture A3 relatum is object
B	co-ordinate relatum (parallel relation) B1 relation between furniture, the super-ordinate structure is the room or a room part B2 relation between objects, they have furniture as supporting structure
M	meta relatum (room as relatum for small objects)
D	same-level object from different super-ordinate structure as relatum
R	reverse of A (occurs only once following an according A localization)
R∼	seeming reverse of A (paraphrased with "is where ... stands")
X	object is localized relative to the image plane ("in the top-left corner of the image")

Table 4.1: This table lists the relationship types which can occur in spatial descriptions [Voro9].

Spatial relation & relatum	Things to be localized	RT
In dem Raum befinden sich In the room are	Spielsachen toys	M
Da ist There is	ein rotes Regal *mit einem schwarzen Raben und ... a red shelf *with a black raven and 2 ... figures	A1/ *A2
Darunter Below of them	ein Hase und ein Frosch a hare and a frog	B2
Darunter befinden sich Below of them	Bücher books	B2
und darunter befinden sich and below of them there are	Spiele games	B2
Daneben ist Beside of it is	ein weißes Regal a white shelf	B1
dort sitzt there sits	ein Hase a hare	A2
da ist There is	eine Schale a bowl	B2
Daneben sind rechts daneben Beside of them there are to the right	Arbeitsmaterialien printed materials	B2
Darunter befinden sich Below of them there are	Spiele games	B2
Daneben Beside of them	ein Kerzenständer a candlestick	B2
in der Mitte # des Raumes In the center # of the room	Dann ist da ein Tisch Then there is a table	A1#
darauf befinden sich auch on it there are also	Stofftiere, ähm eine Vase <>, ein Auto stuffed animals, um a vase <>, a car	A2
<... darin> <... in it>	<mit einer Blume ...> <with a flower' ...>	A3
Davor ist In front of it is	ein Hocker a stool	B1
Darauf befindet sich On it there is also	ein Löwe a lion	A2
Davor befindet sich In front of it is	irgendein roboterartiges Spielzeug some robot-like toy	B2
und hinter dem Tisch steht and behind the table stands	´ne Lampe a lamp	B1

Figure 4.7: An example description of a playroom is transformed to a list of relations with the first column holding the relatum and the given relation and the second column the referenced object. The last column (**RT**) specifies the relation type: **A** indicates an orthogonal relation and **B** a parallel relation. Additional details to the other relation types can be found in Table 4.1. Relations to a specific structure like a table are tagged with the same color. The arrows visualize interdependencies given between higher-level supporting structures. This diagram is taken from [Voro9].

4.4 THE COMPUTATIONAL MODEL

This section introduces the computational model for acquiring the *Aligned Scene Model*. It reflects the tutor's scene representation which is communicated by a verbal description and the visual reality perceived as bottom-up extracted 3D planar patches. Figure 4.8 gives an overview over the necessary steps. Based on the results of the empirical analysis of spatial descriptions (→ Section 4.3) a methodology is developed that transforms a verbal description to a representation keeping the hierarchical character of the description. This hierarchical character can be encoded suitably by a set of trees. The trees are constructed by applying structuring rules specifically designed to handle *orthogonal* and *parallel* spatial relations. Details on this processing step are given in Section 4.4.1. The next step is presented in Section 4.4.2. Given an object detector that provides small objects with their 3D positions this output is used to infer the hypothetical position, size, and orientation of the spatial structures given in the set of trees. As scene descriptions are ambiguous and underspecified, Section 4.4.3 introduces a way to solve this problem by integrating the visual reality. Bottom-up extracted

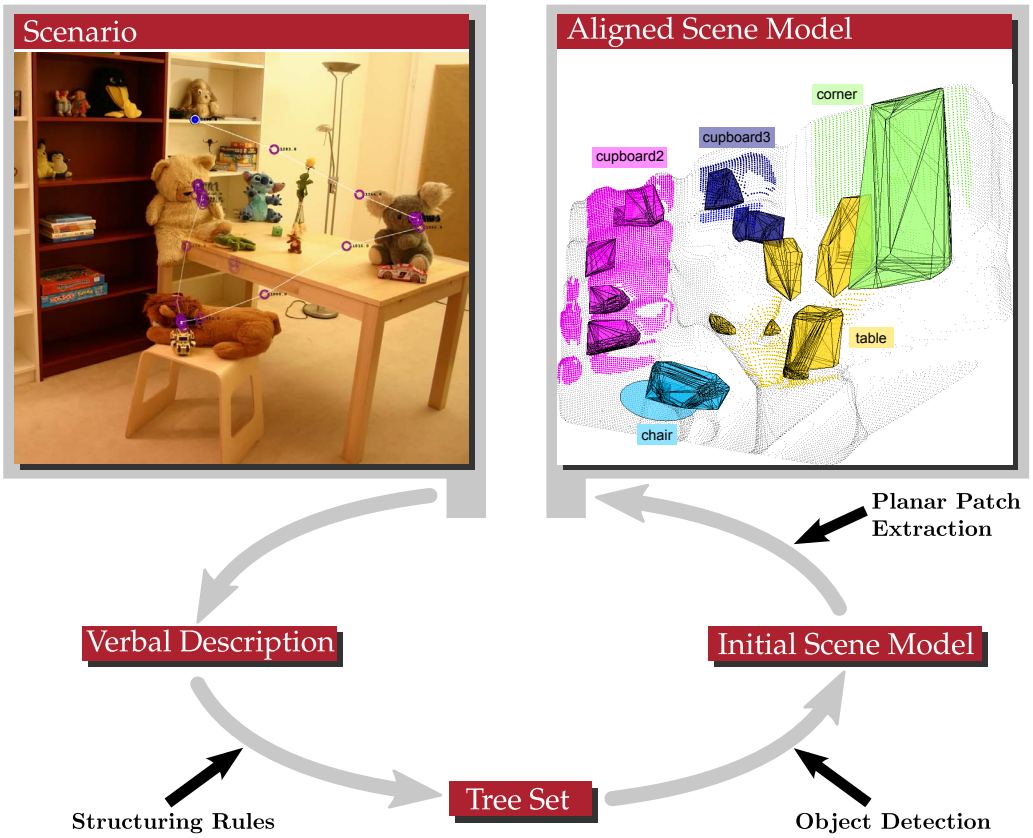


Figure 4.8: The main steps for acquiring an aligned scene model using the interplay of verbal description and visual perception are: (1) transforming a spatial description into a hierarchical representation, it consists of a set of trees generated from specific structuring rules, (2) estimating an initial scene model using 3D object positions provided by an object detector, the model assembles from supporting structures given in the tree set, (3) matching and adapting the initial model to the visual reality resulting in the final aligned scene model, more precisely the potential patches from the initial model are matched to the bottom-up extracted 3D planar patches.

3D planar patches are used to adapt and correct the initial scene model. The resulting aligned scene model meets both the intermediate structures provided by the human tutor and the robot's visual perception. I have presented the main idea of this work at the International Joint Conference on Artificial Intelligence in 2009 [Swa09].

4.4.1 *From Verbal Descriptions to Set of Trees*

As outlined in Section 4.1 spatial descriptions reveal information about the situation model of the communication partner. Due to an empirical analysis (→ Section 4.3) spatial descriptions mainly consist of *explicit* and *implicit* relations to supporting structures. These relations are named orthogonal and parallel relations and defined as follows:

Definition. ***Orthogonal relation.***

A relation is of the orthogonal type if an object is related explicitly to its super-ordinate structure as in, e.g., "a lion on the chair".

Definition. ***Parallel relation.***

A relation is a parallel relation when two items localized on the same structure are related to each other like in "a car in front of the koala (both objects are lying on the table)". The supporting structure (here, "table") is referenced implicitly.

This definition follows the basic physical fact that gravity causes objects not to float in the air but to be placed on tables, attached to walls, or contained in cupboards [Tor09, Oli94]. These explicit and implicit references to super-ordinate structures reflect the hierarchical character of the underlying situation models, so that trees are a suitable structure for maintaining such models. Unfortunately, human-given descriptions are incomplete, ambiguous, and are not provided ideally arranged for tree construction. Consequently, a *set* of dependency trees, indicated by $\mathcal{T}$, is a proper representation format for the spatial content given in a description. The objective of this section is to illustrate the transforming of a scene description to a suitable set of trees.

Scene descriptions can be assumed to be sequences of object relations. At any point in time there exists a tree set $\mathcal{T}_{t-1}$ (at the beginning of a description it will be the empty set $\mathcal{T}_0 = \emptyset$). The current relation updates this tree set $\mathcal{T}_{t-1}$ to a new tree set $\mathcal{T}_t$ following certain rules. Our system has to deal with two types of relations, parallel and orthogonal one. The parallel relations are relations between items on the same level and orthogonal relations are relations between items at different levels in the model hierarchy. The structuring rules for handling these two reference types are presented below, but first I would like to give a short introduction to my notations.

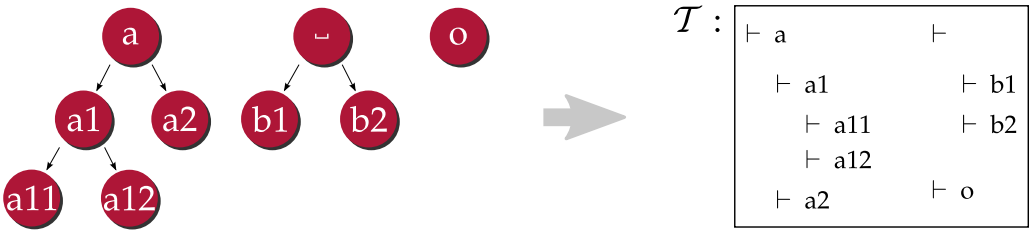


Figure 4.9: On the left side, a set of three trees is visualized as three directed graphs. For example, the node a is the parent node of the nodes a1 and a2. The right side shows the compact notation of the set of three trees. All children of a node are listed directly below the parent node. The indentation before a node (denoted by ⊢) corresponds to the position of the node in the tree. The node label can have a specific value (like a, b1, ...) or can be empty if the supporting structure is not named explicitly.

Figure 4.9 introduces a compact illustration of a tree set. Nodes are indicated by ⊢. The size of the indentation before a node corresponds to its position in the tree. All children of a node are listed directly below it. The node labels could have a specific value like "a", "a1", ..., or could be empty like the root node of tree b. An empty label represents the fact that the supporting structure has not be named explicitly which is the case when a parallel relation between two objects is specified. The nodes of the trees correspond to scene items in general while the edges indicate the relations between them. Due to the hierarchical character of scene descriptions, each parent node constitutes the supporting structure of all its children. This interpretation goes along with the fact that descriptions consist of naming objects and providing relations between them. A typical expression like

"o1 is related to o2"

can be formally written as

$$o_1 = \texttt{obj}(\text{“o1”}), \quad o_2 = \texttt{obj}(\text{“o2”}), \quad \texttt{rel}_{\{\parallel,\perp\}}(o_1, o_2)$$

assuming that the type of the relation can only be parallel or orthogonal. A tree set is updated according to certain rules:

Definition. ***Rule.***

relation $\Rightarrow$ *sequence of tree operations.*

The available tree operations are three modification operations and one query operation:

Definition. ***Tree operations.***

`obj(.)`	*adding nodes,*
`delete(.)`	*deleting nodes,*
`child(.,.)`	*adding edges, and*
`ischild(.,.)`	*indicating the existence of an edge.*

The following listing gives details for these four tree operations:

$o = \texttt{obj}(\text{“o”}) \rightarrow n_o$

An object label “o” in a description means that there exists an object o. The `obj`-function returns a pointer ($\rightarrow$) to the object node n_o in $\mathcal{T}$ representing the mentioned object. Depending on the type of the object label “o” different inserting behaviors are required. If an object label can be grounded to exactly one object in the scene than it is called *distinct object* label. Distinct object labels are, for example, “the blue doll” or “a radio”. In the first case, the robot knows two dolls but the accompanying adjective allows to determine the doll that have been referenced. In the second case, the robot knows only one radio in beforehand so that it can easily assign the label “radio” without having to resolve ambiguities. A node representing a distinct object label is added to the tree set $\mathcal{T}$ when the object is referenced the first time. All further references are handled by just returning a pointer to this node. An object label matching several objects is called *category* label. This is the case if the label is in plural like “soft toys” or if a distinguishing adjective is missing when a label fits more than one object like “a doll”. Category labels are handled by creating a new node in $\mathcal{T}$ every time this category label occurs as it cannot be said whether the same objects are meant or not. There exists only one exception when directly consecutive relations contain the same category label. Here, it can be assumed that the same objects are meant so that it is reasonable to provide for the second relation a pointer to the corresponding node generated in the preceding relation.

$\texttt{child}(n_o, n_p)$

A directed edge is inserted from the parent node n_p to the child node n_o expressing the supporting characteristic of the parent structure to the child object.

$\texttt{bool} = \texttt{ischild}(n_o, n_p)$

`true` is returned if the nodes n_o and n_p exist in $\mathcal{T}$ $\left(\exists\{n_o, n_p\} \in \mathcal{T}\right)$ connected with a directed edge from n_p to n_o. Else, `false` is returned.

$\texttt{delete}(n)$

The node n is deleted from the tree set $\mathcal{T}$.

HANDLING PARALLEL RELATIONS. All relations between object o_1 and o_2 assigned to the *parallel* type, $\texttt{rel}_{\parallel}(o_1, o_2)$, are of the form

"o1 lies in front of /behind /next to /above /below o2"

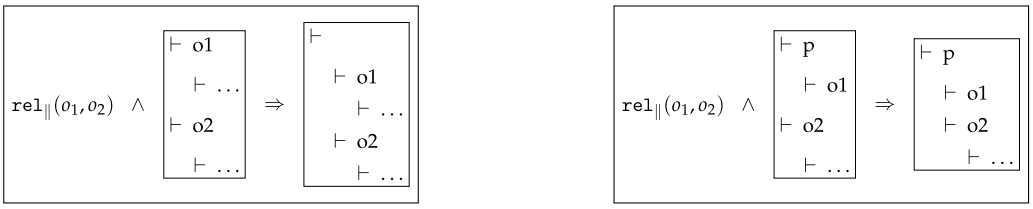
like "the car is in front of the koala". Due to the empirical finding that only objects located *on* or *in* the same super-ordinate structure are related in this way, it can be inferred that both objects are located on the same supporting element (e. g., the table). Hence, the basic rule for updating $\mathcal{T}$ given $\texttt{rel}_{\parallel}(o_1, o_2)$ is:

$$\begin{aligned} \texttt{rel}_{\parallel}(o_1, o_2) \quad \Rightarrow \quad & \exists p = \texttt{obj}(\text{“ ”}) \rightarrow n_{\text{p}} \text{ with} \\ & \texttt{child}(n_{\text{o1}}, n_{\text{p}}) \text{ and } \texttt{child}(n_{\text{o2}}, n_{\text{p}}) \end{aligned} \tag{4.1}$$

It states, that there exists a supporting element $p = \texttt{obj}(\text{“ ”})$ for which currently no label is known. This object is inserted as new node n_{p} in $\mathcal{T}$. The hierarchical relation between o_1, o_2 and p is established by inserting edges using the `child`-operation. It sets n_{o1} and n_{o2} as child nodes of n_{p} regardless if the object nodes have further children or not. There exists only one exception when n_{o1} has already a parent node n_{p}. In this case Equation 4.1 is altered to

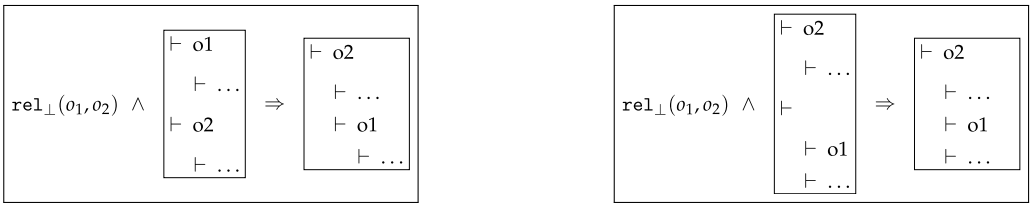
$$\texttt{rel}_{\parallel}(o_1, o_2) \wedge \exists n_{\text{p}} \in \mathcal{T} : \texttt{ischild}(n_{\text{o1}}, n_{\text{p}}) \Rightarrow \texttt{child}(n_{\text{o2}}, n_{\text{p}}). \tag{4.2}$$

This means that n_{o2} becomes with all its children the child of n_{p}. The parallel relation is commutative in the mathematical sense as $\texttt{rel}_{\parallel}(o_1, o_2)$ and $\texttt{rel}_{\parallel}(o_2, o_1)$ point to the same supporting structure thus update $\mathcal{T}$ in the same way. Figure 4.10 illustrates the update behavior of rule 4.1and rule 4.2.



(a) rule 4.1 (b) rule 4.2

Figure 4.10: Visualization of rules handling parallel relations. (a) the basic rule, (b) handling the case that the node n_{o1} has already a parent node n_{p}.



(a) rule 4.3 (b) rule 4.4

Figure 4.11: Visualization of rules handling orthogonal relations. (a) the basic rule, (b) handling the case that the node n_{o1} has already a parent node n_{p}.

HANDLING ORTHOGONAL RELATIONS. A relation having the form

"o1 lies on /in o2"

like "there are soft toys on the table" is called *orthogonal* relation, $\texttt{rel}_{\perp}(o_1, o_2)$. It provides a relationship between an object o_1 and its super-ordinate structure o_2 by locating o_1 relative to o_2. Formally, this can be written down as:

$$\texttt{rel}_{\perp}(o_1, o_2) \Rightarrow \texttt{child}(n_{o1}, n_{o2}) \tag{4.3}$$

Between the nodes n_{o1} and n_{o2}, standing for the named objects, a directed edge is introduced turning n_{o2} to be the parent of n_{o1}. The exceptional case, that n_{o1} has already a parent node, can appear, too. It is handled by the following modification of the basic rule 4.3:

$$\begin{aligned} &\texttt{rel}_{\perp}(o_1, o_2) \wedge \exists n_p \in \mathcal{T} : \texttt{ischild}(n_{o1}, n_p) \\ &\Rightarrow \forall n : \texttt{ischild}(n, n_p) \text{ do } \texttt{child}(n, n_{o2}),\ \texttt{delete}(n_p) \end{aligned} \tag{4.4}$$

This rule is only applicable if no label is specified for this parent node or if the existing label is identical to the label of n_{o2}. In this case both trees or subtrees can be fused to one tree with n_{o2} as the root node. If a label different to the label of n_{o2} is assigned to the original parent node a conflict arises that cannot be resolved automatically. Relations causing this conflict are postponed for clarification in a subsequent dialog with the communication partner. Figure 4.11 visualizes both rules for orthogonal relations showing the tree set $\mathcal{T}$ before and after the update.

A basic principle of the rules is to fuse two originally independent trees when an according association is provided by a new spatial relation. This construction procedure is found to be similar to the assumed procedure in humans building their mental models. Johnson-Laird [JL80] has conducted an experiment where he has examined subjects' mental models constructed from a given instruction. He has contrasted, for example, the continuous description, "the knife is in front of the spoon, the spoon is on the left of the glass, and the glass is behind the dish", with a discontinuous one, "the glass is behind the dish, the knife is in front of the spoon, and the spoon is on the left of the glass". He has found that a task following the instruction is much harder to accomplish if the instruction is given in the discontinuous order. Johnson-Laird argues that subjects first construct two models and combine them afterwards. This is also the main construction principle in the computational model proposed here, because the rules defined cause always a fusion of trees representing the known spatial relations.

(1) In the corner is a lamp.

(2) Soft toys are on the table,

(3) a rose is on the table, and

(4) a car is in front of the koala.

(5) A lion is on the chair.

(6) A small robot lies in front of the lion.

(7) Further, books are in the left cupboard (cupboard2).

(8) Next to fred is a raven.

(9) The pokemon are below the raven.

(10) Games are in the right cupboard (cupboard3).

(11) In cupboard3 there is also a candle.

(12) Above the candle is a dog.

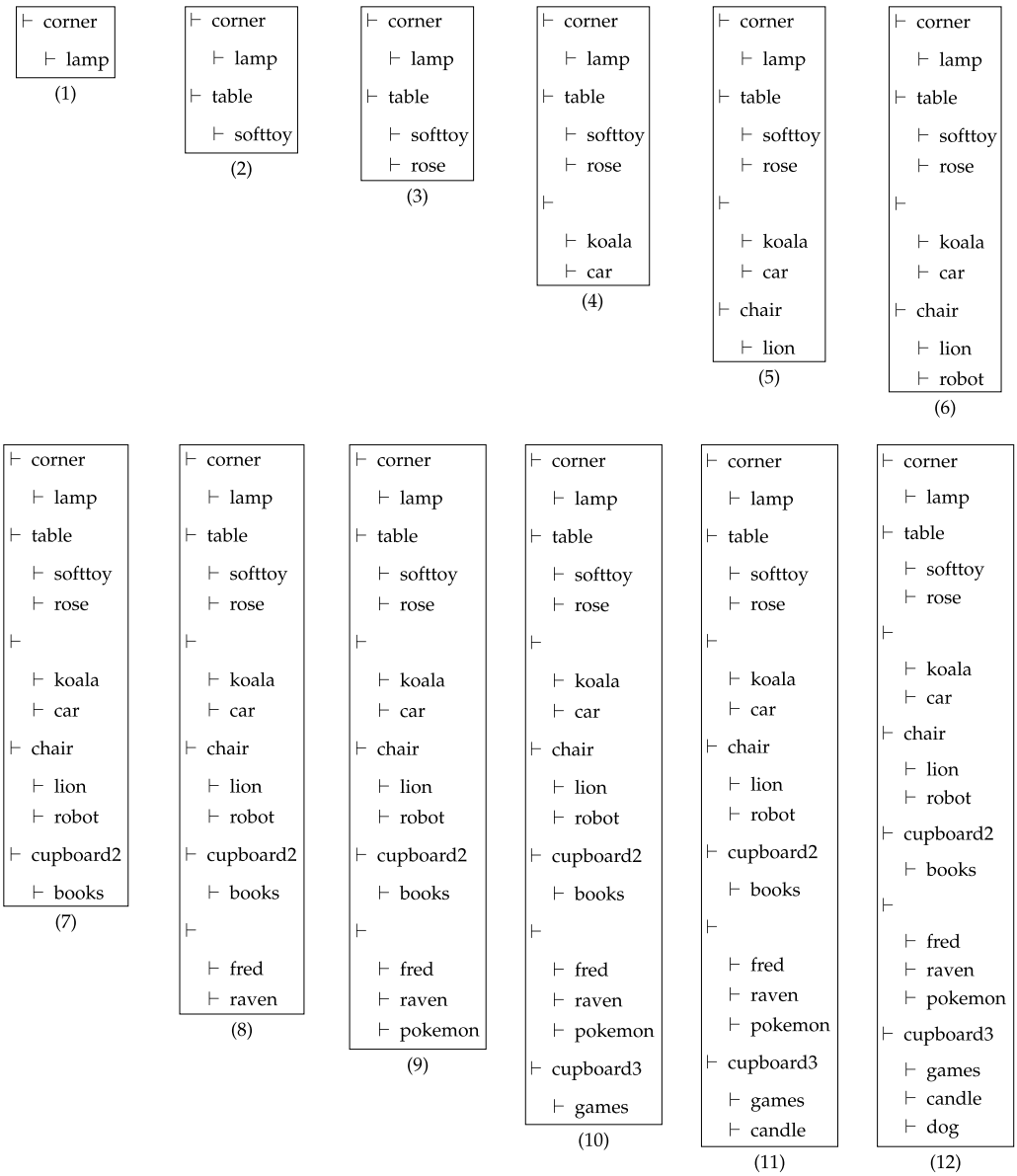


Figure 4.12: Top: Example description about the playroom given by subject 3(p) in the pilot study. The red framed words mark the objects and the green underlined words the relations. Double underlined means parallel relation and single underlined means orthogonal reference. Bottom: If processing the above description as a sequence of relations the displayed development of tree set $\mathcal{T}$ can be observed.

PROCESSING OF AN EXAMPLE DESCRIPTION. Finally, I am going to show the changes of the tree set $\mathcal{T}$ when processing an example description. Here, the scene description of the playroom (→ Figure 4.5(a)) given by subject 3(p) in the pilot study is utilized (→ Section 4.3). The verbal description is interpreted as a sequence of relations, see the top box of Figure 4.12. Figure 4.12 shows also the development of tree set $\mathcal{T}$ while processing the description sequence-wise. The (i)-th tree set is a result of updating the $(i-1)$-th tree set with the (i)-th relation. As an example, I am going to discuss relation (3) ("a rose is on the table") and (4) ("a car is front of the koala") in more details. Relation (3) is a standard orthogonal relation producing a tree in $\mathcal{T}$ with "table" being the parent node and "rose" the child node. As the relations are treated independently from each other, my algorithm does not catch the fact that "car" and "koala" are located on the table. This information is only given implicitly through the connection of the two sentences by the word "and". Directly, it can only be assumed that "koala" and "car" have a common supporting structure which is modeled in the (4)-th tree set by a common parent node with an empty label. Humans infer the information that "car" and "koala" are on the "table" by analyzing the sentence context. For example, they see that both relations are combined by an "and". Alternatively, this information can be concluded from the visual perception of the scene. In my system, I have modeled the second option by implementing a mechanism which allows the robot to infer from bottom-up extracted planar patches trees in $\mathcal{T}$ that can be merged (→ Section 4.4.3).

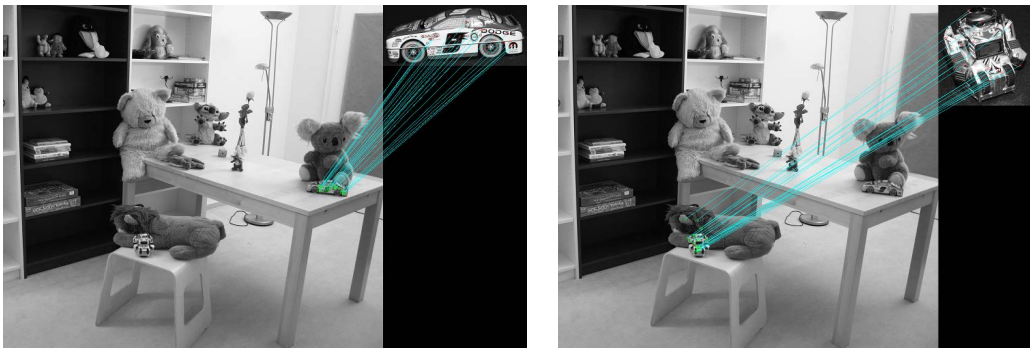
4.4.2 *Inferring Initial 3D Scene Structures*

This section describes how a set of trees can be transformed to a set of 3D planar patches. These patches assemble the *Initial Scene Model* by estimating potential supporting structures of objects given in a description (→ Figure 4.16). The described room is characterized on an intermediate scene level. As man-made environments contain many planar patches, it is reasonable to model supporting structures as planes. Instead of detecting meaningful structures like "table" or "cupboard" in a bottom-up way, the main idea is to infer their 3D location, orientation, and expansion from small movable objects attached to them. I suppose that detectors for small compact objects, which do not contribute to the spatial layout of a room like "lion", "koala", ..., are easier to train and can provide much more stable detections [Nüco8] than detectors for spatial structures like "chair" or "table". The common property of these detectable objects is that they are located in the leafs of my trees. A potential planar surface representing a parent node can be estimated from the 3D world positions of the assigned objects. This models the gravity constraint since no movable object can float in the air. As the camera orientation is known (→ Section 2.2.1) the 3D point cloud can be transformed so that the ground plane is parallel to the xz-plane of a left-handed coordinate system. The orientation of a supporting structure depends on the type of the used relation. Supporting structures where objects are placed *on* them can be estimated as horizontal planes. While supporting structure for objects being placed *in* or attached *at* are typically vertical planes. The second

assumption is not valid in general. In exceptional cases, it could happen that an *at*-relation is used as parallel relation ("lamp at the wall" has let to competing labels for a supporting structure in Figure 4.25(g) so that relations have to be ignored) or that an *in*-relation is better modeled by a horizontal plane. But, the evaluation in Section 4.5 shows that the initial assumptions on the orientation of the supporting patches are sufficient for processing most depictions reliably.

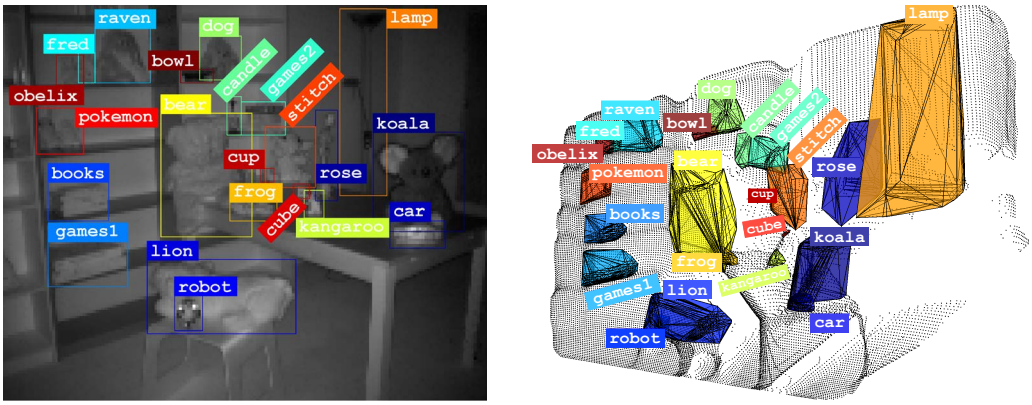
Figure 4.13 shows two outputs of an automatic object detection using Scale-Invariant Feature Transform (SIFT) matching [Low99] and a RANdom SAmple Consensus (RANSAC) based rejection of outliers [Fis81]. SIFT features from several object views are matched to the scene image and rated regarding to their error. The Sift features of the best matching view are enclosed by a 2D box providing an object detection result. If, for example, a calibration of a 2D camera and a 3D ToF camera is given the corresponding 3D object hull can be extracted by mapping the 2D object box into the SwissRanger image. SIFT feature based object detectors work well for textured objects, like the toy car or the toy robot, but will fail for less-textured objects, like the teddy bear. As development of a robust object detection system is out of the scope of this work, objects are labeled manually in the SwissRanger amplitude image simulating an output of an object detector. Figure 4.14 shows all objects known to the robot in the playroom scene $\mathcal{S}_1$ ($\rightarrow$ Figure 4.5(a)). A 2D box is drawn manually around each object in the SwissRanger amplitude image. The pixels within an object box determine the corresponding 3D points which are used (after removing some outlier points lying outside a 3D bounding box) to compute the 3D object hull providing object location and extension in 3D space. The object location is assumed to be the 3D object point with the smallest y-value (due to gravity this point will touch the supporting structure first). Table 4.2 gives the known objects $\mathcal{O}$ and their categories of different degree of universality.

An initial scene model is derived for a set of trees $\mathcal{T}$ by estimating for each parent node on the level above the leaves in $\mathcal{T}$ a potential patch which represents the supporting nature of the node. The following algorithm shows how to compute a patch $\mathcal{P}$ for a node n_p by encountering the 3D locations of the objects assigned to the node n_p. The parameters of a patch are a normal vector $\vec{n}$ and a distance d. The estimated structures are so-called *level*-1 structures since their child nodes are leaves in the trees. If the child nodes are distinct objects, like "koala", "games1", ... and their relations to the parent node are known, the patch parameters can be computed directly from the object locations. A problem arises in cases where the human tutor has given a category label which refers to a set of objects, e. g., "there are soft toys on the table". Normally, only a subset of soft toys will be on the table. If distinct objects are known to be on the table, the planar patch computed from these objects can be used to resolve the ambiguous labels. This is done by picking those objects from an object set that have the specified category and are located on the computed planar patch. Category labels can be resolved if at least one distinct object per parent node exists. In the following, the necessary steps for estimating a potential planar patch $\mathcal{P}^p_\mathrm{pot}$ for a node n_p are explained in more details. The steps are illustrated by processing the example tree set $\mathcal{T}$ shown in Figure 4.12(12).



(a) detecting the toy car (b) detecting the toy robot

Figure 4.13: Two outputs of an automatic object detection using Sift matching are shown. The toy car and the toy robot are best suited for this kind of object detector.



(a) Manually drawn object boxes with labels. (b) 3D object hulls.

Figure 4.14: This figure shows all objects known to the robot. (a) The object boxes are manually drawn around each object. (b) Considering the 3D points of an object box a 3D convex hull can be computed.

categories \ objects	bear	books	bowl	candle	car	cube	cup	dog	fred	frog	games1	games2	kangaroo	koala	lamp	lion	obelix	pokemon	raven	robot	rose	stitch
soft toys	•						•	•	•			•	•		•	•	•	•			•
toys	•				•	•	•	•	•			•	•		•	•	•	•	•		•
books		•																				
games										•	•											
decoration			•	•			•							•						•		
objects	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 4.2: The list gives for each object the assigned categories of different generality.

(I) $\mathcal{O}$ ($\rightarrow$ Figure 4.14) holds the set of objects known to the robot. This set is divided into a set of *confirmed* objects $\mathcal{O}_{\text{con}}$ and a set of *potential* objects $\mathcal{O}_{\text{pot}}$. $\mathcal{O}_{\text{con}}$ is comprised of the distinct objects in $\mathcal{T}$ and $\mathcal{O}_{\text{pot}}$ holds the remaining objects which are not part of $\mathcal{T}$. It is named potential because it is used for resolving ambiguous labels in $\mathcal{T}$. In our example the two object sets are:

$$\begin{aligned} \mathcal{O}_{\text{con}} &= \{ \text{ lamp, rose, koala, car, lion, robot, books, fred,} \\ &\quad \text{raven, pokemon, candle, dog } \} \quad \text{and} \\ \mathcal{O}_{\text{pot}} &= \{ \text{ bear, bowl, cube, cup, frog, games1, games2,} \\ &\quad \text{kangaroo, obelix, stitch } \}. \end{aligned} \tag{4.5}$$

(II) For each parent node $n_{\text{p}} \in \mathcal{T}$ its potential planar patches $\mathcal{P}^{\text{p}}_{\text{pot}}$ is computed using the distinct objects $\mathcal{O}^{\text{p}}_{\text{con}} \subset \mathcal{O}_{\text{con}}$ assigned as child nodes to n_{p}. In general, a planar patch is described by an orientation and a position in the global coordinate system:

$$\mathcal{P} : \vec{n} \cdot \vec{x} - d = 0. \tag{4.6}$$

Its expansion can be modeled by an ellipse enclosing the locations of the assigned object. There are three cases inducing different patch computations:

1. If the objects are related to their parent in an *on*-relation the potential planar patch $\mathcal{P}^{\text{p}}_{\text{pot}}$ is estimated as a horizontal plane parallel with:

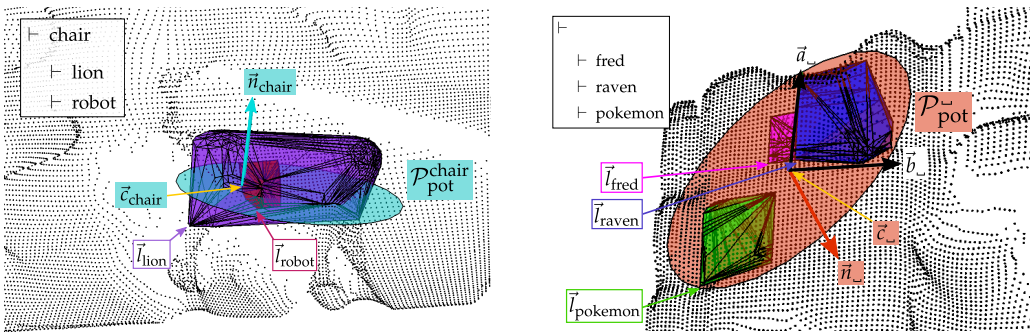
$$\vec{n}_{\text{p}} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad d_{\text{p}} = \vec{n}_{\text{p}} \cdot \vec{c}_{\text{p}}. \tag{4.7}$$

   $\vec{c}_{\text{p}}$ is the ellipse centroid computed from the 3D locations of the confirmed objects. E. g., $\mathcal{P}^{\text{chair}}_{\text{pot}}$ of n_{chair} is computed using the 3D convex hulls of the objects $\mathcal{O}^{\text{chair}}_{\text{con}} = \{\text{lion, robot}\}$. The object's 3D location, here $\vec{l}_{\text{lion}}$ and $\vec{l}_{\text{robot}}$, is the object point with the smallest y-value. Figure 4.15(a) visualizes the computed patch $\mathcal{P}^{\text{chair}}_{\text{pot}}$.

2. If objects are related by an *in*-relation to their supporting structure, a vertical plane models the parent node best. The normal vector $\vec{n}_{\text{p}}$ is a cross-product of the vectors $\vec{a}_{\text{p}}$ and $\vec{b}_{\text{p}}$ spanning the vertical plane:

$$\vec{n}_{\text{p}} = \vec{a}_{\text{p}} \times \vec{b}_{\text{p}}. \tag{4.8}$$

   where $\vec{a}_{\text{p}} = (0, 1, 0)^{\text{T}}$. The vector $\vec{b}_{\text{p}}$ is obtained by estimating the best line through all objects points of $\mathcal{O}^{\text{p}}_{\text{con}}$ projected in the xz-plane (ground plane) via RANSAC [Har03]. The distance d_{p} is determined by applying Equation 4.7 to the centroid $\vec{c}_{\text{p}}$ of all object points. Figure 4.15(b) visualizes the computed vertical patch $\mathcal{P}^{\sqcup}_{\text{pot}}$ using the confirmed objects $\mathcal{O}^{\sqcup}_{\text{con}} = \{\text{fred, raven, pokemon}\}$.



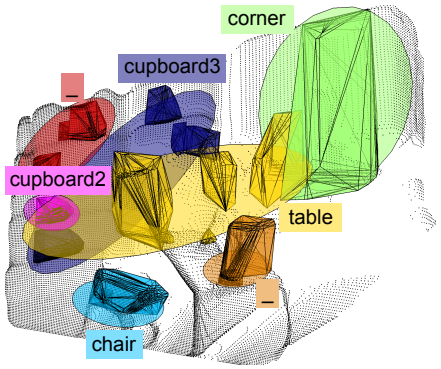
(a) a horizontal plane (b) a vertical plane

Figure 4.15: The figures show two example potential patches. A patch is represented by an ellipse, its normal vector ($\vec{n}_{\text{chair}}$, $\vec{n}_{\text{␣}}$), and its centroid ($\vec{c}_{\text{chair}}$, $\vec{c}_{\text{␣}}$). Each object is represented by its convex hull and its 3D location ($\vec{l}_{\text{lion}}$, $\vec{l}_{\text{robot}}$, $\vec{l}_{\text{raven}}$, $\vec{l}_{\text{fred}}$, $\vec{l}_{\text{pokemon}}$).

3. In the case that the relation between child and parent node is unknown, the 3D arrangement of the confirmed objects determines the orientation of the plane. The orientation of the plane is computed from the object locations and compared to with the orientation of the ground plane. Depending on the result one of the above computations is chosen. An angle smaller than 45° votes for the first computation, otherwise the second computation is chosen. Figure 4.15(b) shows an example where the object positions vote for a vertical supporting plane.

(III) The resulting patch $\mathcal{P}^{\text{p}}_{\text{pot}}$ can be utilized to resolve category labels. Objects in $\mathcal{O}_{\text{pot}}$ having the specified category are tested for their distance to the computed planar patch. If the distance is smaller than a given threshold it is assumed that the human tutor has referred to these objects. Therefore, they can be assigned to the potential patch. Finally, all object points projected on the patch plane are used to recompute the expansion of the patch ellipse by determining its two principal axis using Principal Component Analysis (PCA).

Figure 4.16 shows the computed initial scene model. It consists of patches for "corner", "table", "chair", "cupboard2", and "cupboard3", and two patches with an empty label "␣". The category label "softtoy on the table" is resolved to "kangaroo", "bear", "frog", and "stitch". The category label "games" in "cupboard3" is resolved to "games1" and "games2" (highlighted in red in Figure 4.16). Due to wrong object assignments and underspecified descriptions the initial scene model contains erroneous and fictive potential patches. Finding solutions for these problems is the scope of the next section.



- corner
  - lamp
- table
  - rose
  - kangaroo
  - bear
  - frog
  - stitch
-
  - koala
  - car
- chair
  - lion
  - robot
- cupboard2
  - books
-
  - fred
  - raven
  - pokemon
- cupboard3
  - candle
  - dog
  - games1
  - games2

Figure 4.16: Set of potential planar patches $\{\mathcal{P}^p_{\text{pot}}\}_{p=1...7}$ computed for the tree set $\mathcal{T}$ of subject 3(p) using the proposed algorithm are visualized as colored ellipses. In the tree set on the right the resolving of the ambiguous labels "softtoy" and "games" is highlighted in red.

4.4.3 *Adapting the Initial Scene Structures to the Visual Perception*

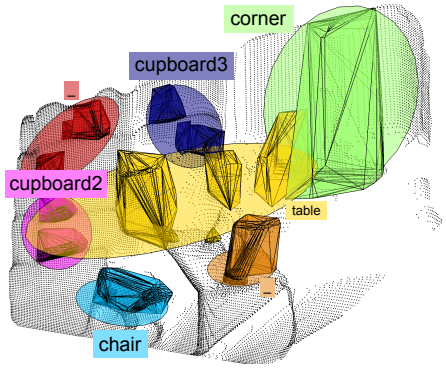
Since the potential planar patches are derived without knowledge about real planar patches in the scene, the initial scene model may have two main problems as can be seen in Figure 4.16. First, errors in resolving ambiguous category labels lead to wrong assignments of objects to parent nodes. For example, "games" as child of "cupboard3" is resolved to "games1" and "games2" even though "games1" is in "cupboard2". This happens because the expansion of the potential patch is not considered when resolving category labels. In cases where other supporting structures like the "cupboard2" lie in the same infinite plane and contain objects of matching category the risk of mis-assigning is quite high resulting in erroneous potential patches like "cupboard3" has. Second, verbal descriptions are often underspecified which means that references to supporting structures are only given implicitly, e. g., through sentence construction, resulting in parent nodes with empty labels. For example, the table in Figure 4.16 consist of two potential patches, $\mathcal{P}^{\text{table}}_{\text{pot}}$ and a virtual patch $\mathcal{P}^{\sqcup}_{\text{pot}}$, modeling the left and the right part of the table. This is due to the fact that the verbal description has not provided relations between objects of the two patches like, e. g., "the koala is right of the frog" or relations naming the supporting structure like "the koala on the table".

Both problems will be addressed by considering real 3D planar surfaces which are extracted by the region growing algorithm presented in Section 2.4.2. Figure 4.5(a) shows such extracted patches $\{\mathcal{P}^i_{\text{real}}\}_{i=1...m}$ in 3D data sampled from the playroom scene $\mathcal{S}_1$. A real patch $\mathcal{P}^i_{\text{real}}$ can be mapped to a potential patch $\mathcal{P}^{\text{p}}_{\text{pot}}$ if the angle between the normal of the real patch and the normal of the potential patch is smaller than an angle threshold. Further, the two patches has to be close together. This is true if there exists at least one point in the real patch which distance to the centroid of the potential patch is smaller than a given distance threshold. Several real patches can be assigned to one potential patch and one real patch can be assigned to multiple potential patches.

CORRECTING WRONG OBJECT ASSIGNMENTS. If a real patch $\mathcal{P}^i_{\text{real}}$ is assigned to different potential patches with different labels, e. g., "p1" and "p2" (not considering the empty label "␣"), this means that some of the objects are mismatched. The goal is to find an injective mapping from real patches to potential patches. This means that a real patch should only be assigned to one potential patch or to a set of potential patches where at most one patch has a label while the remaining ones must have empty labels. If a real patch is mapped to two potential patches with competing labels this indicates that an object is falsely assigned. This can be identified by checking for all objects of the potential patch $\mathcal{P}^{\text{p1}}_{\text{pot}}$ and patch $\mathcal{P}^{\text{p2}}_{\text{pot}}$ whether they are positioned in/on the real patch $\mathcal{P}^i_{\text{real}}$. $\mathcal{P}^i_{\text{real}}$ will be put to that potential patch holding the biggest percentage of objects lying in/on $\mathcal{P}^i_{\text{real}}$. All objects of the other patch that also lie in $\mathcal{P}^i_{\text{real}}$ are moved to this potential patch, too. The ellipses representing the potential patches are updated considering the new added or removed objects. Figure 4.17 shows the correction of the initial model given in Figure 4.16. Initially, "games1" has been assigned wrongly to "cupboard3", after applying the described procedure it is moved to "cupboard2".

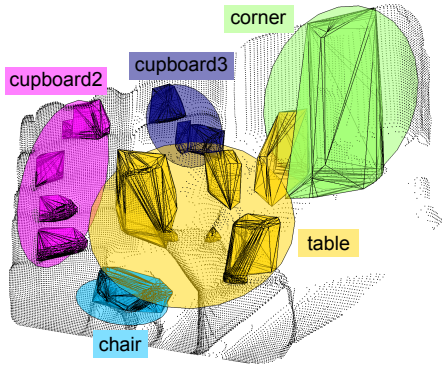
INFERRING LABELS FOR VIRTUAL PATCHES. After correcting mismatched objects and recomputing potential patches the bottom-up extracted planar patches can be used to infer labels for parent nodes currently assigned an empty label, here called *virtual patches*. Such parent nodes arise in cases where the descriptor has related objects by parallel relations to each other. A label for the common supporting structure has not be given explicitly or it has not been possible to conclude the label after processing the whole description. Inferring a label for a supporting structure can be done for cases where a bottom-up patch $\mathcal{P}^i_{\text{real}}$ is assigned to a set of potential patches where exactly one patch in the set has a label "p" differing from the empty label "␣". In this case, I propose to merge all parent nodes to one node labeled with the non-empty label "p". All child nodes are assigned to the new parent node using the standard tree fusion technique shown in Figure 4.11(b). The orthogonal relation between the objects and the structure "p" is assumed from the fact that all objects lie within the same real planar patch. Utilizing all objects assigned, the ellipse representing the new node "p" can be recomputed. Figure 4.18 shows the two fused patches in our example model. The parent node of "koala" and "car" is merged with the "table" node which means that "koala" and "car" are located on the table. "fred", "raven", and "pokemon" are assigned in the same way to "cupboard2".

Finally, Figure 4.19 shows the resulting *aligned scene model* for the example description. It encounters the situation model of the communication partner and the perceptual reality of the scene by computing initial patches from given verbal descriptions and adapting them to fit the bottom-up extracted planar patches. The aligned model gives a set of patches which represent meaningful structures in the scene. Their labels are equal to those used by the interlocutor. Such models can support a smooth Human-Robot-Communication about the surrounding environment itself and about task instructions requiring knowledge of given spatial conditions.



- corner
  - lamp
- table
  - rose
  - kangaroo
  - bear
  - frog
  - stitch
-
  - koala
  - car
- chair
  - lion
  - robot
- cupboard2
  - books
  - games1
-
  - fred
  - raven
  - pokemon
- cupboard3
  - candle
  - dog
  - games2

Figure 4.17: The wrong assigned objects of the initial model are corrected. Here, "games1" is moved to "cupboard2". This is highlighted also in the tree set.



- corner
  - lamp
- table
  - rose
  - kangaroo
  - bear
  - frog
  - stitch
  - koala
  - car
- chair
  - lion
  - robot
- cupboard2
  - books
  - games1
  - fred
  - raven
  - pokemon
- cupboard3
  - candle
  - dog
  - games2

Figure 4.18: This figure shows the scene model after fusing virtual patches with named potential patches. Patches are fused if they share the same bottom-up patch. In this example, "koala" and "car" are added as children to the "table" and "fred", "raven", "pokemon" to the "cupboard2".

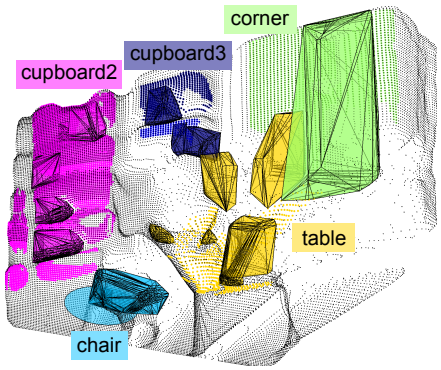


Figure 4.19: This figure shows the matching of an aligned scene model on a set of bottom-up extracted planar patches $\{\mathcal{P}^i_{\text{real}}\}_{i=1\ldots m}$. The result is a subset of patches enhanced semantically with meaningful names.

4.5 EVALUATION

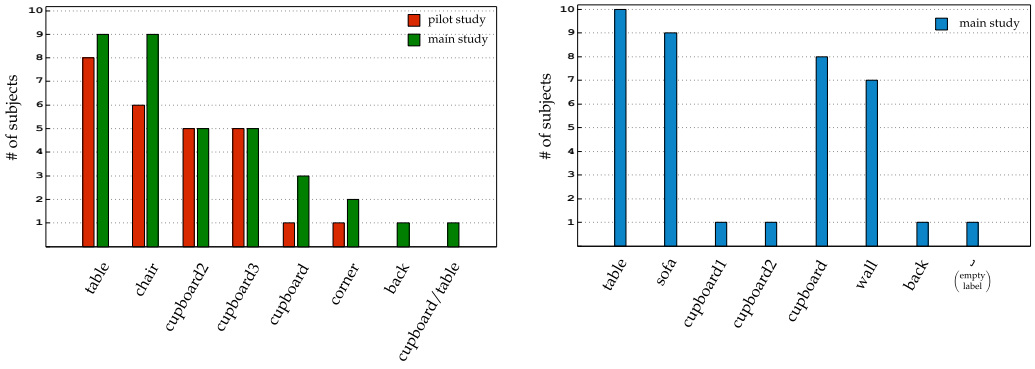
This section is going to evaluate the meaningful structures provided by the aligned scene model approach. First, the model generated for subject 3(p) is analyzed in more details (→ Section 4.5.1). Further, models generated from a large amount of descriptions are evaluated quantitatively (→ Section 4.5.2, → Section 4.5.3). Overall, 30 descriptions are processed. 20 descriptions deal with the playroom scene $\mathcal{S}_1$ (→ Figure 4.5(a)) – 10 acquired in the pilot study and 10 in the main study – and 10 descriptions deal with the living room scene $\mathcal{S}_2$ (→ Figure 4.5(b)) acquired in the main study. In the pilot study, a picture of the scene has been presented as print-out and the participants have been instructed to describe what they "see in the picture". In the main study, the participants have been shown the scene on a computer screen and have been asked to describe what they "see in the room". All original German descriptions and their translation into English and a machine-readable representation can be looked up in Appendix B. Last, the influence of object detection errors on the models is examined (→ Section 4.5.4).

4.5.1 *Analysis of an Example Model*

In this section, I have chosen the description of subject 3(p) for a qualitative analysis of the derived aligned scene model. This description has been used throughout the whole Section 4.4 to visualize results of the intermediate computation steps. Figure 4.19 shows the meaningful spatial structures generated from the example description. The highlighted structures meet the expected ground truth, as meaningful structural elements, which are "cupboard2", "cupboard3", "corner", "table", and "chair", are chosen and the correct labels are provided. In most cases one potential patch is mapped on one real patch. The "cupboard3" is an exception because it consists of two bottom-up patches (colored in two different blues). No bottom-up patch is found for the "chair" because currently the chair is hidden by the objects on top of it. However, a mapping will be possible in subsequent data where the objects are removed.

4.5.2 *Analysis of Level-1 Structures*

Figure 4.21, Figure 4.23, and Figure 4.25 show all generated aligned scene models. The room structures represented by ellipses are localized on level-1 in the tree sets $\mathcal{T}$ since objects positioned in the leafs of the trees are used to estimate these structures. The histograms in Figure 4.20 show the distribution of the structure labels found in the models. A structure appearing in many models is a prominent element of the scenario. In general, no differences in the descriptions of the pilot study (subjects 1(p) – 10(p)) and the main study (subjects 1(m) – 10(m)) can be observed. Each model is aligned to the specific level of detail given in the description. Most of the descriptions are quite detailed with many relations



(a) distribution of structure labels derived in the playroom scene $\mathcal{S}_1$

(b) distribution of structure labels derived in the living room scene $\mathcal{S}_2$

Figure 4.20: This figure shows the distribution of the concluded structure labels in the models generated by applying our algorithm to all descriptions acquired in our studies (pilot study: playroom (10 participants); main study: playroom, living room (10 participants) → 30 descriptions). The bar shows the number of models in which the corresponding structure label can be found.

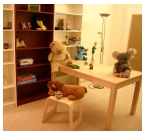
between objects and room structures like the model shown in Figure 4.21(c). There are only few descriptions that are just simple object listings (see 1(p), 2(p), 5(p), and 7(p) in Figure 4.22) so that nearly no spatial structures can be concluded like show in Figure 4.21(a), 4.21(b), 4.21(e), and 4.21(g). The trivial case that every object listed has its own small supporting patch is not displayed because this does not provide any higher-level semantic information of the room.

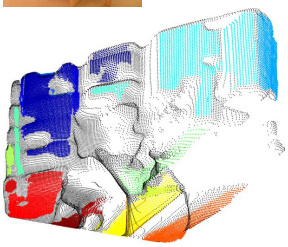
In the playroom scenario (→ Figure 4.20(a)) the "table" is the most dominant structure as 8 resp. 9 descriptions provided the table. The second prominent structure is the "chair" because it is mentioned by 6 resp. 9 subjects. The potential patch for the table is generated in 17 cases with a reliable position and orientation. Only in the model of subject 1(p) (→ Figure 4.21(a)) no patch has been computed because it has been mentioned by the descriptor without relations to informative objects. It is only known that there are soft toys on the table. As no distinct object is available, the "table" patch cannot be estimated and the category label "soft toys" cannot be resolved. Subject 8(m) (→ Figure 4.23(h)) has fused the arrangement of table and shelves to one structure, here labeled as "cupboard/table". The structure has not been divided into smaller parts by locating objects explicitly to the subparts. Instead, the subject has simply said: "the mentioned objects are spread all over table and cupboard". Except from the "table" and the "chair", the shelves at the wall are also interesting structures. Half of the participants have given enough information to conclude patches for "cupboard2" (5 resp. 5) and "cupboard3" (5 resp. 5). Only 1 resp. 3 participants fused the cupboards to one construct named "cupboard". The reason for this observation may be the different colors of the shelves which led the describer to refer to individual shelves. This argument is even supported by the contrary observation in the models of the living room scene $\mathcal{S}_2$. In most cases (8 participants) all shelves are fused to the structure "cupboard" (→ Figure 4.20(b)). Here, the shelves have the same color and are therefore harder to perceive as separated structures. The other meaningful structures in the living room are "table" (10),

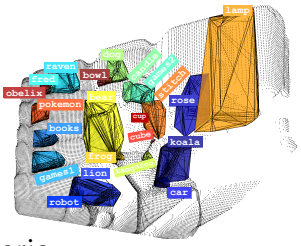
"sofa" (9), and "wall" (7). In general, the system fails to compute a potential patch for a spatial structures (indicated by "Cannot compute potential patch for ...", e. g., in Figure 4.21(d)) when only the category of objects located in resp. on the structure is known (like "there are soft toys in the cupboard"). Without knowing at least one specific object this category label cannot be resolved. Further, a potential patch cannot be computed if the sole specific object is not known to the robot like the "carpet" in the description of subject 10(m) given as tree set in Figure 4.26(j). If other objects are available for the supporting structure, this unknown object will be simply ignored.

Some interesting artifacts can be observed in the models of the living room . The model of subject 6(m) (→ Figure 4.25(f)) contains for "lamp" and "picture1" a common structure with an empty label. Considering the gravity of objects and their fixation in the scene it can be seen that "lamp" and "picture1" do not share a common supporting structure even though the describer has related them. A reason could be that he/she has taken into account the objects' 2D arrangement in the picture instead of their 3D arrangement in the real scene. In future work, the robot would indent to get in a subsequent dialog names for meaningful structures with an empty label "␣". For example, it could ask "what is the supporting structure of lamp and picture1". In cases where objects have a real common structure the human will be able to give a label. Otherwise, he/she would maybe say: "the lamp and the picture1 do not share a common structure; the picture1 is attached to the wall and the lamp stands on the floor". From this, the robot should conclude that the parent node in the tree relates both objects incidentally and has to be removed. Another artifact can be observed in the description of subject 7(m) (→ Figure 4.25(g)). There, my algorithm ignores several relations because they have provided competing labels for a parent structure. The "sofa" is localized in the "room", with the lamp behind the "sofa", and the "lamp" at the "wall". From the latter, the inference mechanism would conclude that the sofa is also at the "wall", hence, the "wall" is the supporting structure for the "sofa". But the "sofa" is already assigned to the "room". The reason for the competing labels is maybe the fact that the "at"-relation between "lamp" and "wall" is in this case a parallel and not an orthogonal relation.

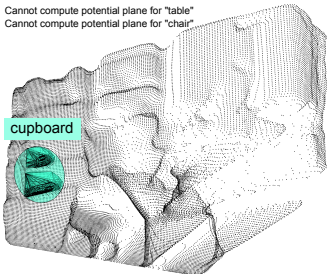
In a nutshell, it can be said that all generated aligned scene models are an appropriate representation of the respective scenario. If tables are part of the scenario they seem to be quite prominent parts of the scene as they are in most cases communicated to the robot. The strength of my modeling approach is that room structures like the table are reliably modeled across the scenes even though the table of the playroom differs significantly from the table in the living room. It would be challenging to design a table model in advance that could detect both the big table in the playroom and the small table in the living room. As furniture has a large amount of degrees of freedom it is quite complex to consider all parameters and parameter combinations. Further, parts of the room frame itself like walls or corners of rooms are important if objects are attached to them (like, the wall in the living room) but could be ignored if the tutor does not refer to them. This results in a resources preserving and aligned representation best suited for modeling spatial awareness.



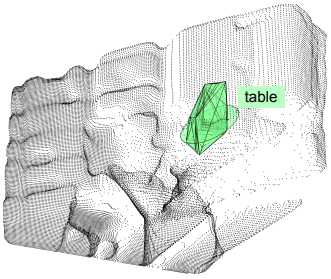




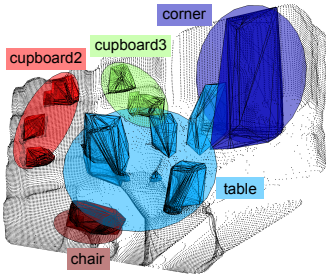
scenario



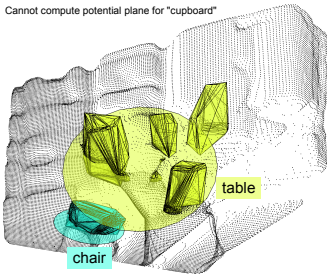
(a) subject 1(p)



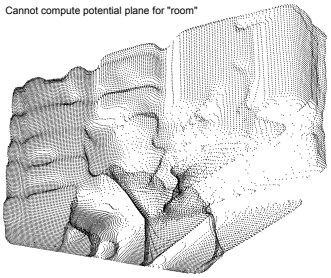
(b) subject 2(p)



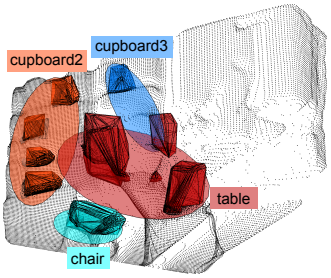
(c) subject 3(p)



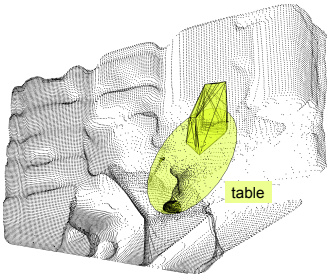
(d) subject 4(p)



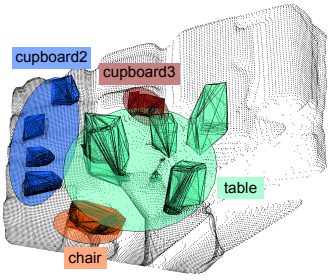
(e) subject 5(p)



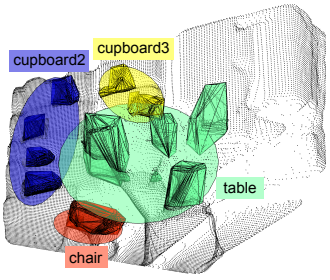
(f) subject 6(p)



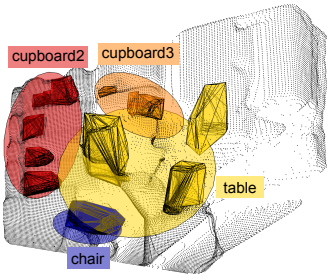
(g) subject 7(p)



(h) subject 8(p)

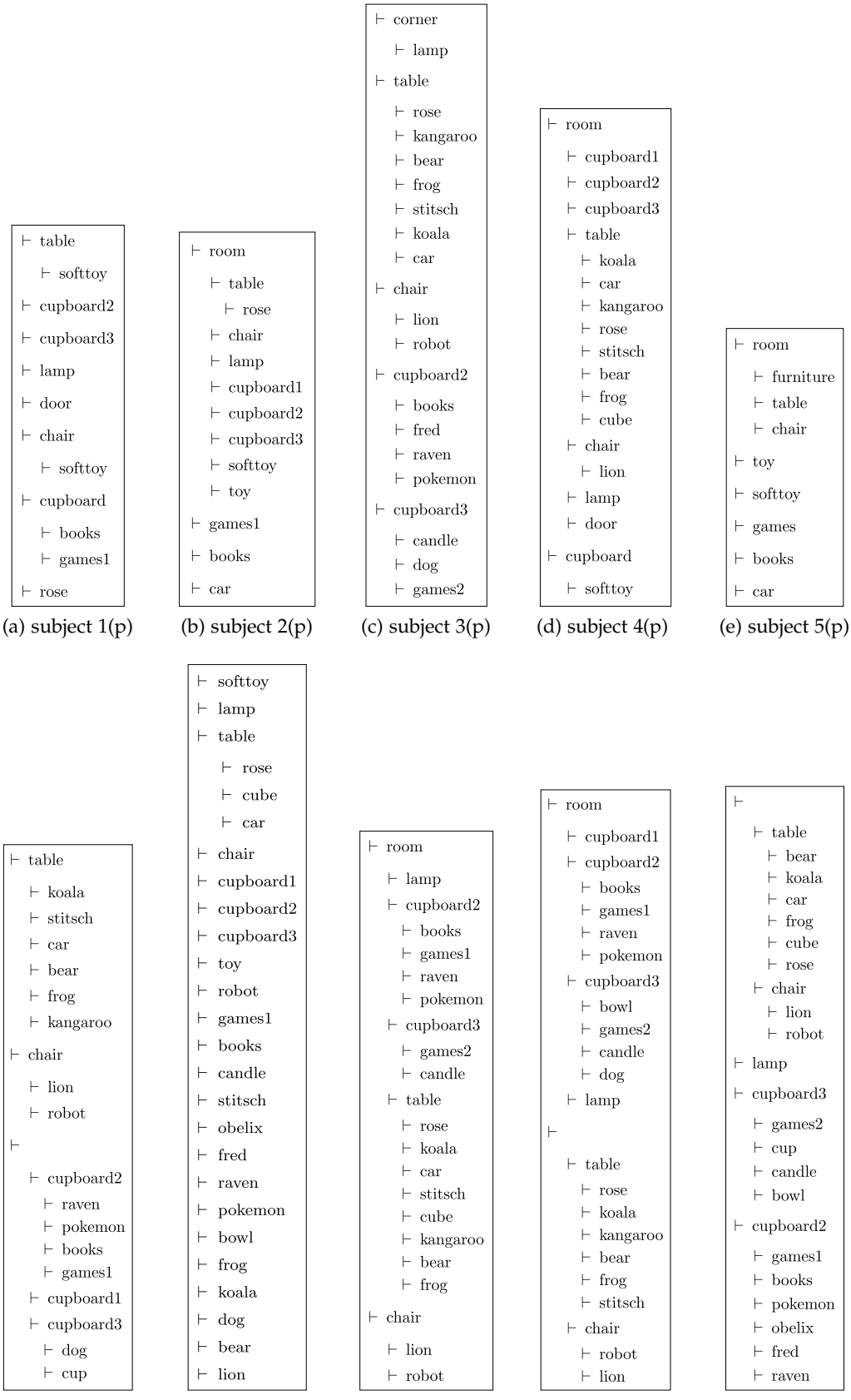


(i) subject 9(p)



(j) subject 10(p)

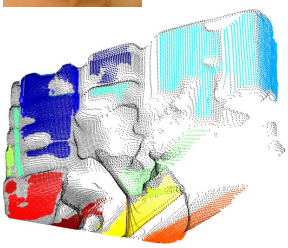
Figure 4.21: Aligned scene models of descriptions about the playroom scene given during the pilot study ($\rightarrow$ (p)). The corresponding descriptions transformed to tree sets are given in Figure 4.22. It can be seen that the subjects 1(p), 2(p), 5(p), and 7(p) just listed objects without relating them to each other. Therefore, it has not been possible to estimate supporting structures. The trivial case that every mentioned object has its own small supporting patch is not displayed because this does not provide any semantic information about the room.



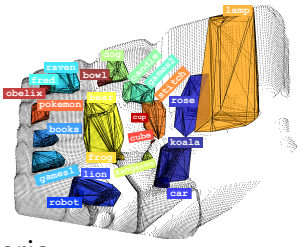
(a) subject 1(p) (b) subject 2(p) (c) subject 3(p) (d) subject 4(p) (e) subject 5(p)

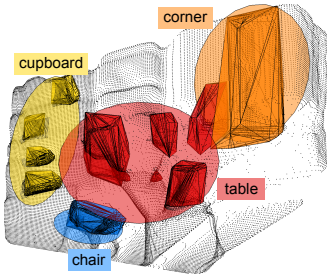
(f) subject 6(p) (g) subject 7(p) (h) subject 8(p) (i) subject 9(p) (j) subject 10(p)

Figure 4.22: Corresponding tree sets of playroom scene models generated from descriptions given during the pilot study (→ (p)). The corresponding aligned scene models are displayed in Figure 4.21.

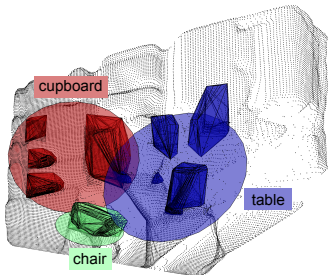


scenario

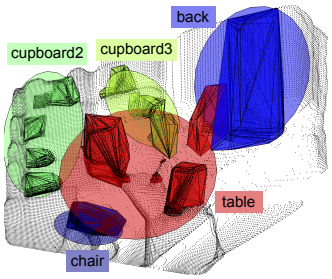




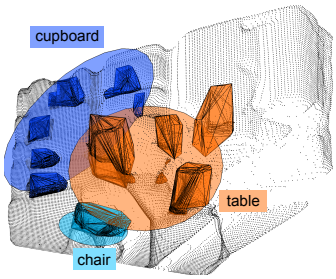
(a) subject 1(m)



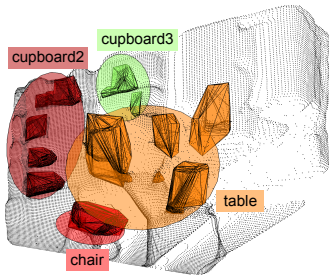
(b) subject 2(m)



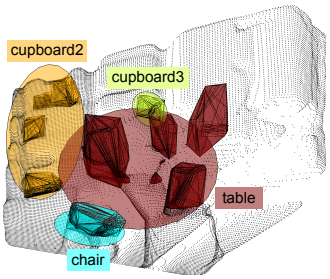
(c) subject 3(m)



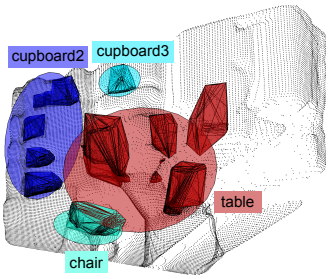
(d) subject 4(m)



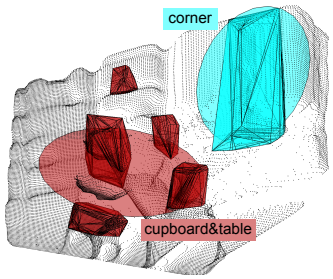
(e) subject 5(m)



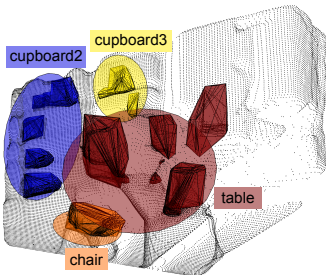
(f) subject 6(m)



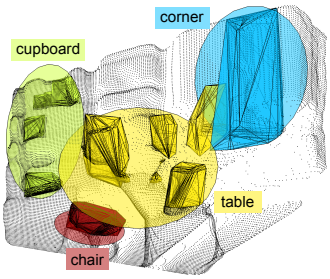
(g) subject 7(m)



(h) subject 8(m)

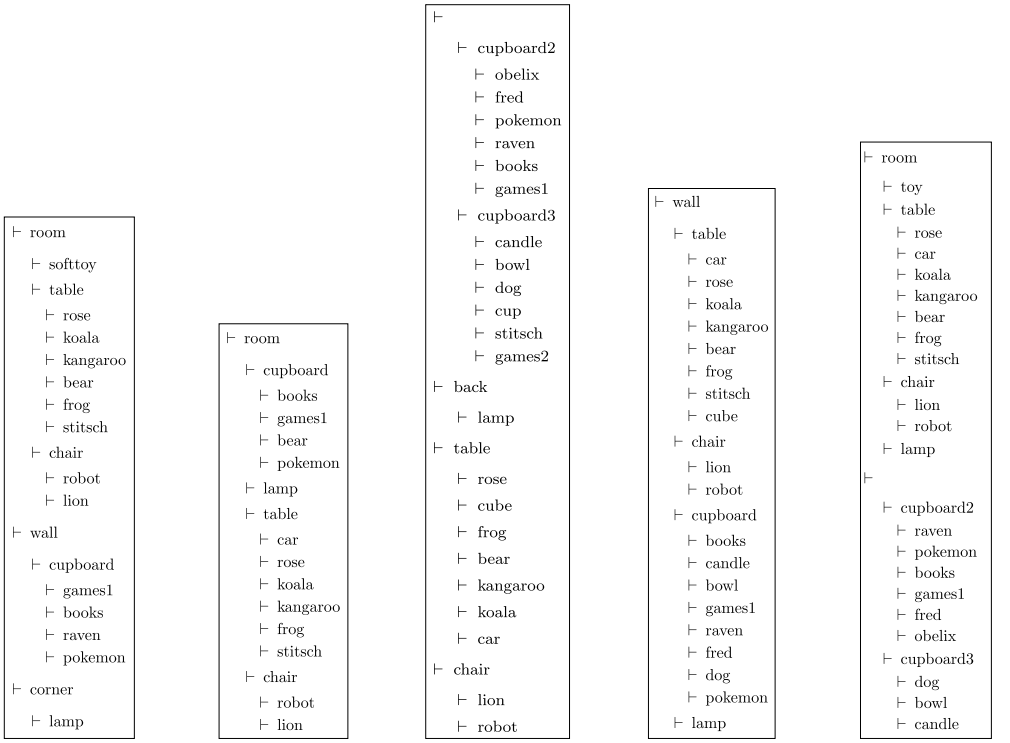


(i) subject 9(m)



(j) subject 10(m)

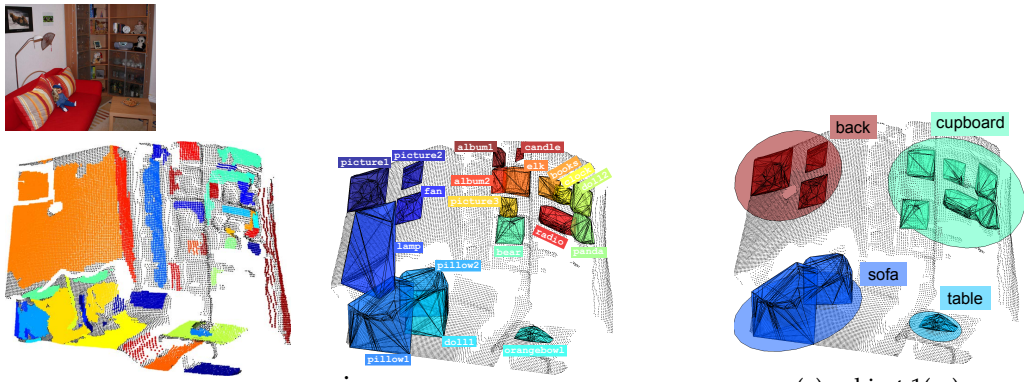
Figure 4.23: Aligned scene models of descriptions about the playroom scene given during the main study (→ (m)). The corresponding tree sets are given in Figure 4.24.



(a) subject 1(m) (b) subject 2(m) (c) subject 3(m) (d) subject 4(m) (e) subject 5(m)

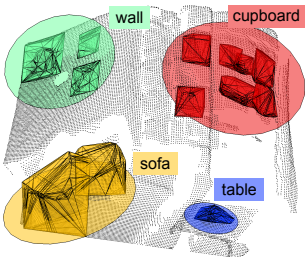
(f) subject 6(m) (g) subject 7(m) (h) subject 8(m) (i) subject 9(m) (j) subject 10(m)

Figure 4.24: Corresponding tree sets of playroom scene models generated from descriptions acquired during the main study (→ (m)). The computed aligned scene models are displayed in Figure 4.23.

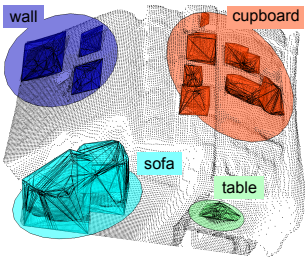


scenario

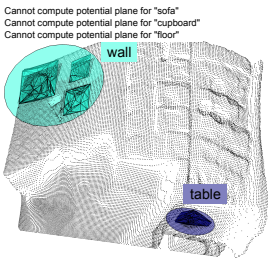
(a) subject 1(m)



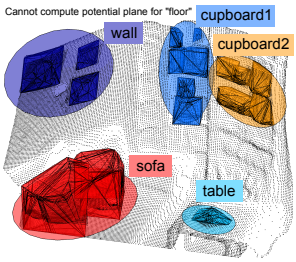
(b) subject 2(m)



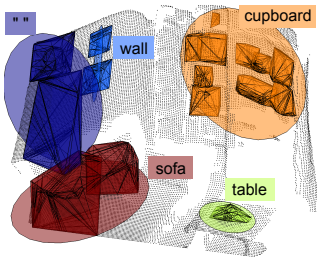
(c) subject 3(m)



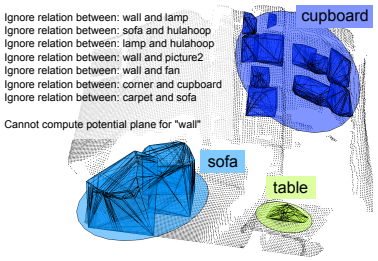
(d) subject 4(m)



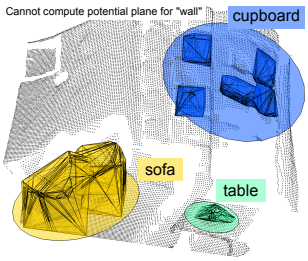
(e) subject 5(m)



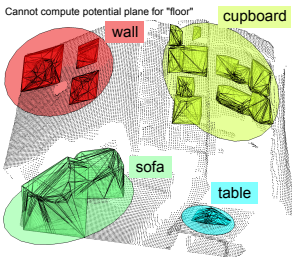
(f) subject 6(m)



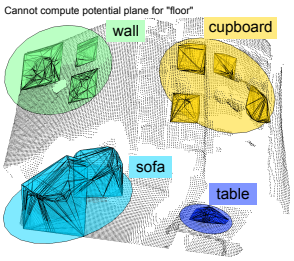
(g) subject 7(m)



(h) subject 8(m)



(i) subject 9(m)



(j) subject 10(m)

Figure 4.25: Aligned scene models of descriptions about the living room scene given during the main study (→ (m)). The tree sets can be looked up in Figure 4.26.

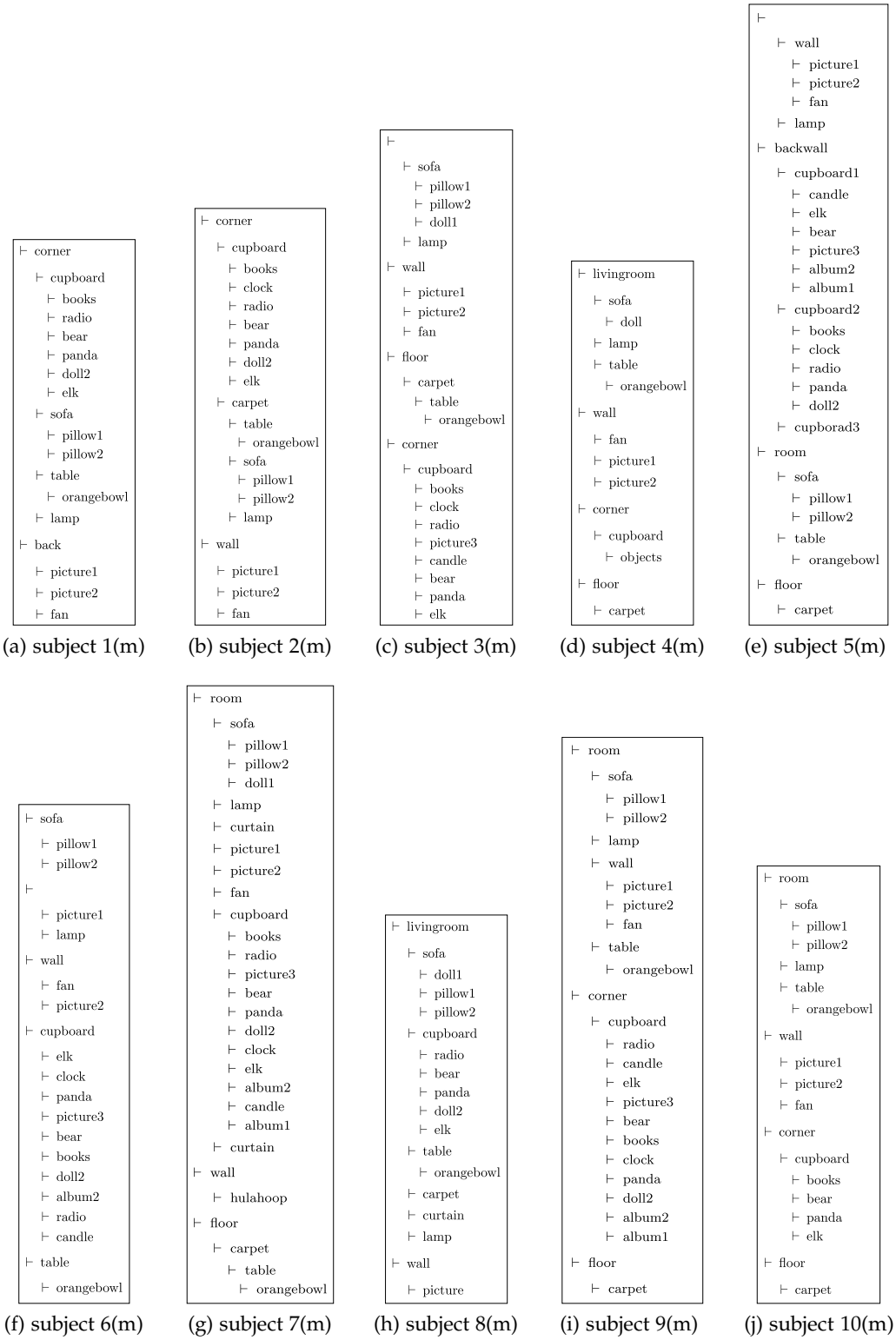


Figure 4.26: Corresponding tree sets of living room scene models generated from descriptions acquired during the main study (→ (m)). The corresponding aligned scene models are shown in Figure 4.25.

4.5.3 *Analysis of Level-2 Structures*

Besides the combination of small objects along their physically supporting structures the level-1 structures are also sometimes related to each other through a common parent node in the tree set. I refer to this kind of structures as level-2 structures. In the two scenarios the most prominent groupings of furniture structures are:

- playroom
  - "table"–"chair",
  - "cupboard2"–"cupboard3",
  - "table"–"chair"–"cupboard2"–"cupboard3",
- living room
  - "table"–"sofa',
  - "cupboard",
  - "table"–"sofa"–"cupboard".

The separated grouping of "table"–"chair" and "cupboard2"–"cupboard3" often corresponds to grouping of furniture relatively to the observer. E. g., the parent node of "table" and "chair" is named "front" by subject 9(m): "in the front of the picture is a table and ahead of the table stands a chair" (→ Figure 4.24(i)). In absolute numbers, "table" and "chair" are related in 5 out of 20 tree sets. A comparable relation is given between "cupboard2" and "cupboard3". The parent structure for these two pieces of furniture is either "wall" in the sense of "cupboard2 and cupboard3 stand at the wall" (→ Figure 4.24(g)) or "back" as in "in the back of the room are cupboard2 and cupboard3" (→ Figure 4.24(f)). In 9 of 20 models "cupboard2" and "cupboard3" have a common parent structure. In 6 models all furniture elements, here "cupboard2", "cupboard3", "table", and "chair" are related to each other by localizing them in the "room" (see, e. g., Figure 4.22(d)). The grouping of cupboards standing at a wall can also be found in 5 of 10 living room models. Here, the cupboards are perceived and represented as one level-1 structure, "cupboard", which itself is localized relatively to the "back wall (→ Figure 4.26(e)) or to the "corner" (→ Figure 4.26(i)). In the same models also "table" and "sofa" are combined. Their parent node is in most cases labeled with "room". This corresponds with the result that this two pieces of furniture are the most prominent structures in the living room. In the living room scenario some participants, e. g., subject 1(m) and subject 8(m), even have clustered together all pieces of furniture, which are the "cupboard", the "table", and the "sofa". But contrary to the playroom scenario, where the walls are only level-2 structures, the "wall" in the living room scenario is a level-1 structure as detectable objects ("picture1", "picture2", "fan") are attached to the wall. If a "wall" is used as level-1 supporting structure a difference in usage is visible when compared to normal furniture. The available models show that a "wall" is less often related to other supporting structures than furniture like "tables", "sofas", etc. The reason for that may be the fact that the only common supporting structure would be the room itself. The information that "a wall is in the room" is seldom given explicitly since this knowledge can be assumed to be known as common knowledge about rooms in general.

4.5.4 *Influence of Object Detection Errors on Model Formation*

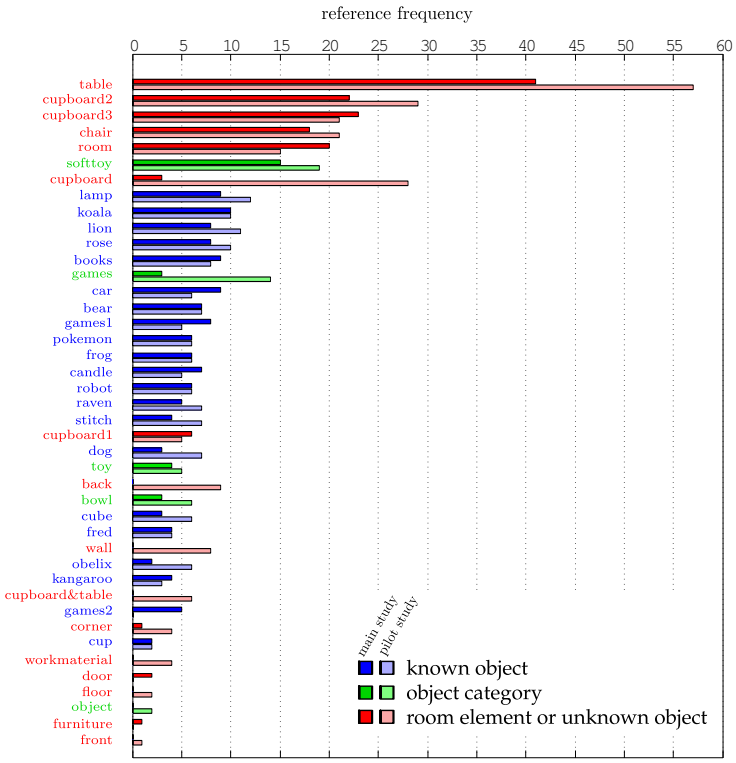
In general, scene descriptions consist of references to objects, categories, and room structures and relations between them. An analysis of the reference frequency of each object can reveal further insights on the importance of particular objects. For each description the number of references to an object is counted. The counts are summed over all descriptions belonging to one experiment and room type. Figure 4.27 shows the counts for each object. The prominent supporting structures, "table", "cupboard2", "cupboard3", "chair" and "cupboard", "sofa", "wall", occur most frequently which supports the importance of these spatial structures. In the playroom scenario big objects like the "koala" on the table or the "lion" on the chair are among the most referenced objects. In the living room scenario objects at the wall (like "picture1", "fan", and "picture2") are mentioned most often followed by objects on the sofa ("pillow1" and "pillow2"), in the cupboard ("radio"), and on the table ("orange bowl"). If detection fails on these objects, problems in estimating the super-ordered structures can arise so that potential patches cannot be computed. The problems can be compensated in cases where other correctly detected objects are located on the same structure.

As object detection is the basic input for estimating level-1 structures the influence of detection errors on the model formation process has to be examined. For this purpose, errors are introduced into the set of perfect (because hand-labeled) object detections presented in Figure 4.14. Errors in the recognition of objects influence the scale and the position of the object bounding boxes. The following errors are introduced to randomly selected objects of the hand-labeled object set:

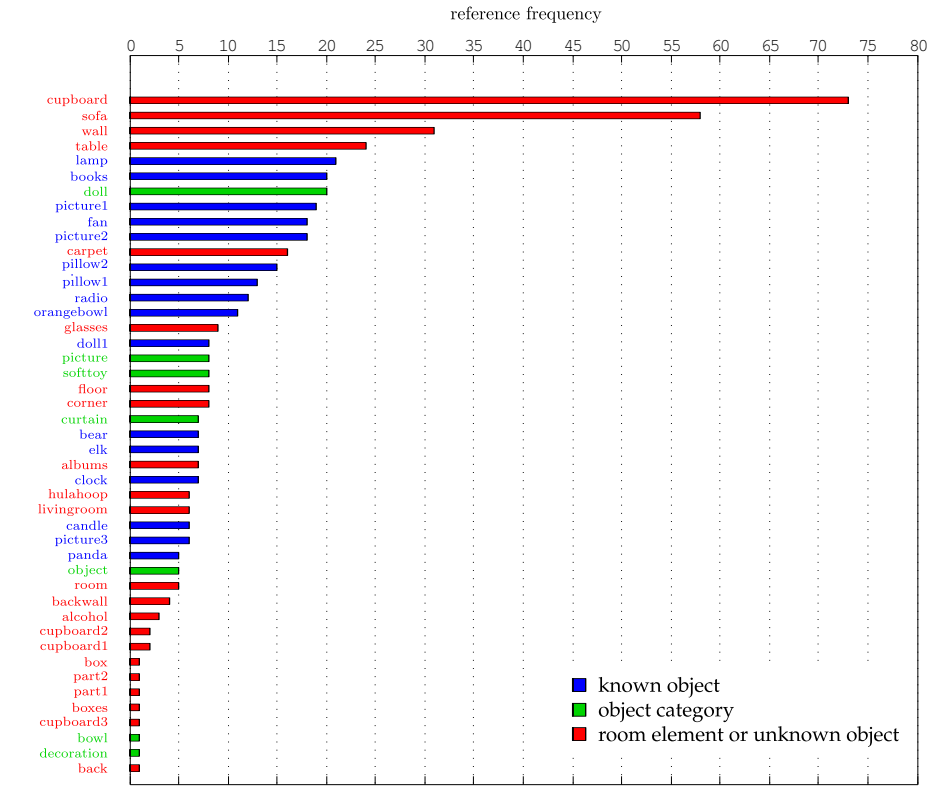
- *Translation errors influence the position of the object bounding box*
  $\Rightarrow$ *no overlap with ground truth from*
  - failing completely to detect an object
    $\rightarrow$ applied to "fred" and "stitch"
  - hallucinating objects at an image position where no object is present (this applies also to false positives)
    $\rightarrow$ applied to "cube" and "robot" by putting their bounding box at a random position in the image
  - recognition errors resulting in a wrong labeling of detected objects
    $\rightarrow$ applied to "candle" and "dog" and to "games1" and "bowl" by swapping their bounding boxes

  $\Rightarrow$ *overlap with ground truth from*
  - small inaccuracies in determining the position of the object box
    $\rightarrow$ applied to "games2" and "frog" by moving their box by a small randomly chosen displacement in x and y direction

- *Scale errors influence the expansion of the object bounding box*
  - the box size is smaller than the ground truth box
    $\rightarrow$ applied to "pokemon" and "books" through shrinking their boxes by a random factor
  - the box size is larger than the ground truth box
    $\rightarrow$ applied to "rose" and "bear" through enlarging their boxes by a random factor



(a) playroom



(b) living room

Figure 4.27: The bars plot the reference frequencies of all objects, category labels, and room elements. The bars are achieved by counting their use. Rooms and experiments are treated separately. As objects can be referenced several times per description each appearance is counted as one reference resulting in, e. g., 5 references to the same object in one description.

deviation	corner	table	chair	cupboard2	cupboard3
conormal	6.01°	0°	0°	0.26°	14.53°
coplanar	0mm	12mm	302mm	22mm	10mm

Table 4.3: This table lists the conormal and coplanar deviation of the potential patches in the aligned scene model shown in Figure 4.29 from the patches in the model shown Figure 4.18. The model in Figure 4.29 is computed using the erroneous set of objects presented in Figure 4.28 while the model in Figure 4.18 is based on the set of correctly detected objects show in Figure 4.14.

In the resulting object set 14 of 22 object boxes are erroneous (respectively, 12 of 20 as two object boxes are missing completely). Figure 4.28 shows the modified set of objects. Figure 4.29 shows the resulting aligned scene model computed for the description of subject 3(p) using the erroneous object set. As the estimated potential patches are represented by a normal vector and a barycenter, these patches can be compared to the ground truth patches shown in Figure 4.18 by computing between the according patches the conormality (→ Equation 2.15) and coplanarity (→ Equation 2.17) measurement. The correctness of a patch can be judged reliably with the coplanarity value as a translation of the patch within the plane is less penalized than the same translation out of the plane. Table 4.3 lists for the supporting structures their conormal and coplanar deviations.

The patches for "cupboard2", "table", and "corner" show only minor deviations from the original patches. The twisting of "cupboard3"-patch by 14.53° is acceptable since the overall position of the patch is correct. It is still possible to assign the correct real patch to "cupboard3" because I allow due to noise a deviation of up to 30° between the potential patch normal and the real patch normal. Only the "chair"-patch is too big and misplaced. If 3D data of the chair itself could be perceived my algorithm presented in Section 4.4.3 would detect that the "robot"-object is misplaced with respect to the chair. The algorithm is designed to handle mismatched objects that occur when category labels are resolved (see Figure 4.17 for an example). As the computation of the 3D object hulls incorporates a removing of outlier points it can handle slightly misplaced object boxes. Further, the orientation of the horizontal patches is only influenced by the orientation of the camera and the RANSAC based approach for computing the vertical patches can deal with remaining outliers. Also, missing objects can be compensated if enough other objects are known for the corresponding parent node. Only in cases where no further object is known, no potential patch can be estimated. The difference between swapping bounding boxes and large box displacements is that for the swapping case the new position of an object is still in or on a supporting scene structure. While for the large displacement this cannot be assured as the object box is positioned arbitrary in the scene. In the best case, a box swapping does not influence the model if it happens between objects localized in/on the same structure. In the worst case, the box swapping has to be handled in the same way like the arbitrary displacement which means that wrong positions has to be detected if bottom-up patches are available. Only if misplaced objects cannot be detected, the model is corrupted. The corruption mostly affects the expansion of a potential patch and its position. The orientation of patches is less influenced because it is computed robustly.

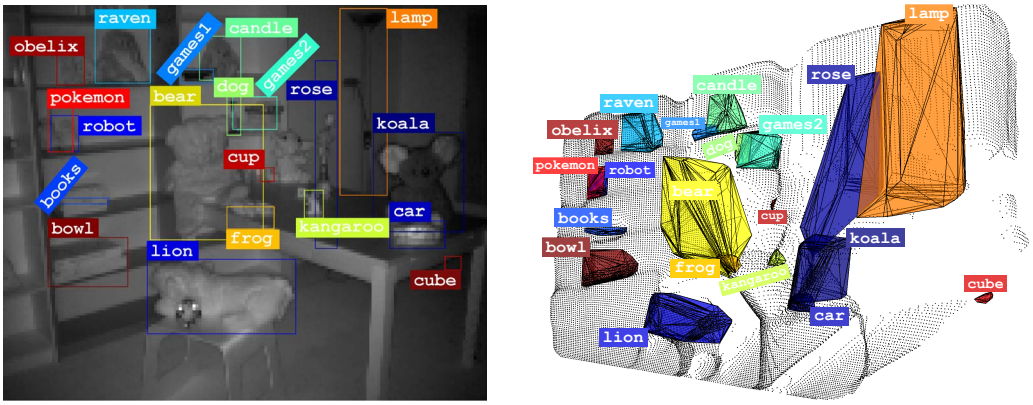


Figure 4.28: (left) erroneous 2D bounding boxes of objects known to the robot, (right) corresponding 3D object hulls.

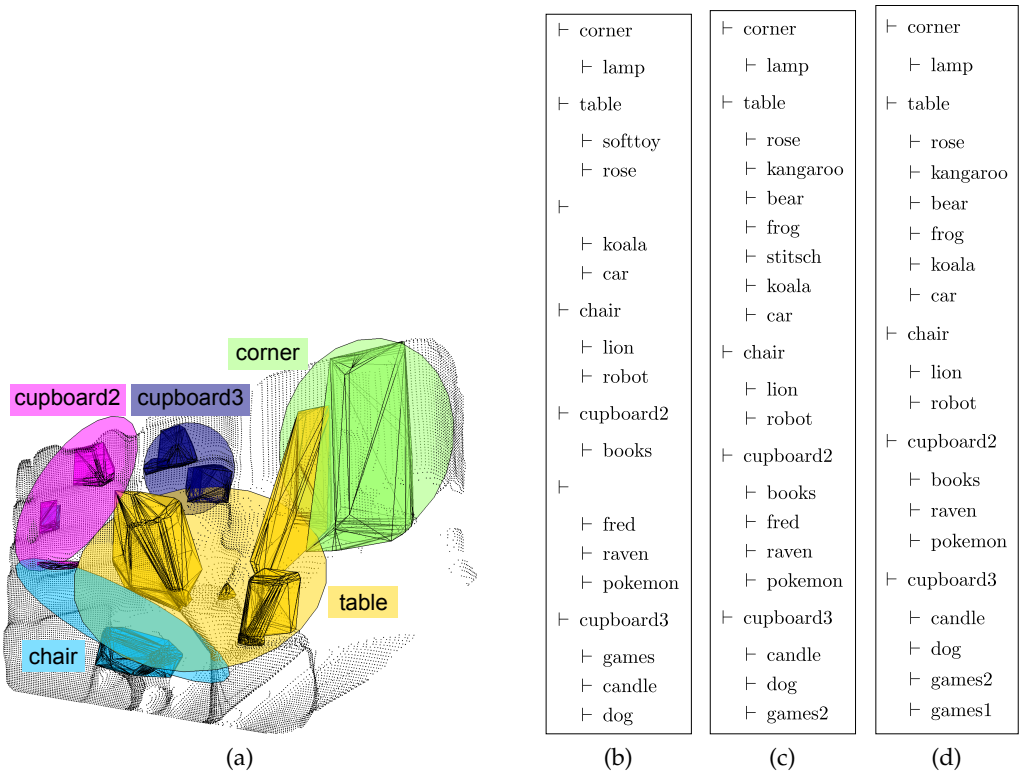


Figure 4.29: Using the description of subject 3(p), (a) shows the aligned scene model acquired using the erroneous object set introduced in this section. For better comparison, (b) shows the tree set representation of the description, (c) the final tree set of the aligned scene model with perfect object detection, and (c) with erroneous object detection.

4.6 CONCLUSION AND OUTLOOK

In this chapter, I have presented a methodology for estimating a 3D scene model from a given scene description. It matches at the same time the sensory perception of the robot and the underlying situation model of the human tutor. It uses the finding that the way spatial descriptions are constructed reveals information about the interlocutor's representation of the observed scene. The construction of spatial descriptions is driven by gravity, which means that the descriptions mainly consist of *orthogonal* relations between objects and their supporting structures, e. g., "lion on chair", and *parallel* relations between objects located on/in the same structure, e. g., "car in front of koala". Therefore, the first step in my computational model is the transformation of relational descriptions into a set of trees using rules handling orthogonal and parallel relations. Each parent node in the trees represents the supporting structure for the assigned child nodes. A first link to the perceptual reality is realized by using results of object detectors to compute the initial 3D scene model. For each parent node a potential planar patch can be estimated using the 3D locations of small detectable objects. These patches initially estimate the supporting structure in a scene. As, per definition, automatically detectable objects are only located in the leaves of the trees, the estimated patches represent level-1 structures in the trees. The patches form the initial model which is further adapted to fit bottom-up extracted patches in the perceived 3D data. The adaptation process corrects errors introduced when resolving category labels. Further, it provides missing links between scene elements which could not be inferred from the description itself because the information given explicitly is often incomplete and under-specified. In an exhaustive analysis of 30 descriptions given by 20 different persons for two scenes, a playroom and a living room scenario, it is shown that my approach can deal with a wide range of descriptions and description styles producing in all cases reliable scene structures. The final scene models produce, on the one hand, semantic structures aligned to the level of details of the respective dialog partner and provide, on the other hand, hints to the most prominent scene elements.

As outline in Section 4.2 the presented 3D scene model closes a gap in scene analysis which is the ability of providing models for complex 3D scenes. The contribution of my approach is a method which provides a link between abstract models and the perceptual reality allowing to resolve underspecified verbal descriptions through information gathered in a bottom-up manner from sensor data. In general, high-level knowledge is seen as a possibility to provide top-down guidance facilitating and improving low-level processes [Neu08]. The scene elements of our models are exactly such semantic knowledge which can be utilized by subsequent tasks with the main advantage that my models are not predefined but formed automatically and individually for each interaction partner and situation. My approach contributes to the goal of using learning techniques to acquire semantic structures of domains [Gal05]. The aligned scene models do not only offer high-level scene information but give also evidence for the question on how space is represented [Vas07b]. I can state that the representation of space is quite likely to be hierarchical as the system has

transformed successfully a huge amount of descriptions into sets of trees by just assuming that spatial descriptions mainly consist of explicit and implicit relations between objects and their supporting structures.

Next steps for improving my computational model will concentrate on developing methods for handling situations which are currently postponed. For example, competing parent labels or nameless structures in the final tree set cannot be handled automatically. Therefore, I recommended a clarification during a subsequent dialog. The challenge here is to develop mechanisms allowing the robot to ask questions to its tutor so that it can gain a maximum of information. Further, it has to be clarified whether new rules have to be invented handling the answers given to the questions.

Besides the extension of one specific model, research continuing scene modeling in general could have three directions: one affecting the shapes estimating the supporting structures, one emerging common structures within a room type, and one determining the degrees of freedom in the generated models. So far, all higher-level scene elements are estimated by planar surfaces. Supporting structures for objects which can be localized *on* it are represented appropriately. Also, structures where objects are attached *to it* are realizable in this way. But furniture for which objects have been appointed to be positioned *in* it are only partially modeled by planar surfaces. Therefore, it would be interesting to examine whether box-like shapes for supporting structures containing objects are better estimates than planar surfaces and whether they extend the scene model substantially. The second direction of research derives from the evaluation results providing prominent structures in a room like the sofa in the living room. The question is whether it is possible to develop mechanisms which extract from a set of models of a certain room type the most prominent scene structures within this room type and to detect them in another so far unknown room of the same room type. Such knowledge transfer is an important step for a robot on the way from specific models taught to it by a human tutor to *abstract scene models* which can be fitted to new scenes providing an initial guess without tutoring. On the one hand, it is not desirable to describe to a robot everything again and again. This means that it is mandatory for a robot to have a representation of structures that are stable over different persons and different rooms of the same type. On the other hand, the mechanisms allowing a flexible adaptation to the tutor's situation model should be kept as this model contains structures that are important to solve the current task. The last research direction considers the fact that parts of indoor environments are subject to changes. For example, chairs can be moved and doors can be opened. Further, the configuration of common structures in rooms of the same room type can differ. All these examples can be captured by learning the degrees of freedom in the specific models being an additional piece of information on the way towards an abstract scene model of a room type.

5

LEARNING ARTICULATED SCENE MODELS FROM SPATIAL CHANGES

So far, only static scenes have been considered. For realizing a spatial awareness in realistic environments, a robot must also be able to deal with dynamic environments where chairs are moved and doors are opened, or more general, where a human moves around and changes the scene layout. Concretely, this chapter focuses on situations where the robot observes (in 3D) a scene with modifications of the scene layout caused by an acting human. The so-called *Articulated Scene Model* is derived from spatial changes in a scene that are detected without any specific object knowledge. The range analysis from a certain view point allows to compute the static background using the farthest static depth measurements observed during an observation period. Arbitrary movable objects can be detected model-less from static depth measurements emerging in front of a known background scene. Moving entities are tracked with a weak cylinder model. The articulated scene model represents a scene on the intermediate level of movable resp. articulated scene parts. This representation level equates the representation level of the aligned scene model (→ Chapter 4). The articulated model gives movable scene structures which have been moved by the tutor while the aligned model provides scene structures the tutor has verbally referred to. This extends the data sources available to a robot for obtaining information about intermediate scene elements. Besides verbal descriptions, scene changes can now be utilized for scene analysis.

Section 5.1 presents some studies that examine the human ability to detect scene changes. Section 5.2 gives related work on background modeling and person tracking. Algorithms for computing the three components of the articulated scene model can be found in Section 5.3. Moving entities are tracked with a particle filter. Static background and movable objects arise simultaneously from comparing static depth measurements. An evaluation of the output is done on a set of different sequences. Results can be looked up in Section 5.4. Section 5.5 outlines some applications of the articulated scene modeling approach. Section 5.6 summarizes this chapter.

5.1 MOTIVATION

This section is going to motivate why spatial changes in a scene are a reliable input to guide a scene model formation process. The core idea of the articulated model is to observe over a short time period changes in the scene like a chair being moved. From these changes a model should be learned that encodes the static unchanging parts (e. g., walls), movable objects (e. g., chair), and moving entities (e. g., a human which could be a possible interaction partners). Instead of building a complex ontology of indoor rooms that describe which scene parts are static and which are movable, my methodology propose a light-weighted approach modeling a dynamic scene in a bottom-up way. The envisioned scenarios are scenes where a human – an independent entity regarding the underlying scene – acts in the environment by changing functional parts of the scene. This functional parts are represented in our model as articulated scene parts which have the property that their position only changes through manipulation of an agent. 3D data in general and SwissRanger data in particular enable the detection of changed scene parts and the adaptation of the known static background via simple difference computation between the current scene view and the learned background. The articulated components can be extracted independent from their shape or the fact that they stay static after their displacement. This representation is contrary to standard background and foreground segmentation techniques where a moved scene part will be detected after a sudden displacement but will be integrated over time into the background if it stays static.

What role does human activity play in the process of building scene representations [Vas07b]? In computer vision it is often assumed that observing motion patterns allows to discover scene structures that are not extractable in static scenes like, e. g., a dirt road [Dee08]. Unfortunately, less studies have examined the influence of scene changes on the formation of situation models in humans, so far. Much more effort has been laid on the contrary effect of *change blindness*. Several studies have investigated what causes that changes in the scene reach awareness or not. For example, Levin [Lev02] and Simons [Sim98] have found that attention and informativeness seem to play an important role. Nevertheless, an interesting definition of the concept *change* versus *motion* has been developed which has parallels to the technical-driven design of our articulated scene model. Rensink proposed to define *motion* as variation referenced to *location* and *change* as variation referenced to *structure* [Ren02]. 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 need therefore more sophisticated processing. The assumption of a separated processing of change and motion is realized in our model by two layers, one responsible for handling the articulated scene parts and one for handling moving entities. Beauchamp and colleagues [Bea02] even found through their fMRI studies evidence for two processing streams in human brains, one responsible

for motion of manipulable objects and one for human movements. The lateral temporal areas which strongly respond to moving stimuli in general are the Superior Temporal Sulcus (STS) and the Middle Temporal Gyrus (MTG). STS prefers human stimuli and the according complex articulated motion characteristic of biological motion. MTG is selective for the inarticulate motion characteristic of tools. Regarding changes, Rensink has further distinguished between *dynamic changes* which means perception of the transformation itself and *completed changes* where at some point the change of structure is perceived. Phenomenologically, the detection of completed changes involves a comparison of currently visible structures with a representation in memory.

Returning back to the examination of change blindness, newer studies seem to have found that even though a change is not detected with awareness it is still noticed implicitly as effects of change are visible in behavioral studies. Thronton and Fernandez-Duque [Thr02] summarize different studies giving evidence for implicit change detection. Further, they have reported for older adults a reduced ability to detect changes compared to young subjects. It is suggested that a narrowing of attentional breadth causes the slowdown. Although psychological experiments still have to give evidence, it seems that the high ability of children to detect scene changes plays a role in infant learning. It would be interesting to explore this role in further studies by examining the link between modification in sensing and learning. The articulated scene model relies on detection of completed changes and separated encoding of movable objects and moving humans. The focus on observation of scene changes follows the perspective of developmental robot learning which premise it is to incorporate cues that have shown importance in child learning. Detection of general movable objects is an important attention mechanisms giving a robot the possibility to be proactive. For example, it can learn automatically kinematic models for observed changes [Stu09]. Or, it can trigger subsequent tutoring situations where further information or demonstration is demanded from the interaction partner [Lüt09].

5.2 RELATED WORK

Relevant work in the field of dynamic scene analysis focuses on two main topics. Section 5.2.1 presents work for moving object detection via modeling the static background. Section 5.2.2 focuses on work for detecting movable objects that can change their location but are not detectable by standard background modeling techniques. Section 5.2.3 points out the contribution of my articulated scene model to the field of dynamic indoor scene modeling and detecting of semantic because articulated objects.

5.2.1 *Detection of Moving Objects and Static Scene Modeling*

In the field of video surveillance many work can be found that learn for a observed scene the static background with the aim to detect persons moving and cars driving. Diverse methods have been developed to model the background. Approaches range from classical Gaussian Mixture Models (GMMs) [Sta99] to codebooks encoding the pixels either separately from each other [Kim05] or incorporating nearby pixels using subspaces [Mit09]. For many approaches a static background is mandatory however Sheikh and Shah have introduced an approach which can cope with uniformly moving backgrounds like a river [She05]. Their approach relies on three innovations which are: the correlation in intensities of spatially proximal pixels, the temporal persistence, and the competitive modeling of foreground and background. Knowing the background allows to extract the moving foreground by subtracting the background from the current image.

Transferring approaches for moving object detection from surveillance to robotic scenarios, the problem of moving cameras has to be considered. This can be done by, e. g., detecting moving objects through inconsistencies in the scene motion computed using optical flow [Kla09]. Another problem in robotic scenarios is the short observation time so that which means that the background cannot

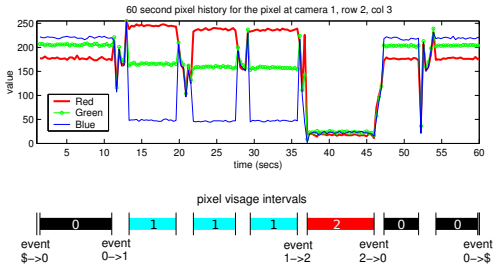


Figure 5.1: This figure is taken from [San02]. It shows for a pixel its history and a clustering of this history into temporal coherent clusters, the so-called temporal signatures.

Figure 5.2: In red an occluding objects is shown that have been detected by analyzing the silhouette distortion of the tracked human [Gua07].

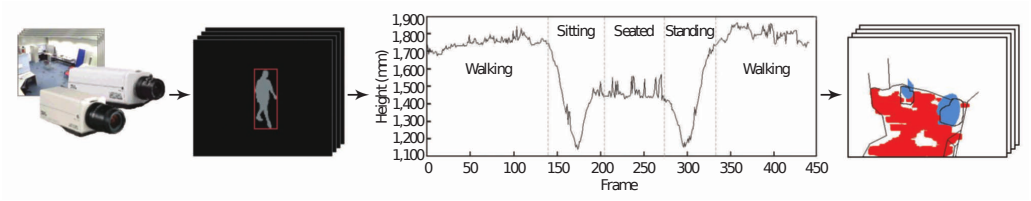


Figure 5.3: This figure shows the four major steps in interaction signature scene labeling of [Peu04]. Human trajectories are segmented into actions and used for incrementally labeling of the scene (blue: chairs, red: floor).

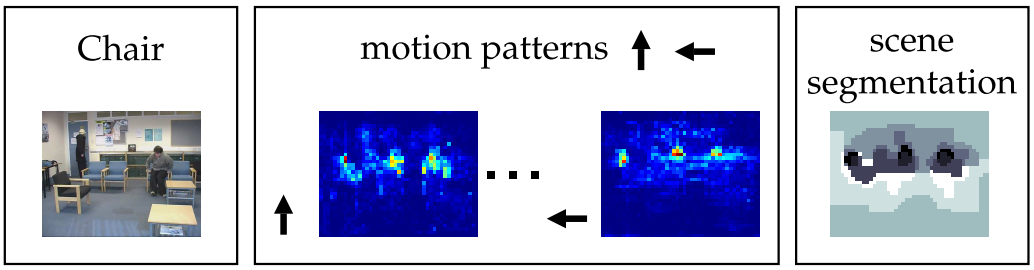


Figure 5.4: This figure shows a scene segmentation as proposed by Dee et al. [Dee08]. The semantic regions are an output of a clustering over dominant motion patterns.

be learned in advance. Therefore, Hayman and Eklundh [Hay03] developed a Bayesian model for incorporating the possibility that the background has not been uncovered yet. Recently in the field cars driving around in traffic scenes, the enhancing of tracking moving objects by a background model has been extended towards using an estimated scene layout. Scene labeling techniques determine the orientation of 2D areas in 3D which prune false positive detections of, e. g., cars and persons [Woj10, Hoi06].

5.2.2 *Detection of Movable Objects and Semantic Areas*

For dynamic scene analysis not only moving persons but also movable objects are of interest as they can become obstacles when driving around or can be salient regions in search tasks. Movable objects are characterized by occasional relocation and longer static periods. In classical background subtraction approaches such objects will be integrated into the background model after relocation which means that they cannot be detected after a while. Sanders et al. [San02] solve this problem by integrating pixel information over time. As shown in Figure 5.1, the pixel history is clustered to temporal coherent clusters, the so-called temporal signatures. This allows to detect quasi-static objects under the condition that these objects first arrive and then depart from the scene. The authors tested their approach on compact tangible objects like a can or a bowl. A restriction of this approach is the "first arrive and the depart"-requirement. Peursum and colleagues [Peu04] overcome this restriction when detecting chairs that can be relocated. They track humans and segment the trajectory into actions using Hidden Markov Models (HMMs)s. An action like "sitting down" can be

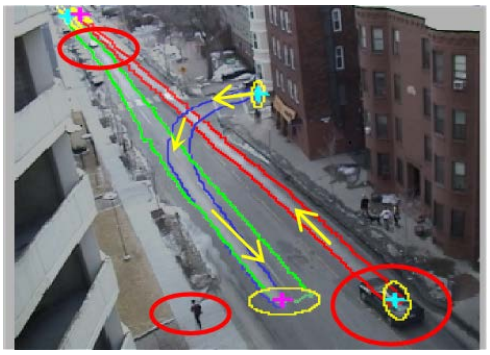


Figure 5.5: The lines and arrows show the automatically learned spatial layout of a far-field scene learned from three vehicle paths [Wano6]. Sources are marked by cyan crosses and sinks by magenta crosses.

associated with an object like a chair. The position of this object can be inferred from the location the action took place. As can be seen in Figure 5.3 the shape of the extracted chairs is quite rough. Figure 5.2 shows that a chair can be segmented with more details if the silhouette corruption of the tracked human is considered in the analysis of the scene [Guao7, Bro99]. It is assumed that from at least one view point the object will occlude the human so that occluding objects can be detected independent from their shape.

Certain actions or motions cannot only be connected to certain objects but points also to more general semantic regions. For example, Dee et al. [Deeo8] analyze per scene cell the dominant motion patterns to come up with a scene partitioning into semantic areas like, e. g., a "chair"-region (→ Figure 5.4). Koile et al. [Koio3] use a human activity analysis to come up with activity zones like a "lounge" area. The zones are used to trigger certain actions like "turn on overhead lights" if an activity in the "doorway" zone is detected. Analyzing person activities and car trajectories in outdoor environments has been used to provide semantic scene information like "roads", "paths", and "junctions" (→ Figure 5.5) [Wano6, Mako3] or, more general, "walkable" ground surfaces [Breo8]. A detailed review of further methods on scene activity understanding is given in [Buxo3].

5.2.3 *Contribution of the Articulated Scene Model*

My articulated scene model aims to combine background modeling with detection of semantic scene elements. I focus on the modeling of dynamic 3D scenes. The assumption, that static measurements which are farthest away determine the scene background, allows an elegant way to model the background especially in robotic scenarios where observation times are short. Subtracting the background in 3D reveals directly quasi-static/articulated objects without special requirements like, e. g., an object has to arrive and depart before it can be detected or its color signature must differ from the signature of the background [Sano2]. It is independent from the object's shape and size or the human activity connected to it [Deeo8, Peuo4]. Detecting arbitrary articulated scene elements using human activity would require recognition abilities of many different daily-life activities. A database of all possible actions would be necessary for training. In contrast to that, my approach provides for range data a bypass to this exhaustive learning problem.

5.3 THE ANALYSIS OF A DYNAMIC SCENE

This section presents our analysis of a dynamic scene resulting in an *Articulated Model* of the scene. Different aspects of this model have been addressed in three different publications [Beu10, Swa10a, Swa08a]. An important assumption in our approach is that the robot observes a vista space scene from a certain view point v for a short time interval Δt_v without moving the camera. The orientation of the camera is known or can be extracted. Hence, the data can be transformed so that the ground plane is parallel to the xz-plane of the robot's global coordinate system (→ Section 2.2.1). $\mathcal{M}^v$ denotes the final model for the view point v. The sequence of frames

$$\left\{ \mathcal{F}_t = \{ \vec{f}_t^{\,i} \}_{i \in \{ 1, \ldots, n \}} \right\}_{t=1}^{\Delta t_v}, \tag{5.1}$$

where each frame $\mathcal{F}_t$ consists of n 3D points $\vec{f}_t^{\,i}$, is processed sequentially producing for each time step t an articulated model $\mathcal{M}_t$.

Definition. ***Articulated model.***

$\mathcal{M}_t = \left(\mathcal{E}_t, \mathcal{S}_t, \mathcal{O}_t \right)$ *with*

$\mathcal{E}_t$, *holds moving entities, e. g., walking persons,*

$\mathcal{S}_t$, *the static background,*

$\mathcal{O}_t$, *movable objects like a relocated chair,*

and $\mathcal{E}_t \cap \mathcal{O}_t = \emptyset$.

The model $\mathcal{M}_t$ is forwarded to the next time step $t+1$ where it is updated with a new perception of the scene. Figure 5.6 shows the processing pipeline at time t. It consists of an entity tracking and a scene modeling component. The tracking module utilizes knowledge about the foregoing static background $\mathcal{S}_{t-1}$ for thinning out the data to potentially dynamic parts $\mathcal{D}_t^{\text{pot}}$. 3D velocity information, $\mathcal{V}_t$, and the past positions of moving entities, $\mathcal{E}_{t-1}$, are used to determine their new positions $\mathcal{E}_t$. Details on the algorithm are given in Section 5.3.1. Section 5.3.2 describes the scene modeling responsible for adapting the knowledge about the static background and for detecting current movable objects. The detected moving entities are used to determine in the current frame $\mathcal{F}_t$ the current static scene parts $\mathcal{S}_t^{\text{pot}}$. They are compared to the old static background $\mathcal{S}_{t-1}$ to determine where the background is confirmed, where a new one is introduced, and where movable objects are visible. As long as the camera stays static our system will accumulate more and more data improving the current articulated model until a camera rotation is detected and the final articulated model $\mathcal{M}^v$ for the current view v is gained. A detection of camera rotation can be done through a notification from motors driving the camera or through computer vision techniques detecting ego motion. For the new view point $v+1$ a new articulated model $\mathcal{M}^{v+1}$ is initialized. Models from different view points are in principle

independent from each other except for cases where the camera rotation between two view points is known. Then, scene knowledge in overlapping regions can be passed from one model to the other. Details can be looked up in Section 5.5.2.

The articulated model is developed to represent dynamic scenes visible in the vista space of the robot. Our assumptions are valid in the case of static cameras and camera motions arising from rotations around the camera axes. They allow a fast and reliable detection of scene changes without the use of strong object models. Incorporating camera motions arising from locomotion are out of scope of this thesis as methods handling such data are localized on the large-scale space level. Intermediate results of the system described in the following are visualized on the test sequence shown in Figure 5.7. There a person enters the scene from the right, picks up a chair and moves it to a room corner, opens the cupboard in the back and takes out a teddy bear, puts it on the table, and leaves the scene.

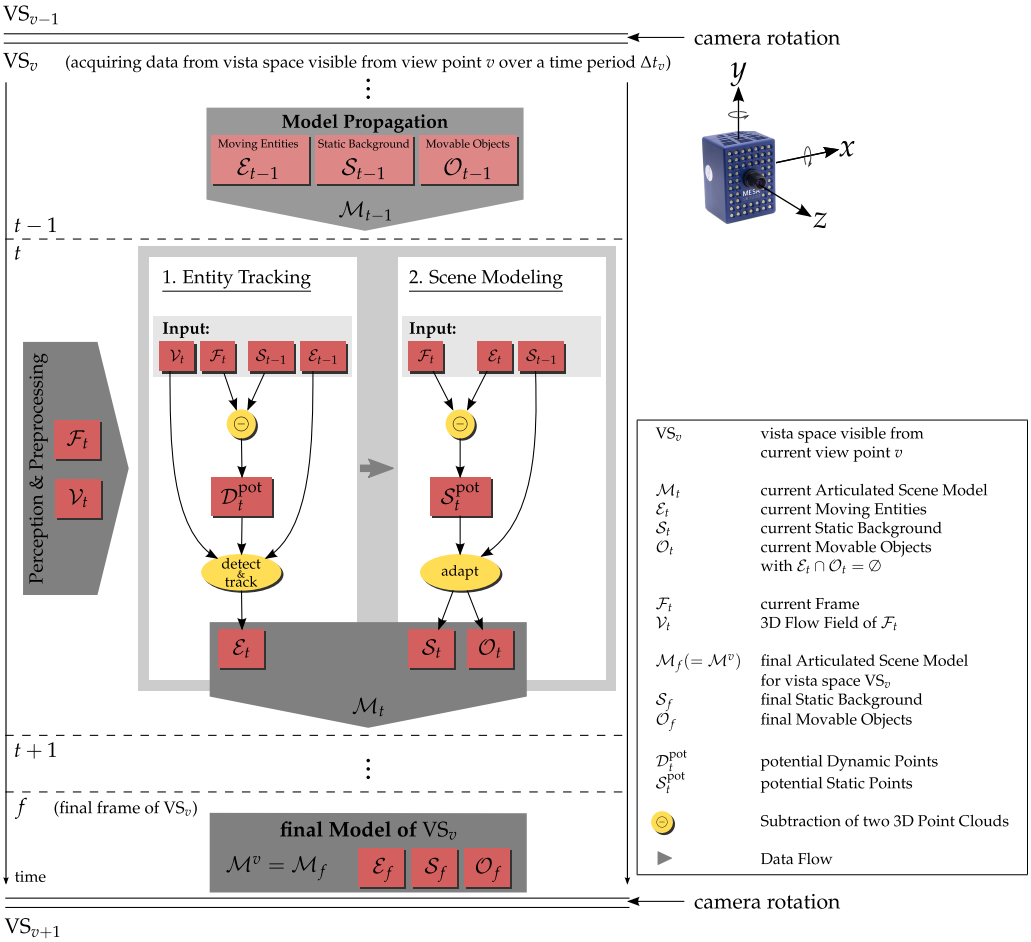


Figure 5.6: The processing of a dynamic scene at time t consists of an entity tracking and a scene modeling part. Both steps incorporate the model $\mathcal{M}_{t-1}$ of the foregoing processing step. The person tracking module subtracts (⊖) the known static background to thin out the current point cloud $\mathcal{F}_t$. The scene modeling part adapts the static background to $\mathcal{S}_t$ and detects movable objects $\mathcal{O}_t$ by extracting static points located in front of the known background. For the first frame of a vista space VS$_v$ the foregoing static background and moving entities are initialized as empty sets so that the entity detection has to be done on the complete point set of the first frame.

Figure 5.7: This figure shows some key frames of the test sequence $\mathcal{Q}_1^1$ which is used to visualize intermediate results of my algorithms. The scenario $\mathcal{Q}^1$ is observed from view point 1. In this scenario a person enters the scene from the right, picks up a chair and moves it to a room corner, opens the cupboard in the back and takes out a teddy bear, puts it on the table, and leaves the scene again.

5.3.1 *Entity Tracking*

Tracking moving entities fulfills in our scene analysis two main functions. On the one hand, the robot has to know potential interaction partners. On the other hand, neglecting moving objects rather than moving points is meant to generate better scene reconstruction results. The reason is that not all parts of a moving entity are necessarily labeled with large velocity vectors. As developing techniques for detection and tracking of moving entities is not the focus of this work, I have utilized the particle based approach developed by my colleague Joachim Schmidt [29] [Sch07]. My colleague Niklas Beuter [30] and I have improved the tracker through encountering static background knowledge. The following paragraphs give a short overview of the implemented algorithm.

DETERMINING POTENTIAL DYNAMIC POINTS. Due to the usage of a SwissRanger camera the algorithm for detecting and tracking moving objects has to deal with a dense 3D point cloud. As the original algorithm of Schmidt clusters sparse 3D data provided from a stereo camera using the complete linkage algorithm [Ber03] we introduce here a subtraction of the currently known static background $\mathcal{S}_{t-1}$ from the current frame $\mathcal{F}_t$. This increases the robustness of the

29 http://aiweb.techfak.uni-bielefeld.de/user/jschmidt
30 http://aiweb.techfak.uni-bielefeld.de/user/nbeuter

clustering introduced below and lowers the computational costs. The result is a subset of potential dynamic points

$$\mathcal{D}_t^{\text{pot}} \subset \mathcal{F}_t \tag{5.2}$$

which are passed to the subsequent hierarchical clustering. The subtraction $\mathcal{F}_t - \mathcal{S}_{t-1}$ can be computed efficiently since each 3D point provided by the SwissRanger camera has an unique identifier i given through the position of the corresponding pixel/pixel sensor in the image plane. A point $\vec{f}_t^{\,i}$ of the current frame $\mathcal{F}_t$ with the identifier i corresponds to the point $\vec{s}_{t-1}^{\,j}$ of the static background $\mathcal{S}_{t-1}$ if $j = i$. The same value of the identifier denotes a correspondence. If two corresponding points differ enough, hence, their Euclidean distance is bigger than a certain threshold θ_{dyn}, this point in the current frame is assumed to be a potential dynamic point. Further, if for a point of the current frame no static point is known it becomes also a potential dynamic point. Formally, the set of potential dynamic points $\mathcal{D}_t^{\text{pot}}$ is calculated as follows:

$$\begin{aligned}
\mathcal{D}_t^{\text{pot}} &= \mathcal{F}_t - \mathcal{S}_{t-1} \\
&= \left\{ \vec{f}_t^{\,i} \;\middle|\; |\vec{f}_t^{\,i} - \vec{s}_{t-1}^{\,i}| > \theta_{\text{dyn}} \vee \nexists\, \vec{s}_{t-1}^{\,i} \right\}, \quad \text{where} \\
\mathcal{F}_t &= \left\{ \vec{f}_t^{\,i} \right\}, \\
\mathcal{S}_{t-1} &= \left\{ \vec{s}_{t-1}^{\,i} \right\}, \quad \text{and} \\
i &\in \{\, 1, \ldots, n \,\}.
\end{aligned} \tag{5.3}$$

Due to the noise level of the SwissRanger camera (→ Section 2.3.1) the threshold θ_{dyn} is set to $\theta_{\text{dyn}} = 100\text{mm}$. For frame $\mathcal{F}_{22}$ of the test sequence $\mathcal{Q}_1^1$, Figure 5.9(a) shows in red the extracted potential dynamic points $\mathcal{D}_{22}^{\text{pot}}$.

HIERARCHICAL 6D CLUSTERING. The set of potential dynamic points is further simplified through clustering. Small contiguous regions are extracted based on spatial proximity of the 3D points and homogeneity of the velocities. It can be expected that the incorporation of velocity information improves the segmentation at this early stage without the need of strong models. For example, velocities can provide additional information for 3D points which ensures a separation of persons passing each other close-by. Therefore, we enhance the 3D points of the current frame $\mathcal{F}_t$ with velocity information $\mathcal{V}_t$ using optical flow estimation as presented in Section 2.5.1. The resulting 6D data is hierarchically clustered using complete linkage [Scho7, Bero3], also called furthest neighbor, deliberately oversegmenting the scene into small motion-attributed clusters. Each emerging cluster l is described through the 2D position of the centroid projected on the ground plane, a weight factor based on the number of assigned points, and the mean velocity computed from the velocities of the clustered points. Figure 5.9(b) shows the clusters computed for the example frame $\mathcal{F}_{22}$.

GENERATING AND TRACKING OF OBJECT HYPOTHESES.

A suitable representation of moving entities could be a simple cylindric object model with variable radius grouping clusters with similar velocity. This weak object model offers an entity hypothesis $e(\vec{a})$ based on a 5-dimensional parameter vector

$$\vec{a} = \begin{pmatrix} x \\ y \\ v_\theta \\ v_r \\ r \end{pmatrix} \tag{5.4}$$

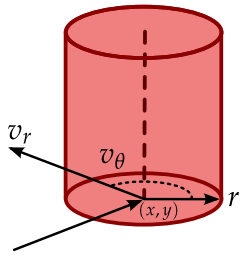


Figure 5.8: The weak cylindric entity model.

with x and y being the center position of the cylinder on the ground plane, r the radius of the cylinder, and v_θ the direction and v_r the magnitude of the object velocity. Figure 5.8 shows such a model cylinder.

A set of object hypotheses is generated from partitioning the observed scene into cylinders and including tracking results of the previous frame. The first one initializes the tracking and allows an error recovery. The second one predicts a set of hypotheses into the next frame and tracks them through a kernel based particle filter [Scho7]. Based on the position, size, and velocity of each entity $e_k^{t-1}(\vec{a})$ in the last frame $\mathcal{F}_{t-1}$ the parameters are predicted for the current frame $\mathcal{F}_t$ using a first order motion model Φ which creates a new hypothesis $e_k^t(\vec{a}^*)$:

$$\begin{aligned} e_k^t(\vec{a}^*) &\overset{\Phi}{\leftarrow} e_k^{t-1}(\vec{a}), \quad k = 1, \ldots, n \\ \vec{a}^* &= \Phi(\vec{a}, \dot{\vec{a}}). \end{aligned} \tag{5.5}$$

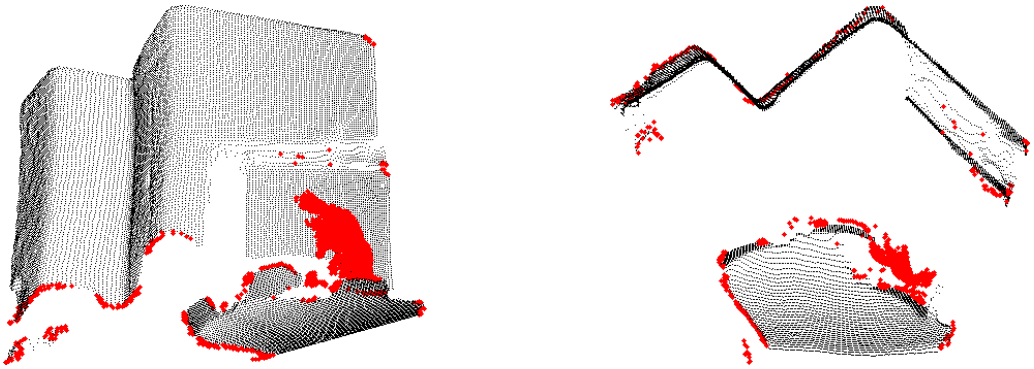
Each of these n hypotheses can be seen as a specific point in the parameter space, a so-called *particle*. Each particle is rated based on its value in the Probability Density Function (PDF) ρ computed from the relative position, relative velocity, and the weight of all motion-attributed clusters l within the cylinder e_k using Gaussian kernels:

$$\rho(e_k) = K_r(e_k) \sum_{l \in e_k} K_d(l, e_k) K_v(l, e_k), \quad \text{where} \tag{5.6}$$

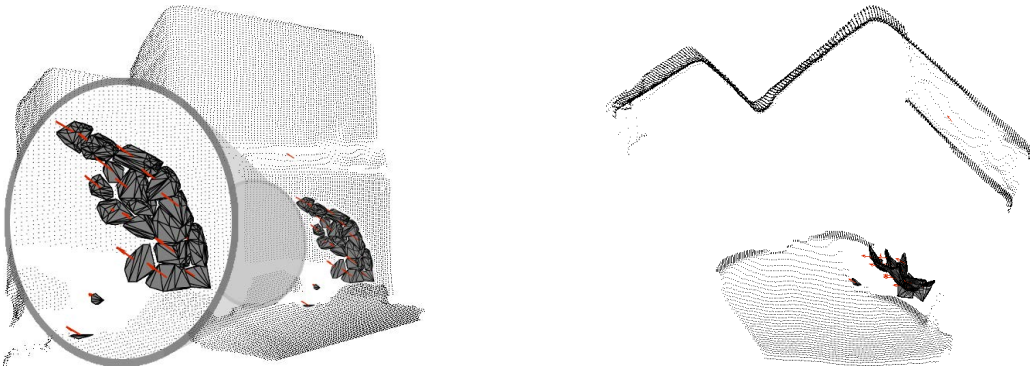
$$K_r(e_k) = \exp\left(-\frac{r(e_k)^2}{2H_{r,\min}^2}\right) - \exp\left(-\frac{r(e_k)^2}{2H_{r,\max}^2}\right) \tag{5.7}$$

$$K_d(l, e_k) = \exp\left(-\frac{||d(l) - d(e_k)||^2}{2H_d^2}\right) \tag{5.8}$$

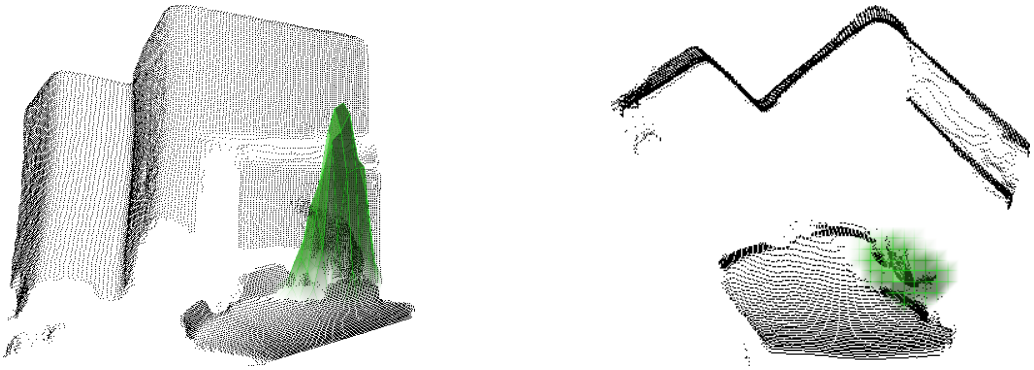
$$K_v(l, e_k) = \exp\left(-\frac{||v(l) - v(e_k)||^2}{2H_v^2}\right). \tag{5.9}$$



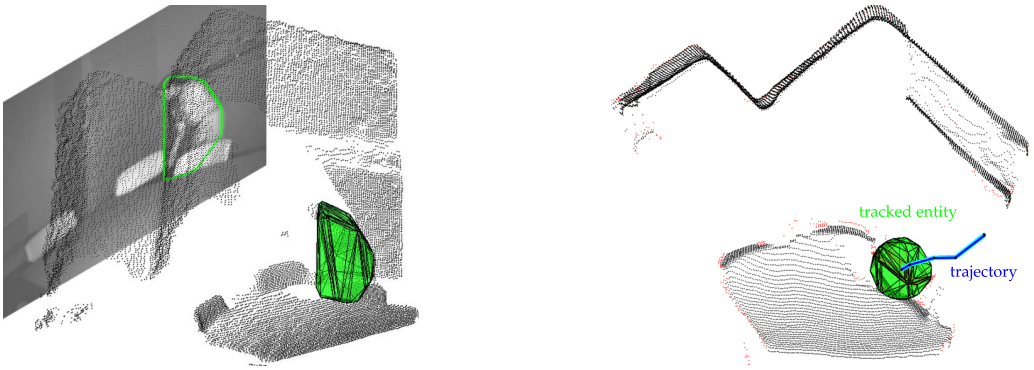
(a) in red: potential dynamic points $\mathcal{D}_{22}^{\text{pot}}$



(b) motion-attributed clusters

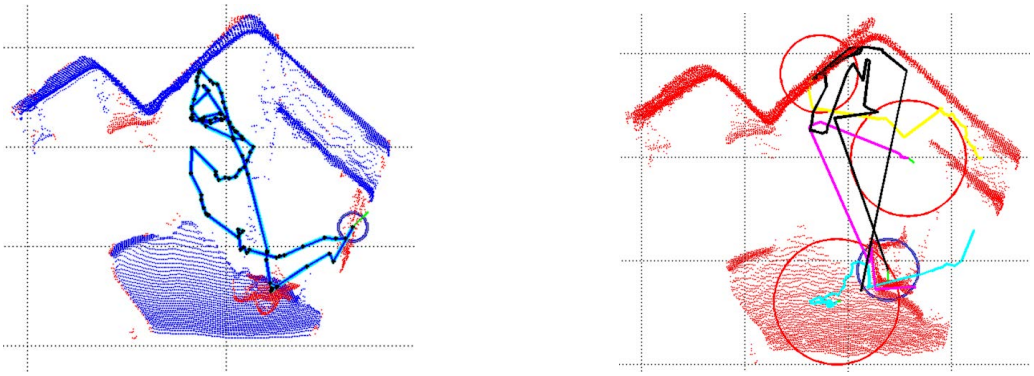


(c) density approximation



(d) in green: resulting entity hypothesis $\mathcal{E}_{22}$ and in blue: trajectory

Figure 5.9: Showing for frame $\mathcal{F}_{22}$ of test sequence $\mathcal{Q}_1^1$ ($\rightarrow$ Figure 5.7) in (a) the potential dynamic points $\mathcal{D}_{22}^{\text{pot}}$ determined by subtracting the background model $\mathcal{S}_{21}$, in (b) the motion-attributed clusters acquired through clustering of the dynamic points using spatial proximity and velocity homogeneity, in (c) the density approximation of ρ, and in (d) the resulting entity hypothesis $\mathcal{E}_{22}$ as 3D entity hull and projected 2D hull and the determined trajectory.

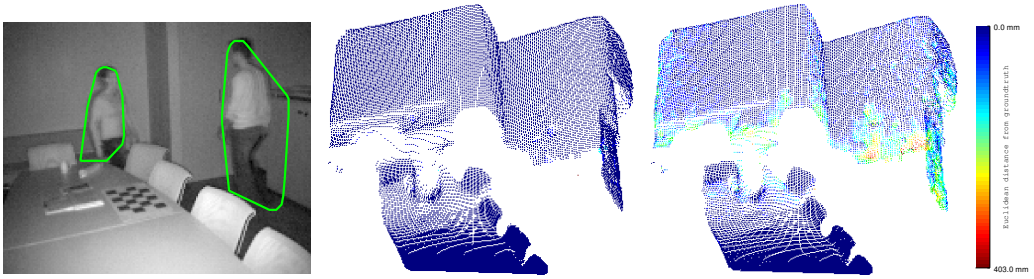


(a) Tracking with feedback of the background model.

(b) Tracking without background feedback.

Figure 5.10: This figure shows two trajectories acquired on the test sequence, one with feedback of the background model [Swa10a] and one without feedback [Swa08a]. It can be seen that knowledge about the static background reduces errors in determining the correct entity hypotheses.

The kernel K_r keeps the radius in a realistic range masking out all hypotheses with a too small or too big radius. K_d reduces the importance of clusters further away from the cylinder center. K_v masks out clusters having differing velocities. The functions $r(.)$, $d(.)$, and $v(.)$ extract the radius, the 2D position on the ground plane, and the velocity of a cluster l or a hypothesis e_k. The kernel widths H are determined empirically. Function ρ in Equation 5.6 is also called observation function of the particle filter. The outcome is a density approximation of the appendant clusters as shown in Figure 5.9(c). The maxima provide the actual moving entities. Several mean shift iterations refine the particles to concentrate at local maxima in the distribution. Individual particles selected from these best modes of the distribution represent entities found in the current frame. All dynamic points $\mathcal{D}_t^{\text{pot}}$ within the convex hull of each tracked hypothesis form the set $\mathcal{E}_t$ representing the found moving entities in frame $\mathcal{F}_t$. The set is passed for exclusion to the adaptive background modeling process ($\rightarrow$ Section 5.3.2). Figure 5.9(d) shows for frame $\mathcal{F}_{22}$ of the example sequence the resulting entity hypothesis $\mathcal{E}_{22}$ as 3D and 2D convex hull. By assigning an identifier to the tracked entities a trajectory can be created to analyze the movement of the entity. In the first iteration of our system [Swa08a] the moving entities are detect using all points of $\mathcal{F}_t$ neglecting the knowledge about the static background like proposed later in [Swa10a]. As visualized in Figure 5.10 the knowledge about the static background reduces errors during determining moving object hypotheses and leads to a more robust tracking results. This in turn has a positive effect on the adaptation of the background model as can be seen in Section 5.4.



(a) example frame of the sequence

(b) background reconstruction via excluding tracked entities, $\bar{e} = 22$mm

(c) background reconstruction via excluding moving pixels, $\bar{e} = 27$mm

Figure 5.11: (a) shows an example frame with detected moving entities. The frame belongs to a sequence where two persons are moving around and the static background stays unchanged. (b) shows the reconstructed static background by accumulating static points which are determined via excluding points that are part of moving entity hulls. (c) shows the reconstructed static background by accumulating static points which are determined via excluding points having velocity vectors larger than a certain threshold (here, $|\vec{v}_t^{\,i}| > 3$cm). The points are colorized using their distance to the corresponding ground truth measurement. Blue stands for a small distance to the ground truth which means a good background reconstruction and red stands for a large distance to the ground truth denoting a bad reconstruction result. The mean error $\bar{e}$ is computed from the Euclidean distance between each point pair consisting of a reconstructed background point and the ground truth point ($\rightarrow$ Equation 5.10).

5.3.2 *Static Background Adaptation and Movable Object Detection*

The knowledge about the currently moving entities can be used to reconstruct the current static scene. In the case of scenes where persons are only going around and do not change the environment we have shown in earlier work that simply accumulating and averaging static measurements reveal a reliable static background of the observed scene view [Swa08a]. A virtual frame with the same resolution like a SwissRanger frame (176×144 pixels) stores the static background. As the camera is not moving during the background reconstruction the same pixel indices define corresponding points in two different frames. For a current SwissRanger frame those points are defined as static which are not part of the detected moving entities $\mathcal{E}_t$ determined by the algorithm proposed in Section 5.3.1. These static points are accumulated pixel-wise over the whole sequence and averaged to one value per pixel. For a sequence containing frames that are only disturbed by moving persons a reliable background model can be estimated. Figure 5.11 shows how well the ground truth is met. A blue coloring of a point means a small distance to the corresponding ground truth point and a red coloring means a large distance. It also shows that using tracking for determining whether a point is static or not is more convincing than simply using the magnitude of the assigned velocity vector. Using a small velocity vector as indicator for a static background point suffers from noise since points on the body of a person that, for example, approaches the camera, may have small velocity vectors. They are misleadingly assigned to the background which introduces noise to the background reconstruction.

The static scene reconstruction based on the assumption of simply accumulating measurements of the static scene is not valid any more in scenarios where the scene is manipulated, for example, by relocating chairs or putting objects on tables. If a chairs is relocated it has been part of the static scene at its old position and is again part of the static scene at its new position. In the above approach, the chair at its old position will be still visible in the background model and will flatten out slowly if the scene behind the chair is observed over a longer time period. The chair at its new position will only appear slowly in the background model as especially in the beginning the measurements of the static scene behind the chair will still dominate the averaging.

> Therefore, I propose an *adaptive background modeling* based on the basic physical rule saying that under a fixed camera position the *farthest* static depth value ever measured for a certain pixel determines the static background.

It is assumed that all static measurements which are in front of a known static background arise from movable objects. Considering changes in the observed distance measurements allows in the situation of the relocated chair an immediate adaption of the background model at the old position of the chair to the now visible static scene behind the chair. Complex adaption techniques known from 2D background modeling are not needed. Further, it prevents the integration of the chair at its new position into the background model. Therefore, the knowledge about the static background is more and more refined and movable objects like chairs can be detected at their new locations. Static measurements which are significantly nearer than static measurements seen before refer to such movable objects.

The pseudo code of Algorithm 1 gives details about the adaptation of the static background and the detection of movable objects. For each time step t the current frame $\mathcal{F}_t$ is processed together with the static background $\mathcal{S}_{t-1}$ of the foregoing time step $t-1$ and the current moving entities $\mathcal{E}_t$ determined through the tracking described in Section 5.3.1. Each point $\vec{f}_t^{\,i}$ of the current frame $\mathcal{F}_t$ is tested whether it is part of a moving entity $\mathcal{E}_t$ ($\rightarrow$ line 2). If not, this point is part of the current static scene and will be compared to the corresponding background point of the foregoing background model $\mathcal{S}_{t-1}$ ($\rightarrow$ line 3). The Euclidean distance between $\vec{f}_t^{\,i}$ and $\vec{s}_{t-1}^{\,i}$ is computed and tested whether the distance is smaller than a certain threshold θ_d. If so, it is assumed that the current static measurement provides information for an already known background point so that it is accumulated to this background point increasing its reliability ($\rightarrow$ line 4). Due to the noise level of the camera, the value of the threshold θ_d is found empirically to be $\theta_d = 100$mm. If $\vec{f}_t^{\,i}$ and background ground point differ significantly it has to be decided whether a new background point or a movable object point is on hand. If the current measurement lies further away from the camera center it defines a new background point that has not been visible beforehand ($\rightarrow$ line 7). Otherwise, it arises from a movable object located in front of the known background $\vec{s}_t^{\,i} = \vec{s}_{t-1}^{\,i}$ ($\rightarrow$ line 9). This algorithm allows background model estimation and movable object detection without

using specialized object models or classifiers. Figure 5.12 shows three articulated scene models $\{\mathcal{M}_t\}_{t=50,70,122}$ computed on the test sequence $\mathcal{Q}_1^1$. The final static background $\mathcal{S}_{122}$, here plotted in blue, models reliably all scene parts that have never moved. It can be pointed out that the moved chair has been removed successfully from the background representation. Only the closed cupboard door is still part of the background (compare $t = 122$ of Figure 5.12) as range sensing of the cupboard's interior has not been possible during the observation phase. In contrast, the detection of the open door as a movable object is done well (compare $t = 70$ of Figure 5.12). In the final frame of the test sequence, the relocated chair and the bear on the table are correctly recognized as movable objects ($\rightarrow$ points colored in orange). Figure 5.16(o) shows a successful detection of a closed cupboard door. In this situation it has been possible to gather data from the interior of the cupboard.

Algorithm 1 *Adaptive Background Modeling and Movable Object Detection*

Input:
- $\mathcal{F}_t = \{\, \vec{f}_t^{\,i} \,\}$ \\ current frame
- $\mathcal{S}_{t-1} = \{\, \vec{s}_{t-1}^{\,i} \,\}$ \\ background of the foregoing time step $t-1$
- $\mathcal{E}_t$ \\ current moving entities

Output:
- $\mathcal{S}_t = \{\, \vec{s}_t^{\,i} \,\}$ \\ new background of current time step t
- $\mathcal{O}_t$ \\ detected movable objects

i: unique position in the 2D image plane with a pixel resolution of 176×144.
($\rightarrow n = 176 \cdot 144$)

for $i = 1$ to n **do**
  if $\vec{f}_t^{\,i} \notin \mathcal{E}_t$ **then**
    if $|\vec{s}_{t-1}^{\,i} - \vec{f}_t^{\,i}| < \theta_d$ **then**

$$w^i = w^i + 1; \qquad \backslash\backslash \text{ number of accumulated values}$$
$$\vec{s}_t^{\,i} = \vec{s}_{t-1}^{\,i} + \frac{1}{w^i}(\vec{f}_t^{\,i} - \vec{s}_{t-1}^{\,i});$$

    else
      if $|\vec{f}_t^{\,i}| > |\vec{s}_{t-1}^{\,i}| \;\vee\; \vec{s}_{t-1}^{\,i} = \varnothing$ **then**

$$\vec{s}_t^{\,i} = \vec{f}_t^{\,i};$$
$$w^i = 1;$$

      else

$$\vec{s}_t^{\,i} = \vec{s}_{t-1}^{\,i};$$
$$\mathcal{O}_t = \mathcal{O}_t \cup \vec{f}_t^{\,i};$$

      end if
    end if
  end if
end for

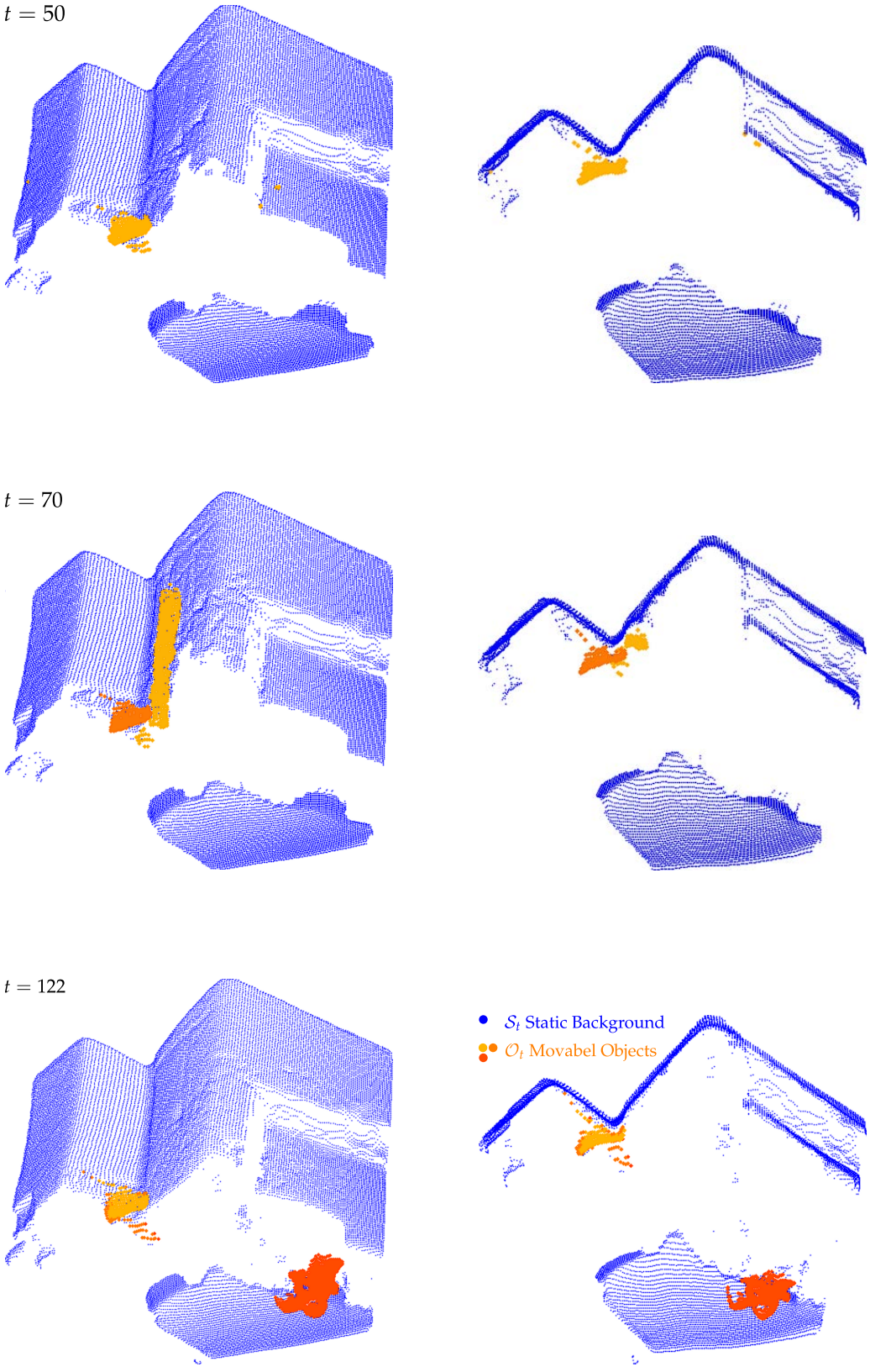


Figure 5.12: Here, three articulated scene models $\mathcal{M}_t$ (with $t = 50, 70, 122$) of the test sequence of Figure 5.7 can be seen. It shows in blue the currently acquired static background model $\mathcal{S}_t$ and in orange the movable objects $\mathcal{O}_t$. $\mathcal{S}_{122}$ and $\mathcal{O}_{122}$ represent the articulated model of the final frame $\mathcal{F}_{122}$ which forms the articulated model $\mathcal{M}^v$ for the current view v on the vista space VS_v.

5.4 EVALUATION

The quality of the acquired articulated scene models is judged by computing the error between the estimated static backgrounds and the ground truth. This is reasonable as moving entities and movable objects are only snapshots of the particular frame while the static background fuses the information of all frames and evolves over time. For each scene used for evaluation a ground truth of the scene without moving persons and movable objects is acquired from the same view point. A mean error $\bar{e}$ and a standard deviation σ between a model $\mathcal{M} = \{\vec{p}_i\}$ and its ground truth $\mathcal{M}^{\mathrm{GT}} = \{\vec{p}_i^{\,\mathrm{GT}}\}$ is computed by averaging the Euclidean distances $\{e_i\}$ between the corresponding point pairs $\{(\vec{p}_i^{\,\mathrm{GT}}, \vec{p}_i)\}$:

$$\begin{aligned} \bar{e} &= \frac{1}{n}\sum_{i=1}^{n} e_i \qquad \text{with } e_i = \left|\vec{p}_i^{\,\mathrm{GT}} - \vec{p}_i\right| \\ \sigma^2 &= \frac{1}{n-1}\sum_{i=1}^{n}\left(e_i - \bar{e}\right)^2 \end{aligned} \tag{5.10}$$

Using the error computation described above different approach for acquiring a static background from a sequence of observed frames are compared to each other:

$\mathcal{M}^{\mathrm{ADAPT}}$	provides a static background using *tracking* and *adaptive background modeling* presented in Section 5.3 and [Swa10a],
$\mathcal{M}^{\mathrm{TRACK}}$	accumulates all 3D points which are not part of *tracked* entity hulls [Swa08a],
$\mathcal{M}^{\mathrm{MPIX}}$	accumulates all 3D points with a *velocity vector* smaller than a certain threshold $\theta_v = 30\mathrm{mm}$, and
$\mathcal{M}^{\mathrm{MEAN}}$	computes a mean frame from all observed frames *without excluding* points.

Results acquired on a test sequence are discussed qualitatively in Section 5.4.1 while a quantitative analysis of diverse test sequences is given in Section 5.4.2.

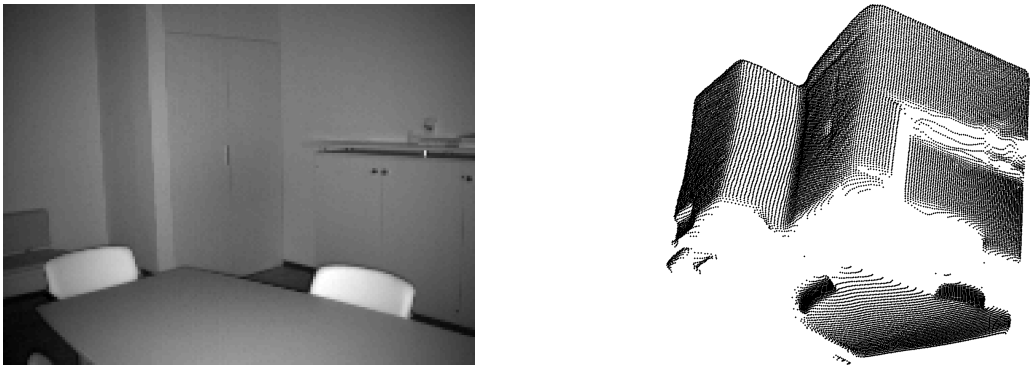
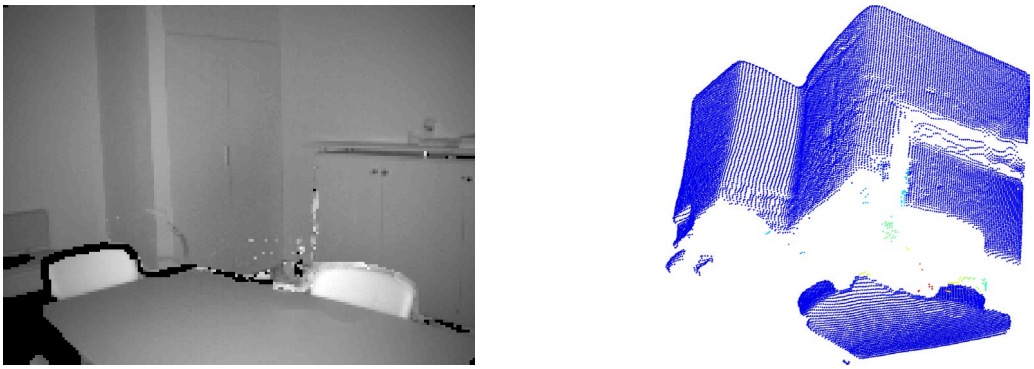
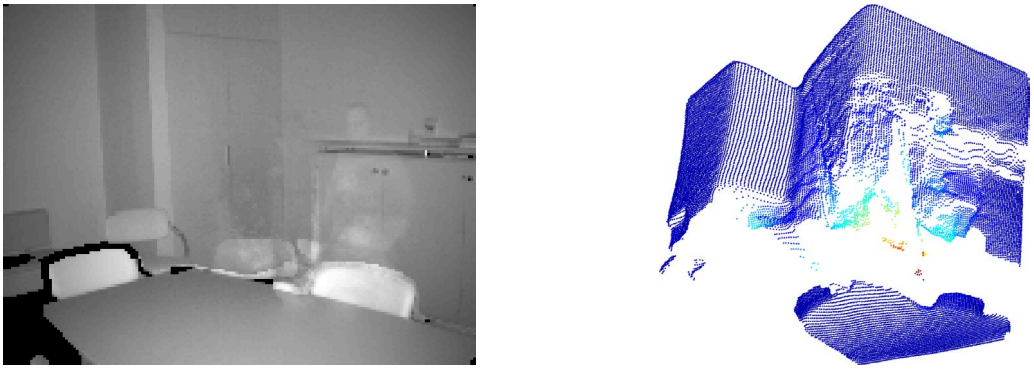


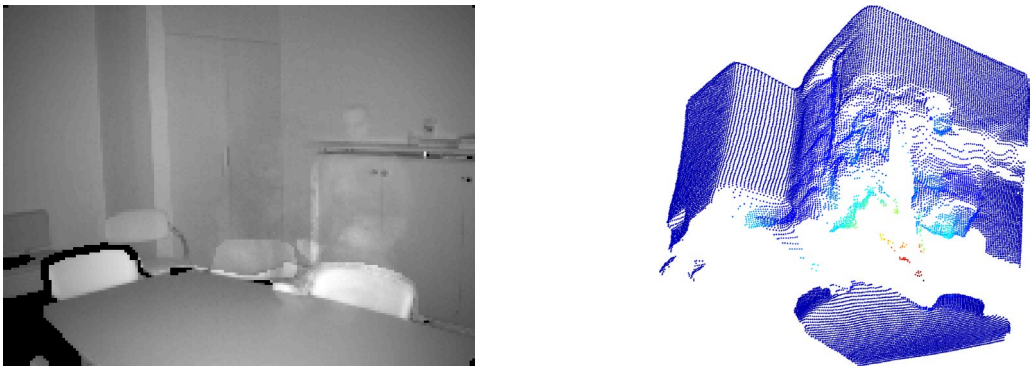
Figure 5.13: Here, the ground truth of the static background of the test sequence $\mathcal{Q}_1^1$ can be seen: (left) amplitude image and (right) 3D point cloud.



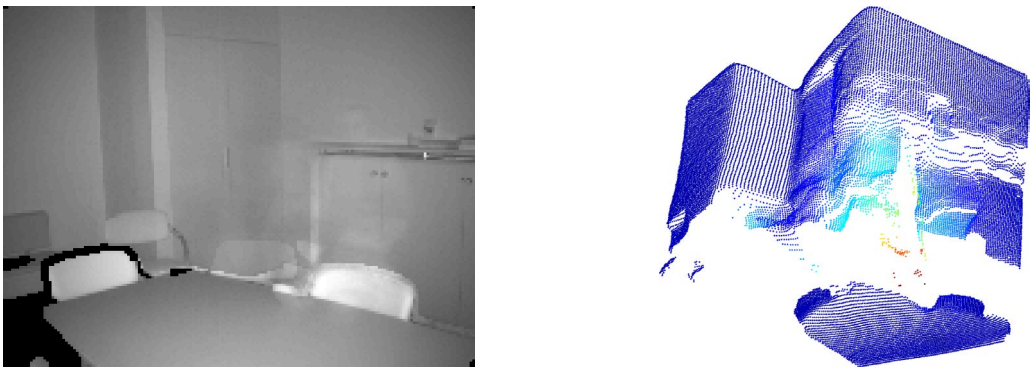
(a) static background of model $\mathcal{M}^{\text{ADAPT}}$



(b) static background of model $\mathcal{M}^{\text{TRACK}}$



(c) static background of model $\mathcal{M}^{\text{MPIX}}$



(d) static background of model $\mathcal{M}^{\text{MEAN}}$

Figure 5.14: Four static backgrounds acquired from the test sequence $\mathcal{Q}_1^1$ using four different approaches are shown here. The points are colored with respect to their distance to the corresponding ground truth point. Blue denotes a small distance and red a large distance. On the left side the amplitude images of the backgrounds are shown.

5.4.1 *Qualitative Evaluation of a Test Sequence*

The backgrounds of the four models $\mathcal{M}^{\text{ADAPT}}$, $\mathcal{M}^{\text{TRACK}}$, $\mathcal{M}^{\text{MPIX}}$, and $\mathcal{M}^{\text{MEAN}}$ shown in Figure 5.14 are acquired from the test sequence $\mathcal{Q}_1^1$ presented in Figure 5.7 and are compared with the ground truth shown in Figure 5.13. For each point $\vec{p}_i$ of the examined model $\mathcal{M}$ the error e_i is computed using Equation 5.10. In Figure 5.14 the points are colored according to the computed error. Similar to the depiction in Figure 5.11 the blue means small error and red a big error. The background model $\mathcal{M}^{\text{ADAPT}}$ of the articulated scene model ($\rightarrow$ Figure 5.14(a)) is nearly ideal while models acquired by the other methods show reconstruction errors. The amplitude images of the backgrounds illustrate clearly the benefit of the adaptive background modeling as, e. g., the chair at its old position does not appear in the background of $\mathcal{M}^{\text{ADAPT}}$. While in the other background models the chair is still visible. Further, the chair at its new position, the open cupboard door, and the pausing person for which the entity detection has failed are slightly apparent. Figure 5.14(d) shows also that averaging techniques from surveillance scenarios encounter huge problems in background modeling as the observation time is too short for removing changing objects from the background. If depth information is available the farthest measurement assumption produces good background models for scenes that can only be observed for a short time period and that contain many dynamics and changes.

5.4.2 *Quantitative Evaluation of a Set of Test Sequences*

A quantitative evaluation is performed on altogether 15 sequences acquired in 5 scenarios. The goal is to support the qualitatively impressions gained in Section 5.4.1. The film strips in Figure 5.7 and Figure 5.15 give a glimpse of what happens in the scenarios. In $\mathcal{Q}^1$ a chair is picked up and put down next to the cupboard, the left cupboard door is opened and a teddy bear is fetched and laid down on the table. This scenario is observed from two view points. The two sequences are tagged $\mathcal{Q}_1^1$ and $\mathcal{Q}_2^1$. In $\mathcal{Q}^2$, soft toys are removed from the table one after the other and positioned on the sideboard at the wall. This action is performed three times and observed from two view points leading to six sequences $\mathcal{Q}_1^2$, $\mathcal{Q}_2^2$, $\mathcal{Q}_3^2$, $\mathcal{Q}_4^2$, $\mathcal{Q}_5^2$, and $\mathcal{Q}_6^2$. Scenario $\mathcal{Q}^3$ shows a person opening and closing a door while leaving the room through this door. Repeating this action twice and observing it from two view points results in four sequences $\mathcal{Q}_1^3$, $\mathcal{Q}_2^3$, $\mathcal{Q}_3^3$, and $\mathcal{Q}_4^3$. In sequence $\mathcal{Q}_1^4$, a person takes a box out of the shelf and puts it on the table. In sequence $\mathcal{Q}_1^5$, a person collects soft toys and places them in the shelf. In sequence $\mathcal{Q}_1^6$, a cupboard door is opened and a watering can is fetched and placed on top of the cupboard. With this sequence it is shown that cupboard doors can be detected if during observation a look into the cupboard has been possible. To summarize, we have aimed to record data that covers different motion behaviors of humans, like going fast or slow or even stopping, and a variety of interactions with the environment, ranging from free object rearrangements to predetermined manipulations of, e. g., doors.

	$\mathcal{M}^{\text{ADAPT}}$	$\mathcal{M}^{\text{TRACK}}$	$\mathcal{M}^{\text{MPIX}}$	$\mathcal{M}^{\text{MEAN}}$
$\mathcal{Q}_1^1$	20 ± 96	84 ± 182	71 ± 155	95 ± 187
$\mathcal{Q}_2^1$	16 ± 37	85 ± 140	80 ± 118	108 ± 147
$\mathcal{Q}_1^2$	18 ± 59	71 ± 166	64 ± 121	103 ± 177
$\mathcal{Q}_2^2$	19 ± 47	108 ± 209	74 ± 184	106 ± 204
$\mathcal{Q}_3^2$	21 ± 61	75 ± 189	79 ± 185	124 ± 222
$\mathcal{Q}_4^2$	24 ± 78	97 ± 212	111 ± 216	157 ± 284
$\mathcal{Q}_5^2$	24 ± 68	79 ± 308	99 ± 230	142 ± 278
$\mathcal{Q}_6^2$	21 ± 55	98 ± 219	95 ± 193	147 ± 262
$\mathcal{Q}_1^3$	14 ± 26	51 ± 165	163 ± 328	219 ± 403
$\mathcal{Q}_2^3$	75 ± 319	74 ± 218	299 ± 635	321 ± 639
$\mathcal{Q}_3^3$	18 ± 64	356 ± 677	229 ± 588	234 ± 451
$\mathcal{Q}_4^3$	98 ± 404	246 ± 601	229 ± 588	246 ± 594
$\mathcal{Q}_1^4$	20 ± 58	71 ± 141	63 ± 145	89 ± 105
$\mathcal{Q}_1^5$	22 ± 52	134 ± 712	61 ± 125	85 ± 183
$\mathcal{Q}_1^6$	55 ± 146	207 ± 317	182 ± 284	182 ± 284

Table 5.1: The mean error and the standard deviation ($\bar{e} \pm \sigma$ in mm) is given for each static background acquired for 15 test sequences using 4 different approach. The mean errors are computed using Equation 5.10. Green colored cells highlight the best reconstructed backgrounds and red the worst reconstructed ones.

Table 5.1 summarizes for the 15 test sequences the mean errors $\bar{e}$ of the computed background models. Per sequence four models are computed which are compared to the ground truth model using Equation 5.10. It can be seen that my adaptive background modeling (column $\mathcal{M}^{\text{ADAPT}}$) produces models with promising small reconstruction errors. The error $\bar{e}$ is never above 100mm and deviates by $\pm$20mm. The best and the worst reconstructed sequences belong both to scenario $\mathcal{Q}^3$ where a person closes and opens a door. The walls of the room and the wall of the hallway behind the door form the static background. The door should be detected as movable object. In sequence $\mathcal{Q}_1^3$, the wall of the hallway is well visible through the open door thus it can be reconstruct well. In sequence $\mathcal{Q}_4^3$ the mentioned wall is hardly visible because the person is mostly covering this wall. Consequently, it is reconstructed quite noisy leading to a bad segmentation of the door.

The small standard deviations σ for the mean errors of $\mathcal{M}^{\text{ADAPT}}$ demonstrate that nearly every point of the background model is reconstructed well. $\mathcal{M}^{\text{ADAPT}}$ outperforms clearly the naive approaches $\mathcal{M}^{\text{MEAN}}$ and $\mathcal{M}^{\text{MPIX}}$. The models of $\mathcal{M}^{\text{ADAPT}}$ are also significantly better than those produced by the $\mathcal{M}^{\text{TRACK}}$ approach. The average improvement is 71%. The $\mathcal{M}^{\text{TRACK}}$ approach has problems in situations where the person which should be tracked is standing. In these cases the knowledge about the static background can help to detect and track these entities. Figure 5.16 presents for the 15 test sequences the static backgrounds and the movable objects of the estimated articulated scene models. They give an impression of the wide variability of detectable movable objects ranging from several soft toys to chairs and doors.

(a) Scenario $\mathcal{Q}^2$ where soft toys are picked up from the table and put on the sideboard at the wall. Six sequence are acquired: $\mathcal{Q}^2_1$, $\mathcal{Q}^2_2$, $\mathcal{Q}^2_3$, $\mathcal{Q}^2_4$, $\mathcal{Q}^2_5$, $\mathcal{Q}^2_6$.

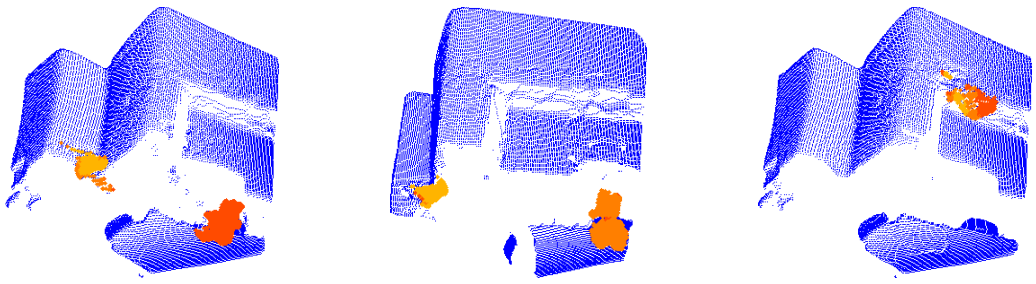
(b) Scenario $\mathcal{Q}^3$ where a person opens and closes a door while leaving the room. Four sequence are acquired: $\mathcal{Q}^3_1$, $\mathcal{Q}^3_2$, $\mathcal{Q}^3_3$, $\mathcal{Q}^3_4$.

(c) Scenario $\mathcal{Q}^4$ where a person picks up a box in the shelf and puts it on the table. One sequence is acquired: $\mathcal{Q}^4_1$.

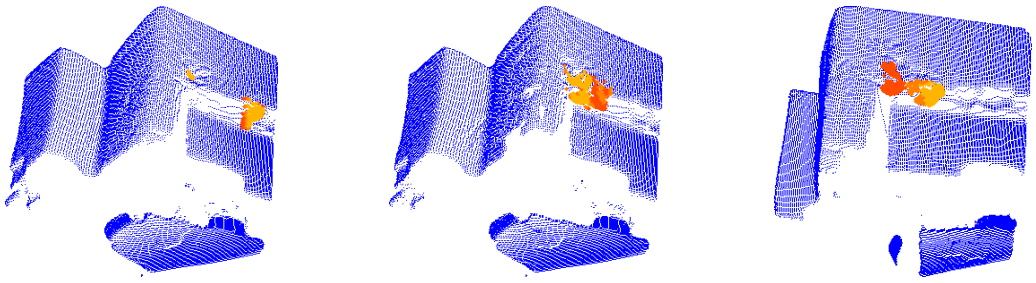
(d) Scenario $\mathcal{Q}^5$ where a person collects soft toys spread over the room and places them in the shelf. One sequence is acquired: $\mathcal{Q}^5_1$.

(e) Scenario $\mathcal{Q}^6$ where a person opens a cupboard, takes out a watering can, and closes the cupboard again. One sequence is recorded: $\mathcal{Q}^6_1$.

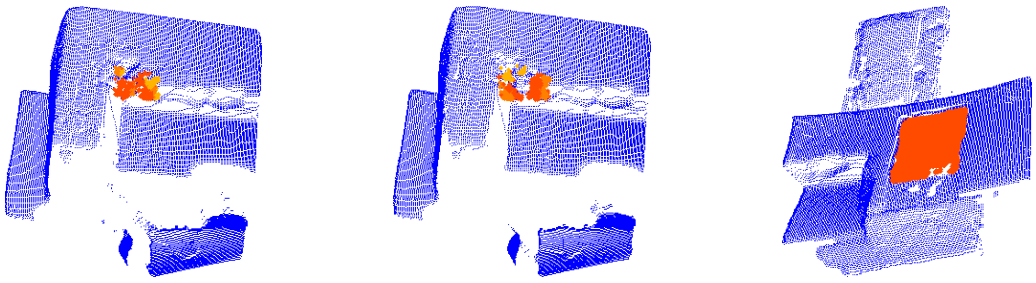
Figure 5.15: These strips show five of the six test scenarios from which in total 15 sequences have been acquired for evaluation. The scenario $\mathcal{Q}^1$ is already shown in Figure 5.7. The corresponding articulated scene models can be seen in Figure 5.16.



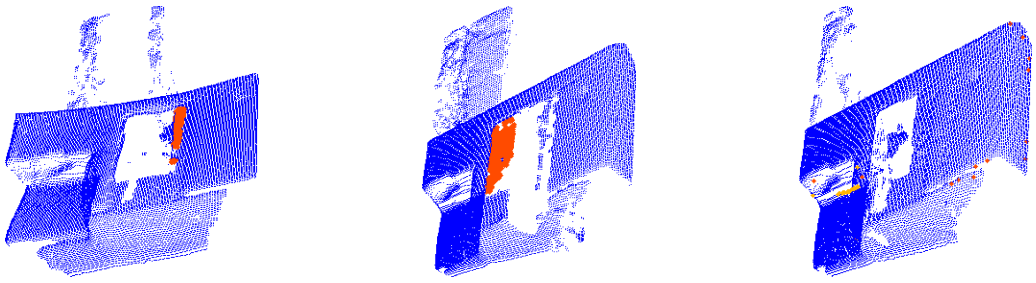
(a) $\mathcal{Q}_1^1$ (b) $\mathcal{Q}_2^1$ (c) $\mathcal{Q}_1^2$



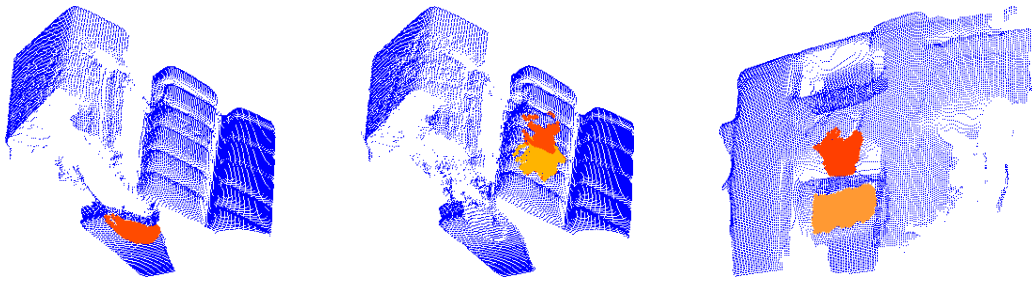
(d) $\mathcal{Q}_2^2$ (e) $\mathcal{Q}_3^2$ (f) $\mathcal{Q}_4^2$



(g) $\mathcal{Q}_5^2$ (h) $\mathcal{Q}_6^2$ (i) $\mathcal{Q}_1^3$



(j) $\mathcal{Q}_2^3$ (k) $\mathcal{Q}_3^3$ (l) $\mathcal{Q}_4^3$



(m) $\mathcal{Q}_1^4$ (n) $\mathcal{Q}_1^5$ (o) $\mathcal{Q}_1^6$

Figure 5.16: This figure shows the articulated scene models of the 15 test sequences shown in Figure 5.15. The blue points mark the estimated static background $\mathcal{S}_f$ of the scene $\mathcal{Q}$ and the orange points the detected movable objects $\mathcal{O}_f$.

5.5 APPLICATIONS OF THE ARTICULATED SCENE MODEL

This section is going to outline example applications for the articulated scene model $\mathcal{M}^v$ acquired for the scene view v. Section 5.5.1 presents how points forming the set of movable objects $\mathcal{O}_t$ can be segmented into coherent object regions. Section 5.5.2 demonstrates how articulated models can be transformed from one view point to another view point. Last, Section 5.5.3 shows how the detection of movable objects can be used to learn kinematic models for the observed manipulation.

5.5.1 *Object Segmentation*

The articulated scene model $\mathcal{M}_t$ of a frame $\mathcal{F}_t$ provides with $\mathcal{O}_t$ points which form objects that have been moved at least once during the observation. $\mathcal{O}_t$ holds for the frame $\mathcal{F}_t$ all movable points. An additional mechanism is necessary to segment this set of points into coherent object regions. This segmentation can be done by incorporating the appearance frequency of an object point. This frequency can be computed by accumulating over all foregoing frames $\{\mathcal{F}_i\}_{i=1,\ldots,t}$ the movable object information $\{\mathcal{O}_i\}_{i=1,\ldots,t}$. Assuming that objects appear one after the other a significant difference in the appearance frequency can be used to segment movable points into separated objects. Figure 5.17 gives examples of segmented objects. Due to the general approach, where first movable object points are detected as points popping out of the static background and second these points are separated based on their appearance or observation frequency, a wide range of different objects can be detect without a necessity for strong object detectors or priors. The object masks in Figure 5.17 can be used to extract object patches that can be passed to object classifiers [Som10], can prompt a label from the human tutor [Lüt09], or can be used to extract further object information like shape or texture.

5.5.2 *Model Propagation from View to View*

In natural observation scenarios an agent will mostly not stare at one point in the scene but will let the view wander around. If a robot simulates this behavior it will observe the environment from a view v for a short time period and will then pan its camera to a new view $v+1$. The panning could be triggered by following a person leaving the current field of view. As outlined in Section 5.3 two independent articulated models, $\mathcal{M}^v$ and $\mathcal{M}^{v+1}$, will be learned for the two views, v and $v+1$. If the camera is only rotated and the rotation is known, the model $\mathcal{M}^v = (\mathcal{S}^v, \mathcal{O}^v)$ can be propagated to the new view $v+1$. The assumption that the farthest measurements determine the static background is still valid. Figure 5.18 shows the projection of the static background $\mathcal{S}^v$ on the new view $v+1$. Instead of initializing the tracking in the new view on the complete set of points a subset of points is already recognized as static and can

be excluded. The final static backgrounds, $\mathcal{S}^v$ and $\mathcal{S}^{v+1}$ can be registered using, e. g., a variant of the Iterative Closest Points (ICP) [Bes92] algorithm tuned to Time-of-Flight (ToF) data [Swa07], and thinned out using, e. g., Virtual Image Plane Projection (VIPP) [Swa08b]. The resulting point cloud covers a wider field of view compared to one SwissRanger frame and fuses knowledge about the static background of two different views. This is one possibility how to integrate vista space and large-scale space representations which can be used for navigation or building a full model of the spatial environment.

5.5.3 *Object Articulation*

In a tutoring situation between a robot and a human one can imagine a situation where the human tries to teach the function of a cupboard drawer or a cupboard door to the robot. For example, this can be done by showing the robot a drawer or a door that is opened or closed. Abstractly spoken, the robot observes a scene where movable objects are moved along a certain path. A characteristic of such objects are potential motion paths for which kinematic models can be learned, for example, using the body model inspired approach of Sturm and colleagues [Stu09, Stu10]. They observe in their work the motion of objects through tracking markers attached to the objects. In cases where flat objects like a drawer are observed an extraction of rectangular patches provides at each time step a position of the drawer. For each object trajectory an articulation model is selected which explains the trajectory best. Figure 5.19 shows a kinematic model fitted to a trajectory of a surface.

My articulated scene model approach can provide position observations for arbitrary objects. Tracking through markers or an object segmentation specialized for specific objects is not necessary. If a manipulation of *one* object is observed the set of movable object points $\mathcal{O}_t$ provides in each time step t the corresponding object position. An accumulation of this movable point sets over a certain time period Δt gives a set of object positions which can be directly passed to Sturm's kinematic model computation. Figure 5.20 shows the different positions of two drawers fused in one image. The positions are provided by the articulated scene modeling of the observed sequence of frames. Each frame is associated with a color allowing to see the articulation of the drawers.

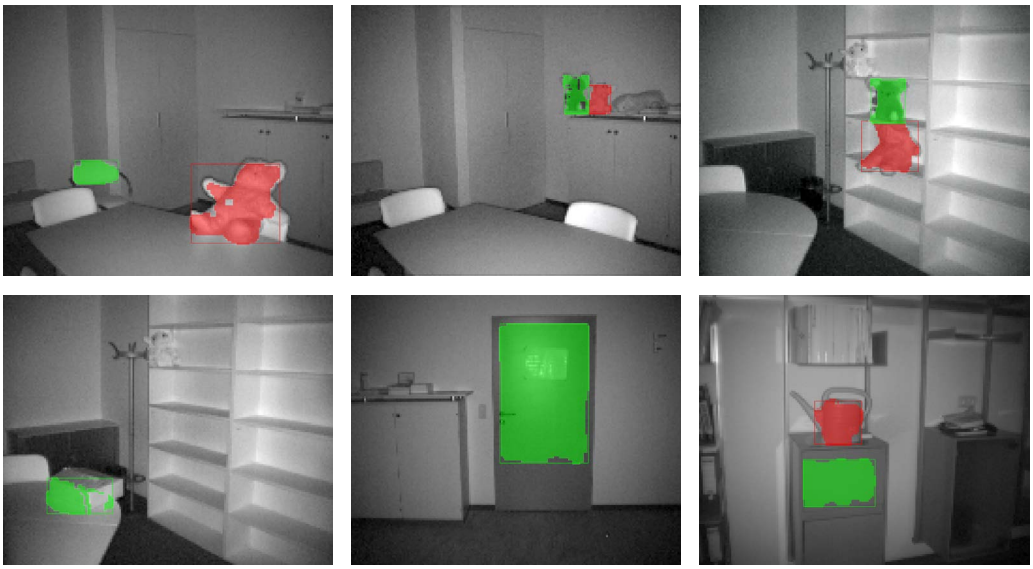


Figure 5.17: Examples of a wide range of segmented objects in different test sequences. The segmentation is based on detected movable object points of the articulated scene model. These points are separated using the points' appearance frequencies. Different colored areas belong to different movable objects. The object masks are projected on the corresponding amplitude image showing the scene.

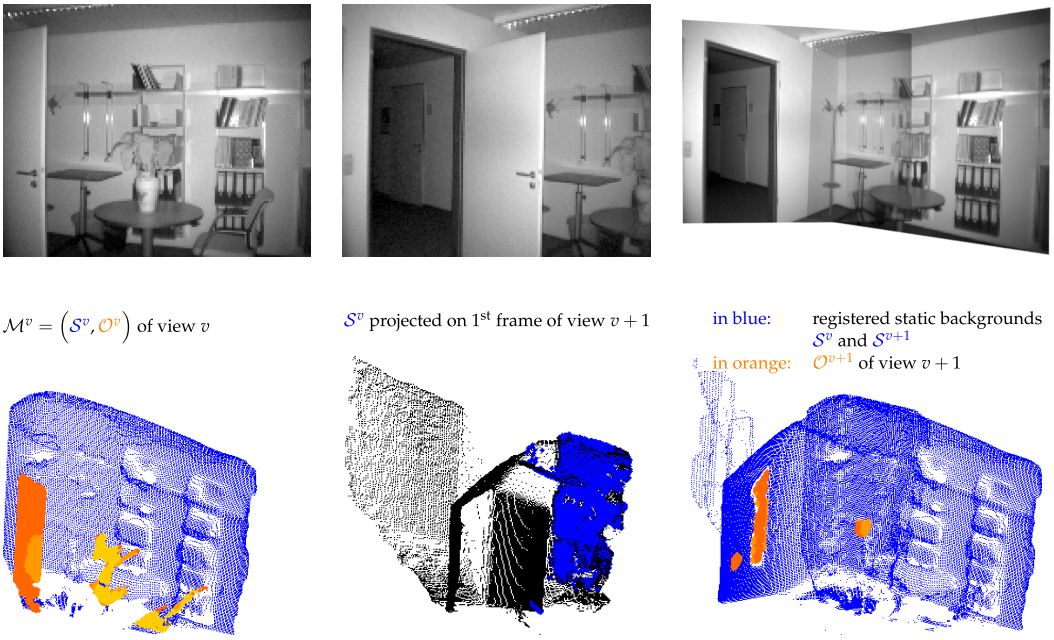


Figure 5.18: The final static background $\mathcal{S}^v$ of the articulated scene model $\mathcal{M}^v$ acquired for view v is projected on the 3D points acquired from view $v+1$. They initialize the scene modeling for the new view. Finally, the registered backgrounds, $\mathcal{S}^v$ and $\mathcal{S}^{v+1}$, are shown together with the movable objects $\mathcal{O}^{v+1}$ of view $v+1$.

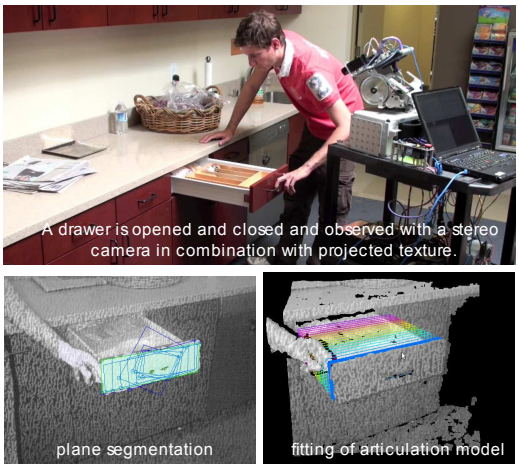


Figure 5.19: This eye catcher is taken from [Stu10]. It shows a fitting of a kinematic model to the trajectory of a planar patch. This patch is the front of a drawer which has been observed during opening and closing.

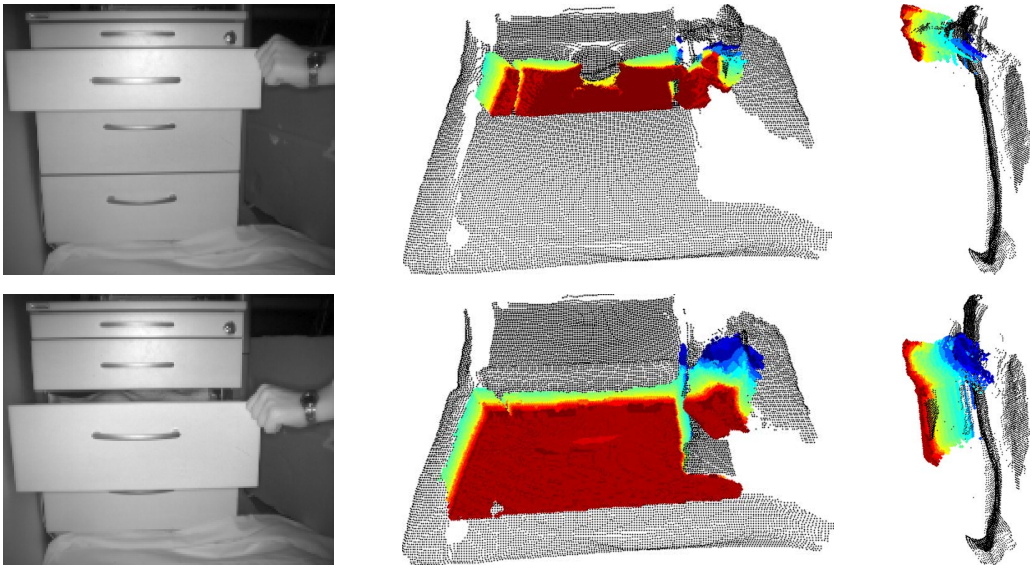


Figure 5.20: In the observed scenario two drawers are opened and closed. Here, the accumulated positions of the drawers can be seen. The color indicates the position of a drawer at a certain point in time. In time step t, $\mathcal{O}_t$ provides the drawer in its current position. From left to right: the amplitude images illustrate the scenario, the front view and the side view of the 3D point cloud show the drawer articulations. The black points assemble the static scene.

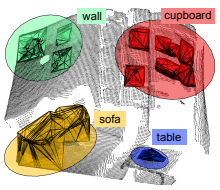
5.6 CONCLUSION AND OUTLOOK

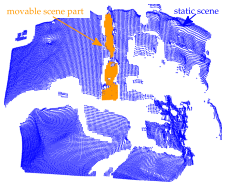
This chapter has focused on scene changes observed from a certain view point. The robot monitors a scenario where, e. g., a human moves around and performs actions which alter the environment like relocating chairs or opening and closing doors. Such scenes are divided into a static background which does not change, entities which are moving actively around, and movable objects that change their position passively. The articulated scene modeling approach offers a methodology for extracting these three scene elements from a short sequence of observations using only few general assumptions. Moving persons can be tracked reliably using the feedback of the static background and a cylinder as weak entity model. The static background is estimated by always adapting it to the farthest distance measurements. Movable objects can be detected if they are placed in front of a known static background. The distinction between tracking motion and detecting changes is apparent in the introduction of different processing methods and a separated encoding in the model. It follows the definition of Rensink [Ren02] and the insights about brain areas specialized either on motion or on change. My results show that using the farthest measurements to build the static background from range sensing reconstructs it reliably. Further, tracking is improved if static background is incorporated. Last, arbitrary movable objects are detectable if they have been positioned somewhere in front of the estimated background. So far, objects cannot be detected if the background behind is not known.

As the articulated model provides for a specific scene information about articulated parts it would be interesting to combine it with an aligned scene model of this specific scene (acquired from a scene description using the algorithms described in Chapter 4). Connecting described and articulated structures will enhance both models. The aligned model can extract from the articulated model the degree of freedom of some model structures. Knowledge about type and amount of these articulations can facilitate the fitting of a learned aligned model to a new situation. The articulated model could benefit from the aligned scene model by utilizing the possibility of automatic label inference for movable objects. Another possible extension of the articulated model could be a mechanism for learning a group of movable objects from a single example. Here, objects of the same type like chairs around a conference table or doors of a huge cupboard are of special interest. The goal is to extract all chairs or doors as movable objects although the human tutor has opened only one door or has moved only one chair for demonstration. Further, it would be interesting to develop methods for transferring the articulated model from one scene to another. Section 5.5.2 shows how this can be done if the same room is observed by rotating the camera. The question is whether it is possible to match a model to a view of an unknown room which has the same room type as the initially observed room (estimating the type of a room from its spatial layout as introduced in Chapter 3).

6

SUMMARY



This thesis has dealt with the question, how to provide semantic scene information for 3D percepts of indoor scenes. As scene analysis is a wide field, I have concentrated on developing models for the vista space of a domestic robot. This means that the models deal with scene percepts taken from one view point during a "home tour" scenario where a robot is shown around in an apartment and taught relevant spatial knowledge. Much work can be found where outdoor scenes are analyzed in 2D, whereas indoor scenes are analyzed by vision approaches mostly in 2D and by robotic approaches mostly in 3D. So far, the developed 3D methods have focused on the large-scale space of indoor scenes where locomotion is required to perceive the complete scene. My work contributes to the analysis of the less examined indoor percepts from the vista space. The scenes are apprehend-able from one view point without the necessity for locomotion. In the context of the "home tour" scenario, such scenes are views of a single room for which my analysis provides detailed semantic information. In particular, a holistic, an aligned, and an articulated representation of a room is learned. The holistic model specifies the type of the room like "living room" (Chapter 3). The aligned model holds supporting structures that have been referred to like "a cup on the table" (Chapter 4). And the articulated representation models movable scene parts like a "cupboard door" by observing scene changes happening when the door is opened or closed (Chapter 5). **This thesis has proposed new scene modeling approaches that incorporate basic physical properties of vista scenes and psychological or psycho-linguistic knowledge about their representation in humans.** As one step towards a complete spatial awareness of a robot, these vista space representations can be combined with large-scale space representations, like navigation maps, due to their complementary nature. The following paragraphs summarize the proposed algorithms providing a holistic, aligned, and articulated modeling of a vista space scene and discuss future research directions.

THE HOLISTIC SCENE MODEL. The combination of physical characteristics of space and psychological findings about the perception of space is reflected in the *holistic* model by the fact that the recognition of the room type is based on the room's spatial layout. Man-made environments are mainly assembled by sets of planar surfaces and it is assumed that people can categorize a room quickly, because brain areas are activated that are sensitive to 3D geometry (see [Heno8]). The holistic scene model consists of Support Vector Machines models that have been trained on a newly defined 3D spatial feature vector. This feature vector is generated by computing histograms over values that encode the shape and the size of perceived patches and the angle and the size ratio between pairs of patches. It captures globally the spatial layout of a room given as a set of planar surfaces and encodes it independent from specific furniture or objects and the knowledge about interdependencies between objects and room types. For testing the performance of the 3D feature vector, we have compiled the probably largest 3D indoor database that is currently available. We have recorded with a SwissRanger camera rooms that are shown in the exhibition of the popular IKEA furniture stores. With this database I have shown that the 3D feature vector encodes information of an indoor room that is complementary to the information encoded by the well-known Gist feature vector introduced by Torralba [Toro3b]. This Gist vector encodes a scene based on edge information present in a 2D image and has also been developed to capture the global scene impression of this 2D image. If both global feature vectors, the newly defined 3D vector and the Gist vector, are fused by a voting scheme following the sum rule, good categorization results are achieved on percepts of indoor rooms taken from a robot's perspective. **It is now possible to categorize indoor percepts based on their spatial layout as it can be observed that room layouts have room type specific features.**

THE ALIGNED SCENE MODEL. The second type of scene representation introduced in this thesis is the *aligned* scene model. It is inspired by the fact that language will be successful in conveying space, because language and cognition schematize space in the same way [Tve98]. This means that a hearer can build from a description a model which is similar to the model the speaker has built from visual perception. Furthermore, interlocutors start to align their models of their visual representations when they talk to each other about the underlying scene [Pico4]. The goal is to equip a robot with capabilities enabling it to extract in a 3D perception of a scene the semantic structures that have been mentioned of the human partner. This ability is implemented in the aligned model by utilizing the effect of gravity on the scene layout and the construction principles of spatial scene descriptions. Basically, this means that every object has a supporting structure and that in the description objects are only related to their supporting structure or to other objects located on the same supporting structure. I have introduced a new terminology for these two relation types. The first one is named *orthogonal* relation and the second one *parallel* relation. The computation of meaningful spatial structures consists of three steps. First, rules are defined that transform a sequence of orthogonal and parallel relations into a set of trees reflecting the hierarchical characteristic of spatial descriptions. Second, poten-

tial planar patches are estimated as priors for supporting structures of objects sharing the same parent node. And third, these potential patches are fused with planar patches extracted in a 3D perception of the scene to resolve ambiguities in the descriptions. The resulting set of patches and their labels form a model that is aligned to the description of the scene. Tests on 30 descriptions of two different rooms have shown that the computed models meet the level of details in the provided depictions. The combination of model estimation from descriptions with bottom-up visual processing is clearly a new approach for modeling complex scenes. So far, models have been extracted automatically from visual percepts without involving the interlocutor or the partner's model is derived from expressions without grounding them in the visual world. **As hearer it is now possible to extract the speaker's scene representation guiding the given description and to ground this representation in its own perception.**

THE ARTICULATED SCENE MODEL. Last, the approach for generating an *articulated* scene representation aims on learning semantic scene structures from observation of spatial changes in a scene. It is inspired by the fact that children have a higher sensitivity towards detection of scene changes compared to adults [Thro2]. It seems that the ability to detect changes in situations is crucial for learning. In general, the term *change* is defined as variation of structure while the term *motion* refers to variation of location [Reno2]. Change can be either *dynamic* if the transformation itself is observed or *completed* if at some point the change of structure is perceived. The last one means phenomenologically that the detection of completed changes involves a comparison between a representation in memory and a representation of visible structures. The computational model developed in this thesis aims on detection of completed scene changes. In 3D perception this can be realized robustly by assuming that from a fixed perspective the farthest static measurements always determine the static scene background. The articulated scene model consists of three parts, namely, dynamic entities, movable objects, and the static background, that are estimated over a short sequence of frames. Knowledge from previous frames is used in the analysis of subsequent frames. Dynamic entities like the moving human are detected in each frame through a particle filter based on a weak cylinder model which is augmented by the knowledge about the static background. The background model is updated instantaneously in each frame to the farthest static depth measurements. Static depth measurements that appear in front of the known background define movable resp. articulated scene parts. **The contribution of this approach is a model-less detection of articulated scene elements through observing scene changes caused by their manipulation and a background learning under disturbed and short observation conditions.**

FUTURE WORK. Future research directions of the work presented here could be the refinement and extension of the individual models and the integration of the models in realistic human-robot interaction scenarios. Since the further development of each model has already been discussed in the conclusions of each chapter (see, Chapter 3.5, 4.6, and 5.6), I will focus at this point on the second direction. An integration of my approaches on a mobile robot faces two main questions:

> *How should the observation and the modeling period be structured so that a natural interaction between robot and human would become possible?*
>
> *How can a robot that moves around integrate its vista space and large-scale space representations?*

So far, it has been assumed that observation and modeling period can be clearly separated. But in an interaction scenario this assumption is not valid anymore. A closer coupling of observation and modeling period is necessary so that feedback can be generated and incorporated. The design of my models allows in principle an interleaving of the observation and modeling phase. The algorithms process one frame or spatial relation and integrate the results in the representations generated on prior data. Technically spoken, it is straightforward to generate and incorporate feedback on the basis of these intermediate models. However, the interesting question is how to design the feedback and how to integrate the response into the models. For example, while constructing the tree set of the aligned scene model a feedback can be generated after each processed relation. Here, the interesting question is what feedback should be generated:

> Is a confirmation feedback like "hmmm" or "I understand the cup is on the table" enough?
>
> Can a feedback be constructed that triggers a response from which additional meaningful information can be extracted like "what is the supporting structure for . . ."?

One could imagine that a scene representation can be constructed faster and more robust if the model formation is designed cooperatively because the robot asks proactively for information. Otherwise, one has to hope that information will be given implicitly so that it can be inferred. Models built with feedback will deviate from models built without feedback, as a robot asking questions may also influence the recipient's situation model. On the one hand, this influence could be positive as the human tutor is triggered to concentrate on relevant information. On the other hand, it could disturb the task instruction underlying the spatial description. This could be the case when the robot wants to get detailed information about a scene structure that is not important for executing a given task. In such situations, an explicit reparation instructions might be necessary like "forget about the sofa and concentrate on the table instead". Besides the interaction with a partner, the interaction with the environment is important for a mobile robot. Modeling scenes on the vista scale requires that the robot stands still during the observation. The question is how such a

requirement can be integrated into the behavior of a moving robot. As it would be computational too exhaustive to compute for each point that can be reached in an apartment a representation of the visible vista space, algorithms have to be developed that generate appropriate view points in an apartment. Optimization criteria could be driven by the task itself, e. g. [Zie10], or by the goal to generate as few or as informative views as possible. The memorization of models from several view points rises the question on the interplay between representations for the vista space and for the large-scale space. For example, a registration of vista space models could be seen as a representation of the covered large-scale space. The model resulting from this fusion would have the same level of details like the underlying vista space models. But for navigation purposes, it might be better to link the vista space models to a SLAM map which is a large-scale representation optimized for localization and navigation [Bee07]. In this context, a lot of interesting new research questions can be formulated:

> How much details do representations of the large-scale space need?
>
> Is it enough if appropriate vista space models are accessed when more details are required?
>
> How can information between different models be exchanged?
>
> And does the representation of the large-scale space influence the formation of models on the vista-scale and vice versa?

Answers to these questions will equip a robot with a spatial awareness that models the complete space in a flexible way with methods for adapting the representation to the current task.

A

APPENDIX – SCENE CLASSIFICATION

A.1 3D INDOOR SCENE CATEGORIZATION – A PROVE OF CONCEPT

This section shortly summarizes our first approach to the indoor scene categorization using planar surfaces as presented in [Swao8c]. It is a prove of concept that shows that it is possible to define a proper feature vector on the set of extracted planar patches.

FEATURE EXTRACTION. The following listing describes the first attempt for defining a feature vector that encodes the spatial layout of a scene given by a set of planar patches $\{\mathcal{P}_i\}$.

(i) *Number of Points per Patch.* First, a feature vector is computed that encodes the size of patches in a frame, e. g., whether it contains large patches or many small planar structures. For simplicity the size of a patch is estimated by the number of points assembling a patch:

$$\forall i : n_i = \frac{|\mathcal{P}_i|}{\sum_j |\mathcal{P}_j|}. \tag{A.1}$$

The resulting terms have values between zero and one with a concentration in the region close to zero. As a feature vector (FV1) a histogram is computed using bins of different size – small close to zero and becoming large towards one. More precisely, 6 bins are chosen with the boundaries according the following listing $[0, e^{-4.5}, e^{-3.5}, e^{-2.5}, e^{-1.5}, e^{-0.5}, 1]$.

(ii) *Angles between Normals of Patches.* Here, the orientation between patches is considered:

$$\forall i \neq j : \alpha_{ij} = \arccos(\vec{n}_i \cdot \vec{n}_j) \tag{A.2}$$

divided by the maximal possible value which is $\frac{\pi}{2}$. The feature vector (FV2) is created as a histogram with 5 bins uniformly distributed over the values between zero and one. The experiments have shown that for classification it is sufficient to compute the median of these angles to encode their information. Both, histograms over number of points per patch and angles between pairs of patches do not contain any structural information about the rooms. This information can be introduced by computing the feature histogram (FV3) over the angles α'_{ij} between pairs of close patches leading to better classification results.

(iii) *Ratios between Sizes of Patches.* This feature (FV4) encodes whether a frame contains many patches of similar or different size:

$$\forall i \neq j : r_{ij} = \frac{\min(|\mathcal{P}_i|, |\mathcal{P}_j|)}{\max(|\mathcal{P}_i|, |\mathcal{P}_j|)}, \tag{A.3}$$

while the feature vector (FV1) over the number of points per patch refers to the absolute sizes of the patches. The histogram here also consists of 5 bins between 0 and 1.

The values in the bins of the feature histograms (FV1, FV2, FV3, FV4) are normalized to the range $[0, 1]$ by dividing the entries by the sum over all values in the bins per histogram.

EVALUATION. This feature vector is evaluated on three room categories which can be found in a university. From two offices, two corridors, and two meeting rooms 300 frames have been recorded by the SwissRanger. The camera is positioned at the door frame and pans and tilts during recording. Figure A.1 presents the acquired rooms. The classifiers are trained with data from `office.1`, `seminar.1`, and `corridor.1`. The categorization of unknown rooms is test with frames from `office.2`, `seminar.2`, and `corridor.2` while the recognition of known rooms is tested with frames of the training rooms which have not been used for training.

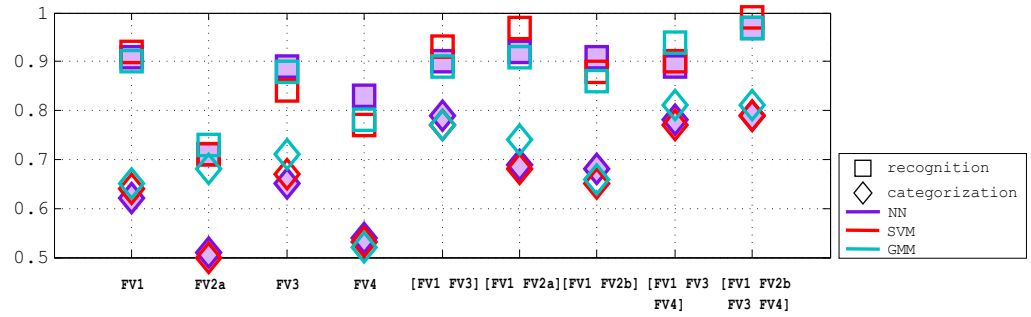
`office.1` `seminar.1` `corridor.1`

(a) Training rooms.

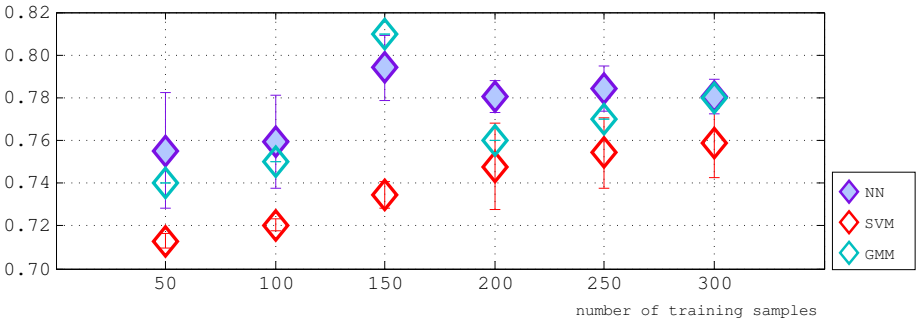
`office.2` `seminar.2` `corridor.2`

(b) Test rooms.

Figure A.1: Photos of the rooms scanned by the SwissRanger camera. (a) Data from these rooms is used to train classifiers. (b) Data from these rooms is used to test the categorization performance of the learned models.



(a) Recognition and categorization results.



(b) Influence of training samples on classification results.

Figure A.2: (a) This plot presents results of the recognition (□) and categorization (◊) using different combinations of the feature vectors (FV1, FV2, FV3, FV4: FV2a the histogram of angles between all patches, FV2b the median of these angles). Three classifiers are tested: a Neural Network (NN), a Support Vector Machines (SVM), and a Gaussian Mixture Model (GMM). (b) This plot shows the influence of the number of training samples on the categorization result using [FV1 FV2b FV3 FV4]. The vertical bars encode the standard deviation of the rates over 10 training runs with identical parameters for NN, SVM or GMM.

Three different classifiers are used to examine the proposed features: a Neural Network (NN) with one hidden layer based on the Neural Network Toolbox of MatLab using back-propagation [Rum86], the support vector machine SVM^{light} (SVM) [Vap95, Joa02] with a 5th-degree polynomial, and a Gaussian Mixture Model (GMM) with five mixed distributions implemented in the toolkit ESMERALDA [Fin99]. Screening experiments have provided five mixed distribution for GMM and a 5th-degree polynomial for SVM as quite suitable to deal with the proposed feature vectors. The examined feature vectors are the number of points (FV1), the angles between patches (FV2a) and the median over these angles (FV2b), the angles between close patches (FV3), and the ratio of number of points between pairs of patches (FV4). The features are tested separately and in combination. The training phase is based on 270 frames of each room from the training set (Figure A.1(a)). The recognition of known rooms is tested with the remaining 30 frames per room. The categorization of unknown rooms is tested with the 900 frames assembling the three test rooms of the test set (Figure A.1(b)). The shown rates are averaged over the three room types and 10 test runs.

Figure A.2(a) presents all classification results from different feature vectors and combinations of them. The first four columns examine the feature vectors FV1, FV2a, FV3, and FV4 separated from each other. FV1 and FV3 turn out as features which contribute most to a good feature vector (recognition rate: 0.90, categorization rate: 0.65). The combination of these two features (FV1 and FV3) leads to a feature vector which provides promising categorization results of up to 0.79 and recognition results of up to 0.93. An additional test is executed to study the influence of FV2a compared to FV2b. FV2b performs similar to FV2a. Therefore, it is assumed that the median of all angles can replace a histogram over all angles. The categorization can be improved up to 0.81 if the feature vector FV4 is added while the recognition rate stays on the level of 0.90. This rate can be further increased to 0.99 using FV2b. As an assumption it can be stated that GMMs provide the most stable and proper classifiers using [FV1 FV2b FV3 FV4] as a feature vector. Round about 75% of the false classified vectors are due to a mix up between meeting room and office. Since both room categories have analogies like a large table area in the middle of the room, this is an expected result.

Figure A.2(b) shows the influence of the amount of training data on the classification rates. The vertical bars encode the variance of the classification rates over the 10 test runs. It can be noticed that especially the NN and GMM classifiers seem to become saturated if more than 150 training samples are used. This leads to the conclusion that acquiring 300 frames per room provides a data set from which general room models can be learned.

Eighty percent of successful room categorization indicates that these planar structures extracted from 3D point clouds provide meaningful information about categories of rooms whereon feature vectors can be defined suitable for classification. The categorization only based on the given 3D data provides promising first results that may be even further improved via applying more different statistics to the set of planar patches, like, e. g., histograms over shapes of patches.

A.2 EQUIVALENCE OF FORM FACTORS FOR 2D BOXES

For an arbitrary 2D patch a standard factor for encoding the shape is defined by

$$c^{U,A} = \frac{U^2}{4\pi \cdot A} \tag{A.4}$$

where U is the outline of the region and A the covered area. For a box with s being the short edge and l being the long edge my shape characteristic computes as follows:

$$c^{s,l} = \frac{s}{l}, \quad \text{with } s < l. \tag{A.5}$$

In the following, I am going to prove the equivalence of these two shape factors

$$c^{U,A} \sim c^{s,l}. \tag{A.6}$$

This will be done by proving the existence of a bijective function that maps $c^{s,l}$ on $c^{U,A}$.

Given a 2D box with the following characteristics:

$$\begin{aligned} s &\quad \text{short edge} \\ l &\quad \text{long edge} \\ U &= 2s + 2l \\ A &= s \cdot l \end{aligned}$$

x and y are set to

$$x = \frac{s}{l} \tag{A.7}$$

$$y = \frac{(2s+2l)^2}{4\pi \cdot sl} \tag{A.8}$$

and y reformulated in such a way that it becomes a function of x

$$\begin{aligned} y = f(x) &= \frac{(2s+2l)^2}{4\pi \cdot sl} = \frac{1}{4\pi} \cdot \frac{4s^2 + 8sl + 4l^2}{sl} \\ &= \frac{1}{\pi} \cdot \left(\frac{s^2}{sl} + \frac{2sl}{sl} + \frac{l^2}{sl}\right) = \frac{1}{\pi} \cdot \left(\frac{s}{l} + 2 + \frac{l}{s}\right) \\ &= \frac{1}{\pi} \cdot \left(x + 2 + \frac{1}{x}\right). \end{aligned} \tag{A.9}$$

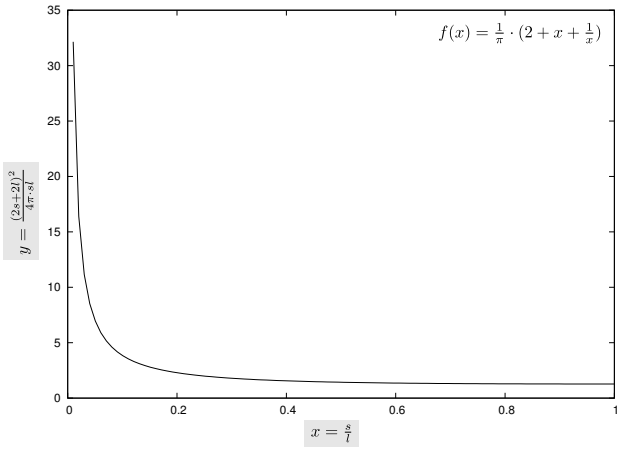


Figure A.3: Figure shows plot of function $f(x) = \frac{1}{\pi} \cdot (x + 2 + \frac{1}{x})$ with $x \in]0, 1[$.

As x can only take values between 0 and 1, $f(x)$ needs only to be considered in the range of $]0, 1]$. Figure A.3 shows the plot of the function $f(x)$. The derivation of $f(x)$

$$f'(x) = \frac{1}{\pi} \cdot \left(1 - \frac{1}{x^2}\right) \tag{A.10}$$

gives that $f(x)$ has for the interval of $x \in]0, 1]$ only values ≤ 0. This means that $f(x)$ is decreasing strictly monotonic on this segment and is therefore a suitable function to map $c^{U,A}$ and $c^{s,l}$ onto each other in a bijective way.

B

APPENDIX – SCENE DESCRIPTIONS

This chapter lists all descriptions acquired in our studies. In two studies – a pilot study and a main study – 20 persons have provided in total 30 descriptions of two rooms presented as print-out or on a computer screen. In the following sections the original descriptions in German and their translation to English and a machine-readable format are given. The translations have been done manually. In the machine readable format

- – denotes a parallel relation,
- | the orthogonal relation "on", and
- o the orthogonal relation "in".
- 0 is used if an object is simply listed and no relation to any other object has been specified.

The descriptions of the playroom, scene $\mathcal{S}_1$, acquired during the pilot study are given in Section B.1 and acquired during the main study in Section B.2. The descriptions of the living room, scene $\mathcal{S}_2$, can be looked up in Section B.3. The Figures B.1, B.2, and B.3 show per subject the raw tree set computed if the rules for transforming spatial relations to trees as defined in Section 4.4.1 are applied on the given scene description.

B.1 PILOT STUDY: PLAYROOM

This section lists the transcriptions of the original descriptions given by 10 participants. The playroom has been presented as photo on the monitor. The participants have benn instructed to describe freely the picture.

Participant 1(p):

ok also ich sehe ein tisch (.) ähm zwei reGAle ein abgeschnittnes reGAL ich sehe eine LAMpe (.) ne TÜR noch einn stuhl und dann ähm (.) viele viele kuscheltiere die verteilt sind auf TISCH stuhl ähm (.) reGAle und dann noch bücher und spiele im regal und eine ROse (.) hm ja

OK, I see a table, two cupboards, a lamp, a door, and a chair. There are soft toys on the table and on the chair. There are books and games in the cupboard. And there is a rose.

table 0 0;	↓	door 0 0;	cupboard o books;
cupboard2 0 0;		chair 0 0;	cupboard o games;
cupboard3 0 0;		table | softtoy;	rose 0 0
lamp 0 0;		chair | softtoy;	

Participant 2(p):

gut (3) ähm ich sehe in diesem raum ähm (.) ein TISCH ein stuhl (.) eine lampe im HINtergrund ähm der stuhl steht ähm vor dem TISCH es ist mehr ein hocker als ein stuhl weil er keine lehne hat (..) ähm im hintergrund stehn äh drei reGAle (.) ähm das linke und dies ähm drei reGAle stehen direkt nebneinanda? (.) ähm di: das regal in der mitte ist in einem rot ton gehalten die rechts und links sind äh be:sch (.) ähm überall in diesem zimma sind stofftiere und anderes (.) spielzeug verteilt? ich sehe (.) ähm monopoli pusl BÜcher (..) ähm (.) kleine (.) AUtos (.) und ich sehe eine wase und eine blume in der mitte auf dem tisch (...) sonst ist das zimma sehr (...) KINDorientiert ausgerichtet würd ich sagn obwohl vielleicht dann au noch zu sehr AUFgeräumt und mehr seh ich eigntlich nich auf dem bild

In the room there is a table, a chair, and a lamp in the back. A chair stands in front of the table. In the back there are three cupboards, they are parallel to each other: in the middle stands the red one and the light ones on the left and right of it. Soft toys and toys are all over the room. I see monopoly (games1), books, and a car, and a rose on the table. Overall it looks like a room of a child though it is a bit too tidy.

room o table;	↓	room o cupboard2;	room o toy;
room o chair;		room o cupboard3;	games1 0 0;
room o lamp;		cupboard1 - cupboard2;	books 0 0;
table - chair;		cupboard2 - cupboard3;	car 0 0;
room o cupboard1;		room o softtoy;	table | rose

Participant 3(p):

okej also ich sehe möbelstücke n TISCH n stuhl . drei reGAle eins davon is ROT zwei sind WEIss (.) der tisch un der stuhl is HOlzfarbn un hintn in ner ecke is ne LAmpe s sieht n bisschen nach KINderzimmer aus weil auf dem tisch mehrere PLÜSCHtiere liegn (.), da liegt äh son kleiner GREMlin (.) n koALAbär n teddibär n kleiner frosch aus plüsch und n känguru (.) n plüsch äh WÜRfel hintn ne wei ne gelbe rose (.) in einer wase un n kleiner spielzeugauto vor dem koalabär auf dem stuhl oder auf dem hocker isn plüschLÖwe glaub ich auch un davor n kleiner ROboter (.) un im regal sind BÜcher spiele ä:hm (...) das is ein obelix ein fred feuerstein un danebn son RA:be (...) un dadrunter keine ahnung ich glaub so (... ähm keine ahnung was das für kuscheltiere sind und in dem weißen regal sind auch spiele un n KERzenständer un darüber n kleiner PLÜSCHhund ne (..) kleine SCHAle ja (..) das seh ich in dem raum (.) und n TEPpich boden (.) weiße wände

I see furniture, a table, a chair, and three cupboards, one red and two white. Table and chair are light and a lamp is back in the corner. It looks a bit like a playroom because several soft toys lie on the table, more detailed gremlin (Stitch), a koala bear, a Teddy bear, a small frog, and a kangaroo, a cube with a rose behind it and a small toy car in front of the koala. On the chair is a lion and before the lion a small robot. In the cupboard are books, games, Obelix, Fred Feuerstein, and besides Fred a raven. Below the raven are objects which in don't now (Pokemon). In the white cupboard are also games, a candle, and above the candle a dog and a bowl. Hm, I also see a light carpet and white walls.

corner o lamp;	↓	chair | lion;	raven - pokemon;
table | softtoy;		lion - robot;	cupboard3 o games;
table | rose;		cupboard2 o books;	cupboard3 o candle;
koala - car;		fred - raven;	candle - dog

Participant 4(p):

der raum sieht aus wie ein KINderzimmer ähm im linken hinterGRUND sieht man reGAle davon sind zwei Weiß und eines rot (...) ähm man kann allerdings nicht alle reGAle komplett sehn (...) von einem der weißen reGAle geht ein (.) TISCH aus (..) der rechts äh ins bild ragt (...) auf dem tisch (.) findet man ähm verschiedene kuscheltiere und spielzeug (...) ganz RECHTS auf dem tisch sieht man einen koALA (..) BÄR (2) davor ist ein SPIELzeugauto (..) in der mitte auf dem tisch sieht man einen sehr ja einen sehr kleinen KÄNGuru (..) würd ich jetz sagn (..) ähm (..) dahinter sieht man eine WAse mit einer gelben ROse drinn ähm (2) in der hintersten ECKE des TISCHes (..) links (.) findet man ähm (3) auch ein kuscheltier ich weiß nich was es is (..) was das darstelln soll (..) dann ähm wiederum auf einer ECke (..) sieht man ein ähm (.) tedDIbärn ziemlich GROßn im verhältnis zu den anderen (..) und davor wiederum befindet sich ein FROSCH als kuscheltier (..) ähm (.) vor dem tisch im vordergrund des bildes sieht man ein hocker aus HOLZ (.) auf dem ist ein LÖwenkuscheltier zu sehn

und noch ein (..) weiteres (.) spielzeug eine art spielzeugROboter (2) ähm (..) in RECHten hintergrund (.) HINter dem TISCH so zusagen sieht man eine (.) STEHlampe (2) und ähm (..) nochweiter rechts davon eine TÜR (2) im regal finden sich wiederum kuschelTIERE (.) und ähm BÜcher (..) und BRETTspiele (3) ja (2) ich glaub darauf geh ich dann nich mehr näer ein (4) der boden ist (2) HELL (2) be:sch (...) und der TISCH ist aus HOLZ (..) ja das wars

The room looks like a playroom. To the left at the back you can see cupboards, two white and one red. However, not all cupboards are completely visible. In front of the white cupboard stands a table. It fills the right part of the image. On the table diverse soft toys and other toys can be found. Right on the table you can see a koala, in front of the koala a car, and in the middle of the table a small kangaroo. Behind the kangaroo stands a vase with a yellow rose. Left on the table you can find Stitch and a big Teddy bear and in front of the bear lies a frog. In front of the table, in the foreground of the image, stands a chair with a lion and a robot on it. Behind the table is a lamp and right of the lamp a door. In the cupboard are soft toys and books and games. The floor is light and the table is wooden.

room o cupboard1;	↓	table ! koala;	bear - frog;
room o cupboard2;		koala - car;	table - chair;
room o cupboard3;		table ! kangaroo;	chair ! lion;
cupboard3 - table;		kangaroo - rose;	table - lamp;
table ! softtoy;		table ! stitsch;	lamp - door;
table ! toy;		table ! bear;	cupboard o softtoy

Participant 5(p):

Alles klar also ich sehe (.) ein raum in dem MÖBEL sind (..) ähm ZWEI beziehungsweise ein angerissenes all ZWEI regale und ein ANerissenes ein tisch und ein HOCker (.) und ähm (.) ich würde vermuten kinderspielzeug verschiedene KUSCHLtiere (..) ähm SPIEle BÜcher (..) n AUto (.) ja kleine gegenstände kleine spielsachen (...) oder soll ichs detaHIERter beschreiben (.) gut dann (...) wer ich damit FERtich

I see a room with furniture – two cupboards, a table, and a chair. Further, I see toys, several soft toys, games, books, and a car.

room o furniture;	↓	toy 0 0;	books 0 0;
room o table;		softtoy 0 0;	car 0 0
room o chair;		games 0 0;	

Participant 6(p):

hm also ich sehe (3) zwei reGAle (2) wandreGAle ähm ein TISCH ein HOcker (.) eine lampe (.) stehlampe in der ecke ähm (.) dann ähm (..) verschiedene kuscheltiere he (lacht) (.) ein koALA ein BÄR daneben STITSCH und ähm ein

löwen auf dem hocker (..) auf dem hocker ist auch ein ähm (..) ro:boter (..) ähm (.) ja also neben oder vor dem teddibärn sitzt ebenfalls ein FROSCH (.) und neben dem frosch (.) naja etwas weiter HINtn sitzt ein (1) KÄNGuru (..) BÄR (lacht) ähm ebenfalls ne BLUMwase auf dem tisch mit ner gelben ROse ähm (.) und ein WÜRfel (..) ähm (.) in den WANDreGAln (1) ist ein RA:BE also das eine wandregal ist ROT das andere ist WEIß in dem ROten wandregal sitz ein RA:be dann äh (2) äh herr (...) heißt der FEUerstein (.) feuerstein und Obelix von ASterix und obelix (.) also der erste (..) dann (2) auf dem allerersten einlegeboden dann auf dem zweiten einlegeboden äh (.) ist ähm POkemon mit das andre tier weiss ich nicht was das ist (.) dritter ähm im DRITten einlegeboden liegn bücher (.) vierten einlegeboden zwei spiele (.) fünften ist gar nichts (3) ähm (2) das WEIße WANDregal wiad zur hälfte vom tisch verdeckt aber das was ich sehen auf dem ersten einlegeboden ähm ist ein HUND da ist ne SCHAle und ähm (..) ähm (.) paar zeitschriften (..) zweiter einlegeboden ähm (..) ist n KERzenständer und auch wieder SPIEle (.) und im dritten (.) ja glaub ich ist ne TASse (2) ne mit wasser drin (.) un in den beiden andern seh ich gar nicht (.) aber ich glaub da ist GAR nichts drinn (..) oKEJ (3) hm (2) da ist ne TÜR (5) kabel (...) von ner lampe (..) hm vor dem koala isn rennauto auf dem TISCH (2) und neben den ROten wandregal also links daneben ist NOch ein weißes also is (...) RECHTS ein weißes und LINKS ein weißes vom roten wandregal (2) ja (3) das wars

I see two cupboards, a table, a chair, a lamp in the corner, and several soft toys. A koala lies next to Stitch and a lion lies on the chair. On the chair is also a robot. A frog sits in front of the bear and next to the frog a kangaroo. Further, a vase with a yellow rose and a cube stand on the table. In the red cupboard (cupboard2) is a raven, Fred Feuerstein, and Obelix. On the second shelf sits Pokemon, on the third shelf lie books, and on the fourth shelf are two books. In the first shelf of the white cupboard (cupboard3) is a dog, a bowl, and some papers. On the second shelf is a candle and some games. And on the third shelf I see a cup with water. There is also a door and a lamp. In front of the koala lies a car on the table. Left to the red cupboard (cupboard2) stands a white cupboard (cupboard1).

koala - stitsch;	↓	cupboard2 o raven;	cupboard3 o cup;
chair | lion;		cupboard2 o pokemon;	table | koala;
chair | robot;		cupboard2 o books;	koala - car;
bear - frog;		cupboard2 o games1;	cupboard1 - cupboard2;
frog - kangaroo;		cupboard3 o dog;	cupboard2 - cupboard3

Participant 7(p):

ähm ja ich sehe viele KUSCHeltiere (.) ähm son DECKenfluter (..) n tisch und n STUHL drei reGAle (.) ähm ja noch weitere SPIELzeuge son kleiner ROboter (.) n auto ähm (..) mehrere spiele unter naderem moNOpoli un das labüRINT ähm BÜcher gibt es noch (.) ein KERzenständer mehrere bekannte figuren also als kuscheltiere wie STITSCH (..) ähm den Obelix und fred FEUerstein (.) ähm der ra:be das is so ne handpuppe die hab ich nämlich auch die kenn ich he

(lacht) (.) hier is noch POkemon (.) ähm vieh (.) glaub ich (.) ähm ja da is noch ne WAse aufm tisch mit ner BLUme .. u:nd ich glaub das is n BAUklötzjen (2) ja und SPIELzeugauto is noch da (4) joa (2) DIE schale (..) weiß ich nich ob ich die schon genannt hab seh ich noch (...) und (2) ja also außer diese kuscheltiere (...) ach das isn frosch son QUELLfrosch der hat ne KUgel im bauch den kenn ich (.) den hab ich AUch (.) hehe (lacht) ja ansonsten glaub ich hab ich alles genannt (...) oder soll ich EINzelne sachen noch aufzähln (...) kann ich (.) ja dann (.) sieht man halt noch den koALAbär (..) daoben das is glaub ich n hund (lacht)(..) n teddibär und n LÖwn (...) u:nd ähm ja stitsch hab ich schon (...) das WARS

I see a lot of soft toys, a lamp, a table, a chair, three cupboards, further toys, a small robot, a car, several games, and books. Further, there are a candle, Stitch, Obelix, Fred Feuerstein, a raven, and Pokemon. There are a vase with a flower, a cube, and a car on the table. Further, I see a bowl, soft toys, and a frog with a bowl in the stomach which I have by myself. I also can a koala, a dog, a bear, a lion, and Stitch.

softtoy 0 0;	↓	car 0 0;	table | rose;
lamp 0 0;		games1 0 0;	table | cube;
table 0 0;		books 0 0;	table | car;
chair 0 0;		candle 0 0;	bowl 0 0;
cupboard1 0 0;		stitsch 0 0;	frog 0 0;
cupboard2 0 0;		obelix 0 0;	koala 0 0;
cupboard3 0 0;		fred 0 0;	dog 0 0;
toy 0 0;		raven 0 0;	bear 0 0;
robot 0 0;		pokemon 0 0;	lion 0 |0

Participant 8(p):

(7) ja der raum sieht aus (.) wie ein e praxiszimma würd ich sagn (.) könnte auchn SPIELzimma sein (.) in dem raum befindet sich eine LAmpe (1) die auch wie ich so erkenn kann beLEUchtet also AN ist (.) dann ist (..) in diesem raum ein WEIßes und ein ROtes regal zu sehn mit mehreren fächern (..) in dem roten regal (.) stehn zum beisbiel verschiedne PLÜSCHfigurren drinn in den ersten beiden OBERen fächern (.) in den dritten fach von unten liegen BÜcher in verschiednen GRÖßen (..) u:nd (.) WEIterhin befinden sich dort (.) auch SBIEle (.) in dem roten schrank (.) moNOpoli zum beisbiel (..) in dem weißen schrank der da direkt neben steht befinden sich under anderm AUch sbiele ne PLÜschfigur KERzenSTÄNder un diwerse andere gegenSTÄNde von dem regal befindet sich ein TISCH dort steht einmal eine ROse mit (.) einer BLUMwase so wie ich erkenn kann (.) mehrere fiGURN sind auf dem tisch äh plüschfigur sind auf dem tisch AUFgebaut das is zum beisbiel ein koALAbär (.) wo vor dem koalabär ein (.) AUto vorne steh vorWEGsteht im hinteren bereich befindet sich ein plüschtier was ich nich weiter beschreibn kann sieht BLAU aus sieht LUStig aus (.) denn liecht da noch eine art (.) weiß nich kleiner gegenstand der ein bisschen quadratisch is (.) ein kleines KÄNGuru befindet sich da ein größerer BÄR mit ähm ja einen FROsch im VORdergrund also frosch als PLÜSCHtier

im vordergrund (.) sitzt auch noch auf dem tisch wobei dieser große bär (.) ein bein nach unten runter hängn lassn hat ja des weiteren seh ich noch ein HOcker auf dem (.) ein LÖwe quer ein eine PLÜSCHfigur ein LÖwe quer da liecht und im voordergrund kann nich genau erkenn was das ist (.) da steht auch noch ne fiGUR das kann (..) keine ahnung irndwie son track sein oder das könnte auch irndwien roboter sein (.)

Well, the room looks like a waiting room in a praxis but could also be a playroom. In the room is a lamp illuminated. In the room is a white (cupboard3) and a red cupboard (cupboard2) with several shelves. Several soft toys stand in the first and the second shelf of the red cupboard (cupboard2) and books of different size lie in the third shelf. Further, there are games like Monopoly (games1) in the red cupboard (cupboard2). In the white cupboard (cupboard3) next to the red one are also games, soft toys, a candle, and other objects. In front of this cupboard stands a table. On the table is a rose in a vase and other soft toys, e. g., a koala. In front of the koala stands a car. In the back of the table Stitch, a cube, a kangaroo, and a big Teddy bear are positioned on it. In the front of the Teddy bear is a frog and the Teddy bear is on the table. Further, I see a chair with a lion on it and a robot ahead of the lion.

room o lamp;	↓	cupboard3 o softtoy;	table | stitsch;
room o cupboard2;		cupboard3 o candle;	table | cube;
room o cupboard3;		cupboard3 - table;	table | kangaroo;
cupboard2 o softtoy;		table | rose;	bear - frog;
cupboard2 o books;		table | softtoy;	table | bear;
cupboard2 o games1;		table | koala;	chair | lion;
cupboard3 o games2;		koala - car;	lion - robot

Participant 9(p):

(2) in diesem raum stehn an der wand drei regale (1) ähm RECHts neben den regaln (.) steht eine STEHlampe? (..) in der mitte befindet sich ein TISCH auf dem verschiedene STOFFtiere zu finden sind (.) sowie eine ROse? (..) hmmm vor dem tisch steht ein HOcker auf dem (.) ein STOFFtier liegt und ein (..) ROboter steht (.) hmmmm in dem regal befindn sich auch noch verschiedene stofftiere ein paar BÜcher un zwei SPIEle (2) in den RECHten reGAL (..) steht eine schale (.) ein STOFFtier verschiedene SPIEle und ein ke:rzenstender (2) hmm (2) ja äh mm am rechten bildrand sieht man noch ne TÜR

In this room three cupboards stand at the wall. On the right side of the cupboards stands a lamp. In the center of the room is a table with several soft toys and a rose on it. In front of the table is a chair with a soft toy and a robot lying on it. In the cupboard (cupboard2) are diverse soft toys, some books, and two games (games1). In the right cupboard (cupboard3) is a bowl, a soft toy, various games (games2) and a candle. At the right image border a door can be seen.

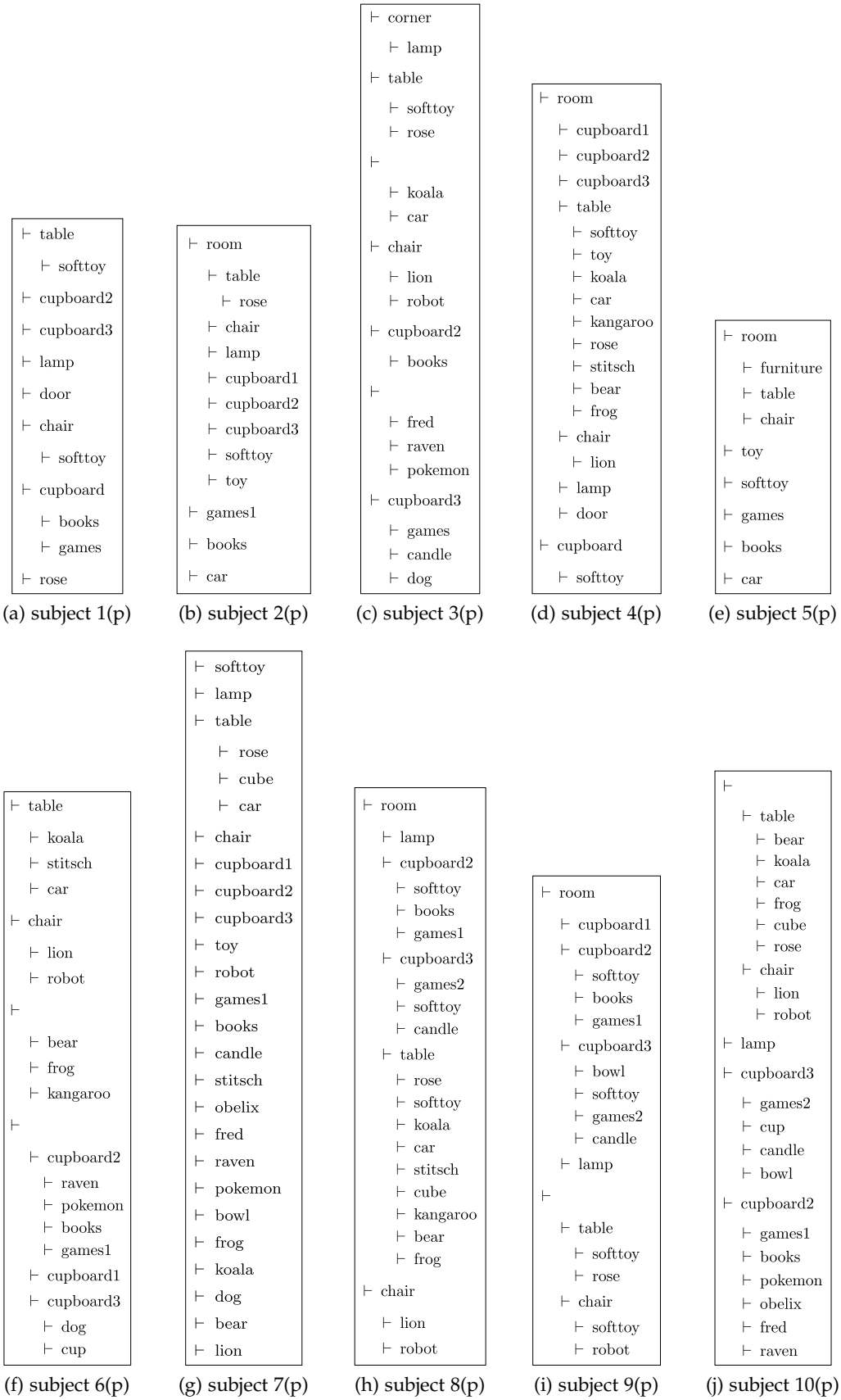
room o cupboard1;	↓	table - chair;	cupboard3 o bowl;
room o cupboard2;		chair | softtoy;	cupboard3 o softtoy;
room o cupboard3;		chair | robot;	cupboard3 o games2;
cupboard3 - lamp;		cupboard2 o softtoy;	cupboard3 o candle
table | softtoy;		cupboard2 o books;	
table | rose;		cupboard2 o games1;	

Participant 10(p):

(3) aalso zuerst einmal sehe ich äh auf dem rechten teil des bildes eine tür vermutlich die EINgangstür des (.) raumes vermut ich einfach mal so meistens gibts nur eine TÜR im raum (..) s is hier auch so (.) dann ham wir einen schönen viereckigen HOLZtisch (.) also viereckig . steht auf vier beinen drauf sind drei TEDDis (..) einer in blau der extremst heßlich is (..) blau (.) weiß ich nich was das sein soll ein normaler be:r den man eh kennt das andere könnte wohl ein äh (..) koALA bär sein joa (.) ein rennauto dann ham wir noch ein frosch ein WÜRrfel (...) ein GRÜN würfel (.) ähm (.) ne WAse mit ner gelben ROse (...) und wenn man in den raum reinkomm will also von mir aus dann RECHts oben (.) wenn du reinkommst (.) auch rechts is ein DECKenfluter ich glaub ein DECKenstrahler deckenFLUter (.) eine lampe die den raum erhellt undnnoch eine lampe die den tisch beleuchten könnte (.) dann ham wir drei schrankteile (.) wobei das (.) hintere nicht mehr zu erkenn is (.) ein schrankteil is hell dort befinden sich hm im (..) ein zwei drei vierten regal von unten vier SPIEle (.) man kann ein typischen spielehersteller erkennen (.) leider kann ichs nich lesn (.) äh drunter im regal sieht man noch ne tasse danebn (.) neben dem spiel steht ein kerzenstender ohne kerze (.) darüber ham wir dann ein weiteres großes spiel was sein KÖNNte danebn is ne schale in dem (.) ROten regal links daNEbn (.) da ham wir untn auch spiele von einem spielhersteller (..) monopoli pokamon (.) scheint als wenns (.) ein SPIELzimma wäre (.) hm das is das labyrinth der (..) ringe (.) kenn ich gar nicht labürinth der ringe GUT dadrunter??? sind noch einmal bücher dann noch ein (.) pokemonfigur (...) das weiß ich nur weil ich das gerade gelesn habe (.) un ähm auch das glaub im fernsehn mal gesehn hab darunter??? ham wir nen ASterix und ähm (2) wie heißt der typ denn noch (.) fred FEUerstein genau (...) danebn noch ein riesign vogel (.) schwarzen vogel mit gelben FÜßen und gelbem schnabel s gibt auch eine fernsehsendung in deutschland wo son ähnlicher RAbe war aber ich weiß nicht mehr ob der wie der rabe nun hieß ja es gibt jedenfalls sind das kuscheltiere und wahrscheinlich (.) aber ich hab keine ahnung wie dieser rabe hieß (.) ich glaub der siebnstein wars SIEBnstein SIEBnstein ja (.) was ham wir noch in diesm zimma zu guter letzt ham wir noch unter dem tisch oder vor dem tisch so eine art hocka (.) oder podest auf dem ein kuscheltierlöwe steht und ein mir nicht erkennbares (.) ROboterähnliches (..) DING keine ahnung was das sein soll (..) ich hab es gesehn ja (.) im schrank also ansonsten ist das is das ein sehr sehr FARBloser raum find ich die wand WEIß (.) der boden is be:sch (.) ge:sch gelb braun (.) also ähm (.) bis auf den roten schrank ist eigntlich alles sehr trist (...) könnte ein spielzimma oder kinderzimmer sein aufgrund der kuscheltiere vielen kinderspiele (.) ja ich glaub ich hab alles

First, I see in the right part of the image a door supposable the entrance. Then, there is a beautiful wooden squarish table with four legs. On the table are three teddies – one in blue and extremely ugly (Stitch), one a normal bear, and one a koala bear –, a car, a frog, a cube, and a yellow rose in a vase. If you enter the room a lamp stands on the right side. It illuminates the room and the table. There are three cupboards where the last one (cupboard1) is only visible half. In the light cupboard (cupboard3) are games (games2) and below them is a cup and next to the games (games2) a candle and next to it a bowl. In the red cupboard (cupboard2) are games like Monopoly (games1) and below them Pokemon figures and next to them Obelix and Fred Feuerstein and next to Fred a raven. I have seen the raven on TV but I do not know exactly its name it could be Siebenstein. In front of the table is a chair where a lion and a robot are lying on it. The floor has a brown color and except from the red cupboard the room is quite dull. Due to the soft toys it could be playroom.

table o o;	↓	lamp o o;	games1 - pokemon;
table | bear;		cupboard3 o games2;	pokemon - obelix;
table | koala;		games2 - cup;	pokemon - fred;
table | car;		games2 - candle;	fred - raven;
table | frog;		candle - bowl;	table - chair;
table | cube;		cupboard2 o games1;	chair | lion;
table | rose;		games1 - books;	chair | robot



(a) subject 1(p) (b) subject 2(p) (c) subject 3(p) (d) subject 4(p) (e) subject 5(p)

(f) subject 6(p) (g) subject 7(p) (h) subject 8(p) (i) subject 9(p) (j) subject 10(p)

Figure B.1: These raw tree sets are generated from the descriptions about the playroom given during the pilot study.

B.2 MAIN STUDY: PLAYROOM

This section lists the descriptions collected during the main study where the participants have seen a picture of the playroom presented on a screen. They have been told to describe freely what they see in the room.

Participant 1(m):

Okay, das sieht aus wie 'n – ich glaub, das ist 'n Wartezimmer irgendwo. Da sind auf jeden Fall viele Kinderspiele und Kuscheltiere drinne. Und 'n, ach genau, also zwei Regale links an der Wand. Dann haben wir 'n Tisch ziemlich zentral in der Mitte, wo auch Kuscheltiere drauf sind, 'ne Blume. Davor steht 'n Hocker, wo auch 'n Kuscheltier drauf steht und irgendein anderes Ding. Hinten in der Ecke steht 'ne Lampe. Rechts ist 'ne Tür. Äh. Ja dann sind da eben, ne?, Labyrinth der ich weiß es nicht, Monopoly seh' ich im Regal stehen, 'n paar Bücher für Kinder, Kuscheltiere. Mehr noch? Okay. Links steht auch noch 'n Regal, da kann ich aber nicht reingucken, da sieht man nur 'n ganz bisschen von. Genau. Das Zimmer ist sehr hell; auffallend ist das rote Regal. Das passt da gar nicht rein. (Hm. Ja, ich glaub, ich bin fertig.)

Well, the room looks like a waiting room. In the room are a lot of soft toys. At the wall is a cupboard and in the center of the room a table with soft toys and a rose on it. In front of the table is a chair with a soft toy and a robot on it. In the corner stands a lamp. On the right is a door. In the cupboard are games, namely Labyrinth and Monopoly, (games1), some books, and soft toys. The room is pretty bright but the red cupboard seems to be out of place. (I think I am ready.)

room o softtoy;	↓	table | rose;	corner o lamp;
wall o cupboard;		table - chair;	cupboard o games1;
room o table;		chair | softtoy;	cupboard o books;
table | softtoy;		chair | robot;	cupboard o softtoy

Participant 2(m):

Okay. Da haben wir einen Raum. In dem Raum sind mehrere Regale an der Wand, eine Lampe, 'n Tisch und ein Hocker. In den Regalen liegen Spiele, Bücher und Kuscheltiere. Und auf dem Tisch haben wir ein Spielzeugauto, mehrere Kuscheltiere und eine Vase mit einer Blume. Und auf dem Hocker liegt ein Kuscheltier und davor ein ähm Spielzeugroboter, würde ich sagen. (Ja. Das war's soweit.)

We have a room. In the are several cupboards at the wall, a lamp, a table, and a chair. Games, books, and soft toys are lying in the cupboards. On the table are a car, various soft toys, and a vase with a rose. On the chair is a soft toy and a robot. That's all.

room o cupboard;	↓	cupboard o games;	table ∣ rose;
room o lamp;		cupboard o books;	chair ∣ softtoy;
room o table;		cupboard o softtoy;	softtoy - robot
room o chair;		table ∣ car;	
wall o cupboard;		table ∣ softtoy;	

Participant 3(m):

Ja, sieht aus wie 'n Kinderzimmer. Viele Kuscheltiere. 'n Tisch und 'n Stuhl. Bücherregale, in denen Spiele liegen. Im Hintergrund 'ne Lampe. Auf'm Tisch steht 'ne Rose. 'n Kerzenständer im Regal. Das mittlere Regal ist rot. 'ne Tür, die zugeklebt ist, scheinbar, die Scheibe. (Alles bis ins kleinste Detail?) 'n Würfel auf'm Tisch und 'n Plüschfrosch und der Teddy, der da drauf sitzt, hat ein Bein runterhängen. Auf'm Stuhl steht – liegt 'n Löwe und, keine Ahnung, sieht aus wie 'n Roboter. Im roten Regal steht noch Obelix und Fred Feuerstein, Pikachu und 'n Rabe. Ähm im weißen Regal daneben steht noch 'ne Schüssel und 'n Hund oder 'n Hase, Zeitschriften, Spiele, 'n Kerzenständer und zwei kleine Tassen hinter diesem blauen Viech auf'm Tisch. Ja, kleines Känguru auf'm Tisch und 'n Koalabär, vor dem ein Auto steht. Ja. In dem weißen Regal da am Rand steht noch irgendwie, weiß ich nicht, 'ne DVD oder CD. Hmhm. Ja.

It looks like a nursery. A lot of soft toys, a table and a chair, cupboards (cupboard2 and cupboard3) with games in them. In the back a lamp. On the table stands a rose. In the right cupboard (cupboard3) is a candle. The middle cupboard is red. The door is sealed. On the table is a cube, a frog, and a bear. On the chair lies a lion and a robot. In the red cupboard (cupboard2) sit Obelix, Fred Feuerstein, Pikachu (Pokemon), and a raven. Aside, in the white cupboard (cupboard3) are a bowl, a dog, books, games, a candle, and a small cup behind Stitch. Stitch is on the table. Well, on the table is a small kangaroo and a koala with a car in front of it. At the border of the image is a kind of white cupboard (cupboard1) maybe containing DVDs or CDs.

cupboard2 o games;	↓	chair ∣ robot;	cupboard3 o games;
cupboard3 o games;		cupboard2 o obelix;	cupboard3 o candle;
back o lamp;		cupboard2 o fred;	cupboard3 o cup;
table ∣ rose;		cupboard2 o pokemon;	stitsch - cup;
cupboard3 o candle;		cupboard2 o raven;	table ∣ stitsch;
table ∣ cube;		cupboard2 - cupboard3;	table ∣ kangaroo;
table ∣ frog;		cupboard3 o bowl;	table ∣ koala;
table ∣ bear;		cupboard3 o dog;	koala - car
chair ∣ lion;		cupboard3 o books;	

Participant 4(m):

In der Mitte steht ein Tisch, auf dem Spielsachen, Plüschtiere sind, ein Auto, eine Blumenvase mit einer Rose. Vor dem Tisch steht ein Hocker, auf dem auch ein Löwe liegt und ein Spielzeug, was auch immer das ist. Und hinter dem

Tisch an der Wand stehn äh Regale, in denen Plüschtiere und Spiele und Bücher liegen. Und daneben, so schräg irgendwie dahinter ist eine Stehlampe, hinter der Tür quasi. Ein Kerzenständer-Männchen vielleicht steht noch im Regal. Und 'ne Schale. Und er hat hellen Fußboden. (Fertig.)

In the center stands a table. On the table are toys, soft toys, a car, and a vase with a rose. In front of the table stands a chair with on lion and a toy on it. Behind the table at the wall are cupboards with soft toys, games, and books in them. Next to the cupboards is a lamp. Yet, in the cupboard is a candle and a bowl. It has a light floor. (Done.)

table | toy;
table | softtoy;
table | car;
table | rose;
table - chair;

↓

chair | lion;
chair | toy;
table - cupboard;
wall o cupboard;
cupboard o softtoy;

cupboard o games;
cupboard o books;
cupboard - lamp;
cupboard o candle;
cupboard o bowl

Participant 5(m):

In dem Raum befinden sich Spielsachen. Da ist ein rotes Regal mit einem schwarzen Raben und zwei Spiel äh zwei anderen Stoffspielfiguren. Darunter ein Hase und ein Frosch. Darunter befinden sich Bücher und darunter befinden sich Stühle ähm Spiele. Daneben ist ein weißes Regal. Ähm dort sitzt ein Hase, da ist eine Schale. Daneben sind rechts daneben Arbeitsmaterialien. Darunter befinden sich Spiele. Daneben ein Kerzenständer. Dann ist da ein Tisch in der Mitte des Raumes. Ähm darauf befinden sich auch Stofftiere, ähm eine Vase mit einer Blume darin, ein Auto. Davor ist ein Hocker. Darauf befindet sich ein Löwe. Davor befindet sich irgendein roboterartiges Spielzeug. Ja. Ähm und hinter dem Tisch befindet ja, steht 'ne Lampe. (Ich glaub das wa's.)

In the room are toys. There is also a red cupboard (cupboard2) with a black raven and two soft toys. Below the raven are the Pokemon figures and below them books and below them games. Next to the red cupboard (cupboard2) is a white cupboard (cupboard3). In it sits a dog, next to it is a bowl, and right to the bowl work material. Below the work material are games and next to the games is a candle. In the room is a table. On the table are soft toys, a vase with a rose, and a car. In front of the table is a chair. On the chair is a lion and in front of the lion is a robot. Behind the table stands a lamp. (I am done.)

room o toy;
cupboard2 o raven;
cupboard2 o softtoy;
raven - pokemon;
pokemon - books;
books - games;
cupboard2 - cupboard3;

↓

cupboard3 o dog;
dog - bowl;
bowl - workmaterial;
workmaterial - games;
games - candle;
room o table;
table | softtoy;

table | rose;
table | car;
table - chair;
chair | lion;
lion - robot;
table - lamp

Participant 6(m):

Ein Raum mit einem großen Holztisch in der Mitte, auf dem viele Plüschtiere sitzen: ein Teddy und ein Koala, ein Frosch und noch ein Koala und ein Löwe, der auf einem Holzhocker sitzt. Dann äh im Hintergrund drei Regale, eins weiß, eins rot, noch eins weiß. In dem sitzen auch Plüschtiere: ein Rabe, Asterix und Fred Feuerstein, äh zwei Spiele: PokÃ©mon-Monopoly und Labyrinth der Könige oder so? – der Ringe? Ähm drei Bücher, vier, fünf. Äh in dem weißen Regal sind auch noch 'n paar Spiele. Auf dem Tisch steht noch eine Vase mit einer gelben Rose und einem grüner Stoffwürfel und im Hintergrund ist eine Stehlampe aus gebürstetem Edelstahl. (Das war's.)

This is a room with a big wooden table where a lot of soft toys sit on it: a bear, a koala, and a frog. A lion sits on a chair. In the back are three cupboards, a white one (cupboard1), a red one (cupboard2), and a further white one (cupboard3). In the red cupboard (cupboard2) are soft toys: a raven, Obelix, Fred Feuerstein, games namely Monopoly and Labyrinth (games1), and some books. In the white cupboard (cupboard3) are further games. On the table is a vase with a yellow rose and a green cube. In the back is a lamp. (That's all.)

room o table;
table | softtoy;
table | bear;
table | koala;
table | frog;
chair | lion;

↓

back o cupboard1;
back o cupboard2;
back o cupboard3;
cupboard2 o softtoy;
cupboard2 o raven;
cupboard2 o obelix;

cupboard2 o fred;
cupboard2 o games1;
cupboard3 o games;
table | rose;
table | cube;
back o lamp

Participant 7(m):

Ja dieses Zimmer ähm, ja, wie fange ich da am besten an? Also, in der Mitte des Raumes steht ein Tisch. Das ist ein Holztisch, der ist quadratisch. Auf diesem Tisch sitzt rechts an der schmalen Seite ein Koalabär, vor dem steht ein rotes Auto. Und dann verteilen sich so nach links über den Tisch noch einige andere Gegenstände. Ähm ja. Links hinten an der langen Tischkante sitzt so 'n ja so 'n blaues Ungeheuer würd' ich's nennen, mit großen Ohren. Rechts daneben steht ähm eine gelbe Rose in einer Vase. Genau. In einigem Abstand davor liegt ein Würfel, mit der Zahl Drei oben. Wieder weiter nach vorn an der Tischkante liegt so 'n grünes Tier, ich glaub, das soll 'n Frosch sein. Ist relativ klein, und direkt links neben dem Frosch sitzt ein Bär. Genau, so. An der Tischkante. Genau, der sitzt eigentlich mit dem Rücken an an der Wand beziehungsweise an 'n pa an 'n paar Regalen, die da stehen und lässt das rechte Baum so von der langen Tischkante nach vorne runterbaumeln. Genau. Und so relativ mittig auf diesem Tisch steht noch 'ne ganz kleine Figur, ich glaube, das könnt' 'n Hase sein oder 'n Känguru, ich kann's gar nicht genau erkennen. Dann vor dem Tisch steht noch ein kleiner Hocker, auf dem liegt ein Stofftier, Löwe und, ja, so 'n Spielzeugroboter, der vor dem Löwen platziert ist. Genau. Links an der Wand, da, wo sich auch der Bär anlehnt, stehen drei Regale. Genau. Rechts hinter dem

Bär ist zuerst ein weißes Regal, in dem, ja, da – das hat eins, zwei, drei, vier, fünf, sechs Regalfächer. Einige sind leer und in anderen sind auch wieder einige, ja, Spielzeugartikel zu sehen. Dann links neben diesem weißen Regal ist ein rotes Regal. Das hat, so wie es aussieht, genauso viele Fächer, und da stehen auch wieder einige Spiele drin. Zum Beispiel 'n Monopoly-Spiel und 'ne Asterix-Figur ist da. Und 'n, ja, so 'n Rabe und noch einige andere Sachen. Genau. Und links neben diesem roten Regal ist auch wieder 'n Regal zu sehen. Aber da erkennt man nur 'n ganz ganz kleinen Teil, dass man da gar nicht mehr zu sagen kann. Genau. Dann, wenn man sich noch mal das erste Regal ganz rechts anschaut, das ich eben beschrieben hab', da steht 'n ganzes Stück weiter nach rechts zur Wand auch eine ja so 'n Deckenfluter, der auch noch eine Leselampe angeschlossen hat. Und die Lampe leuchtet in der Ecke. Genau. Und ganz im Bildhintergrund erkennt man noch eine Tür. (Ja, das war's.)

In the center of the room stands a table. It is wooden and squarish. On the table sits a koala with a red car in front of it. Some objects spread over the table. On the left part of the table sits the blue Stitch. Right to Stitch stands a yellow rose in a vase. In front of that lies a cube. On the table at its front part lies a green frog and left to the frog sits a bear. On the table in the center sits a small figure, a kangaroo. In front of the table is a chair. On the chair is a soft toy, a lion, and a robot which lies in front of the lion. On the left, at the wall are three cupboards. Parts of the cupboards are empty. In the white cupboard (cupboard3) are toys. In the red cupboard (cupboard2) are some games namely Monopoly (games1) and an Obelix figure. A lamp stands next to the right cupboard (cupboard3) and illuminates the corner. At the back of the image a door is visible. (Yeah, that's it.)

room o table;	↓	frog - bear;	cupboard3 - cupboard2;
table | koala;		table | kangaroo;	cupboard2 o games1;
koala - car;		chair | lion;	cupboard2 o obelix;
table | object;		lion - robot;	cupboard2 o raven;
table | stitsch;		wall o cupboard1;	cupboard2 o object;
stitsch - rose;		wall o cupboard2;	cupboard2 - cupboard1;
rose - cube;		wall o cupboard3;	cupboard3 - lamp;
cube - frog;		cupboard3 o toy;	corner o lamp

Participant 8(m):

Okay. Wir sehen ein'n äh Raum mit Regalen an der Wand und einem Tisch davor und einen kleinen Hocker. In einer Ecke steht ein Deckenfluter. Äh. Es macht den Eindruck, als wär's ein Kinderspielzimmer. Es sind sehr viele Stofftiere vorhanden. Es gibt einen Hund, einen Bären, ein'n Koala, einen blauen Koala – oder was auch immer. Äh einen Löwen, mehrere Spiele, die sich – diese ganzen Sachen verteilen sich über Regal und Tisch. Ja.

Okay. We see a room with cupboards at the wall and a table in front of them. Further, there is a chair in the room. A lamp stands in the corner. It seems to be

a nursery. A lot of soft toys lie around like a dog, a bear, a koala, Stitch, a lion, some games. The objects are spread over cupboard and table.

room o cupboards;	↓	bear 0 0;	cupboard&table | bear;
wall o cupboards;		koala 0 0;	cupboard&table | koala;
cupboards - table;		stitsch 0 0;	cupboard&table | stitsch;
room o chair;		lion 0 0;	cupboard&table | lion;
corner o lamp;		games 0 0;	cupboard&table | games
dog 0 0;		cupboard&table | dog;	

Participant 9(m):

Ja, ich fange an mit einem allgemeinen Überblick. Wir haben hier anscheinend eine Mischung aus Kinderzimmer mit Stehlampe, die modernes Design sind. Es gibt hier sehr viele Plüschtiere zu sehen. N – Vordergrund steht äh ein Tisch, ein ja Holztisch, äh auf dem äh äh ja eins, zwei, drei, vier, fünf Plüschtiere sind. Nämlich ein Koala mittlerer Größe, ein kleines Känguru, ein Frosch, mittelgroß, äh ein großer Teddy und äh Stitch von Lilo und Stitch, äh Copyright is by Disney, glaub' ich. Äh außerdem einen Plüschwürfel, Drei Oberseite nach oben; die Eins zeigt auf uns. Äh und eine gelbe Rose, die nicht ganz ins Bild passt. Aber egal. Dann haben wir außerdem noch ein Spielzeugauto, was vor dem – hier vor dem Koala auf der rechten Seite liegt. Ähm vor dem Tisch steht ein Hocker, und auf diesem Hocker ist ein Roboter äh und ein Plüschlöwe. Im Hintergrund sind drei Regale zu sehen, wobei das linke Regal sehr versteckt ist, also nur noch äh ganz ein kleiner Teil zu sehen. Äh links und rechts sind weiß, in der Mitte das ist rot. Äh ebenfalls gefüllt mit sehr vielen Plüschtieren, unter anderem einer Bauchsprechpuppe eines hm in Form eines Rabens. Dann, äh ich glaube, Barnie Geröllheimer, nee, doch nicht (Wie heißt denn der andere noch mal?) – Fred Feuerstein. Und Obelix. Äh das Fach da drunter ist re ebenfalls re is' relativ leer. Es steht nur Pikachu und (Ach verdammt, ich kann die ganzen Pokemon-Namen nicht mehr') – ja, noch ein PokÃ©mon. Äh im dem Fach wiederum darunter, das dritte von unten, äh sind mehrere Bücher auf der Querseite gelegt – auf die Querseite gelegt, man sieht nur noch die Bücherrücken, zum Beispiel "Die Welt der Tiere", um ein kleines Beispiel zu nennen. Da drunter wiederum befinden sich zwei Spiele, und zwar das Pokemon-Monopoly (was ich unbedingt mal spielen muss, weil ich es überhaupt noch nie was davon gehört habe) und das Labyrinth der Ringe? Ja gut. Ich kenn' nur das verrückte Labyrinth, ein Abklatsch von dem verrückten Labyrinth. Weiterhin ist das Fach sehr leer. Da drunter das Fach ist komplett leer, das unterste. Äh das weiße Regal rechts daneben sch äh ist gefüllt mit im obersten Fach einem Plüschhund. Äh rechts daneben irgendwelche Heftchen und links daneben eine grüne Schale mit orangem abgesteppten Rand. Äh da drunter sind wieder o mehrere Spiele äh oben rechts angelehnt. Ein äh Kerzenständer steht daneben und der rechts ist – Rest ist verdeckt von dem Bären, ebenso wie die weiteren Fächer. Man kann noch in einem Fach äh Schälchen entdecken. Ja. Äh der Boden, auf dem alles steht, ist ein heller, ich glaube, Holzboden mit feiner Maserung. Und im Hintergrund sehen wir noch eine Stehlampe mit Deckenfluter äh und zwei Drehreglern, die

zum Dimmen bestimmt sind. Ganz rechts im Bild ist eine Tür mit Glasscheibe. (Damit wäre die Beschreibung dieses Bildes beendet.)

I start with an overview. It seems to be a nursery with a lamp. Further, there are a lot of soft toys. In the front stands a wooden table with five soft toys on it which are a koala of mid-size, a small kangaroo, a mid-size frog, a big bear, Stitch from Disney, a cube, a yellow rose, and a car which lies in front of the koala. In front of the table stands a chair and on this chair is a robot and a lion. In the back are three cupboards where the left one (cupboard1) is hidden. The left and the right cupboard (Cupboard3) are white and the middle one (cupboard2) is red. The red cupboard (cupboard2) is filled with soft toys like a raven, Fred Feuerstein, and Obelix. Below Obelix are Pokemon figures and below them some books and below the books two games namely Monopoly and Labyrinth (games1). All other selves are empty. In the white cupboard (cupboard2) right of the red one (cupboard32) is a dog. Right of the dog are some booklets (work-material) and left of the dog a bowl. Below the bowl are some games and next to them a candle. The floor is bright maybe wooden, in the back stands a lamp, and at the right side of the image is a door with a glass. (So, the description is finished.)

front o table;
table | koala;
table | kangaroo;
table | frog;
table | bear;
table | stitsch;
table | cube;
table | rose;
koala - car;
table - chair;

↓

chair | robot;
chair | lion;
back o cupboard1;
back o cupboard2;
back o cupboard3;
cupboard2 o softtoy;
cupboard2 o raven;
cupboard2 o fred;
cupboard2 o obelix;
obelix - pokemon;

pokemon - books;
books - games1;
cupboard3 o dog;
dog - workmaterial;
workmaterial - bowl;
bowl - games;
games - candle;
back o lamp

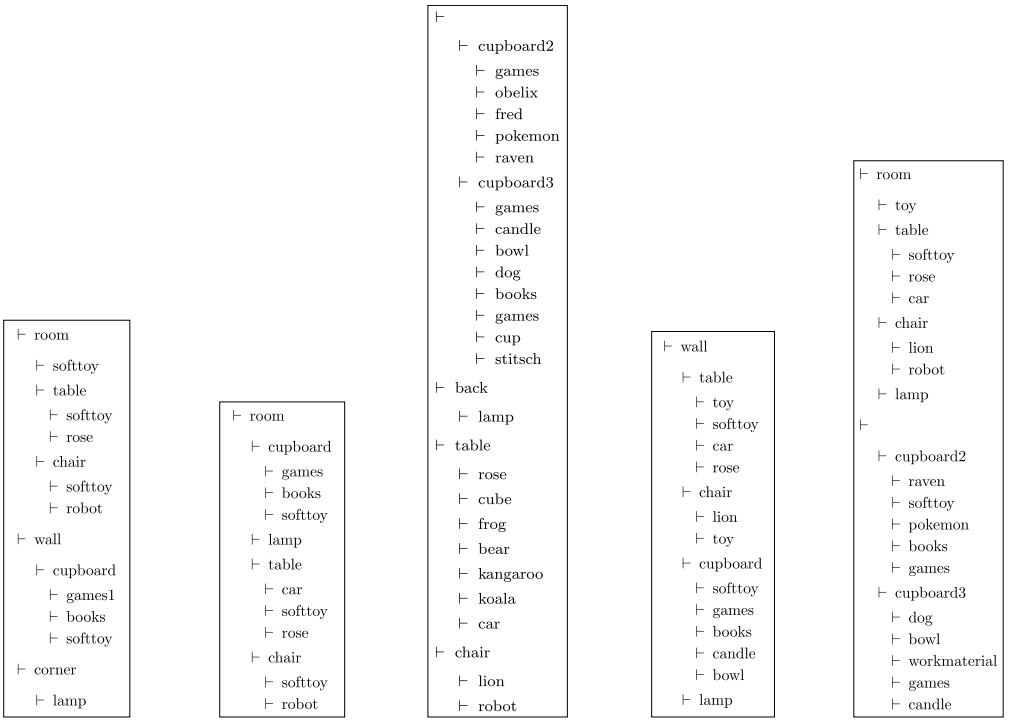
Participant 10(m):

Hm. In diesem Raum ähm gibt's 'n Holzfußboden, da drauf steht ein Holztisch. Davor steht ein Hocker, ebenfalls aus Holz. An der Wand links steht ein Holzregal, beziehungsweise mehrere. Das ganz linke, was man nur teilweise sieht, ist ähm hell, das mittlere ist dunkel, leicht rötlich, und das rechte von den dreien ist wieder hell, holzfarben, gleiche Farbe wie Tisch, Hocker und Fußboden. Äh weiter links daneben in der Ecke hinten steht 'ne Metalllampe, Deckenfluter mit 'ner – mit 'nem extra Arm dran für 'ne Tischbeleuchtung. Hm. Ganz rechts am Bildrand scheint 'ne Tür zu sein. Mit 'nem Glaseinsatz, der ist zugehängt mit 'nem Art Rollor oder, ja, irgend so was. Im Raum befinden sich hauptsächlich sowohl auf dem Tisch als auch auf dem Hocker als auch aufm Schrank irgendwelche Stofftiere. Auf dem Tisch sitzt zum Beispiel 'n Koalabär und 'nen anderer Teddybär. Auf dem Hocker liegt so was wie 'ne Löwe. Außerdem finden sich in dem Raum noch weitere Spielzeuge. In dem Schrank zum Beispiel steht ein Monopoly-Spiel und, na ja, irgend 'n anderes Spiel halt noch. Was sitzt da noch? Es sitzen noch 'ne ganze Reihe anderer Stofftiere überall rum. Im Schrank

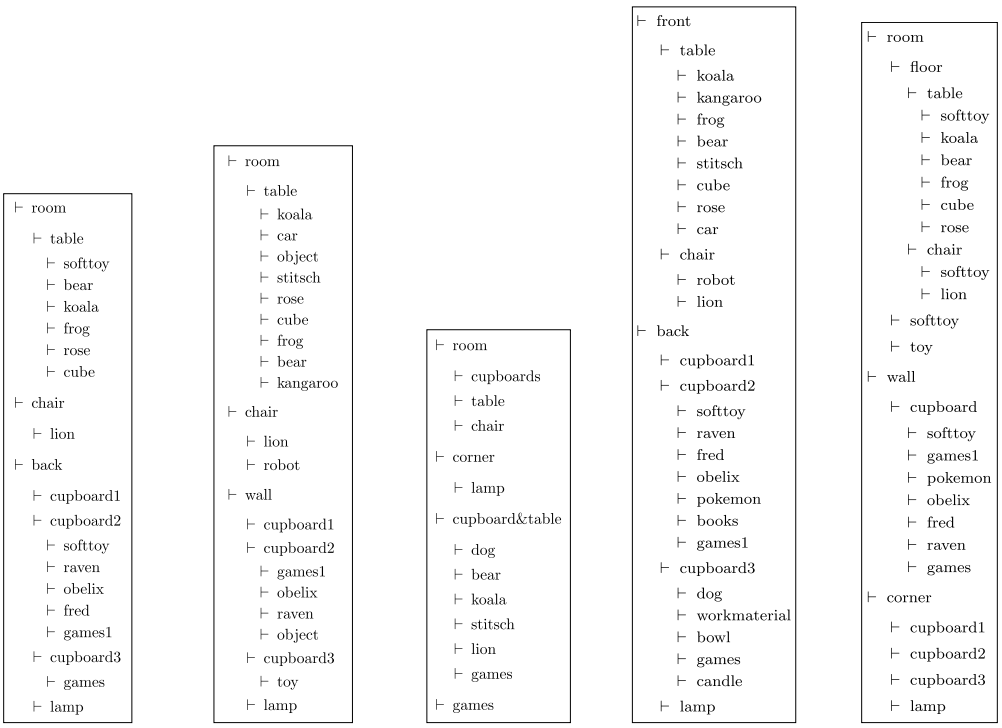
gibt's irgendwelche Pokemons, würde ich sagen. Ähm, Obelix sehe ich, Fred Feuerstein, irgendein Rabe sitzt da noch. Ah, da hinten sind noch 'n paar andere Brettspiele oder Gesellschaftsspiele. Hm ja. 'n Frosch liegt auf dem Tisch noch, 'n kleiner Würfel, kleineres anderes Stofftier, steht 'ne Vase mit 'ner Rose drin auch noch auf dem Tisch. (Und, ja, das war's.)

In this room is a wooden floor where a table stands on the floor. A chair is standing in front of the table also wooden. On the left side at the wall stand several wooden cupboards. The left one is bright (cupboard1), the middle on is dark (cupboard2), and the right one (cupboard3) is light and wooden like the table, the chair, and the floor. Further, a lamp stands in the corner with an extra arm to illuminate the table. On the right image border a door with glass is visible. In the room soft toys are mainly on the table, on the chair, and in the cupboard. A koala and a bear sit on the table. A lion lies o the chair. Additionally, further toys are in the room. For example, a Monopoly game (games1) is in the cupboard. What's else there? Several soft toys are sitting around. Pokemons are in the cupboard. I see Obelix, Fred Feuerstein and next to it a raven. Some further games can be found over there. A frog, a small cube and some other soft toys are lying on the table. Furthermore, a vase with a rose is standing on the table. (That's it.)

room o floor; ↓
floor | table;
table - chair;
wall o cupboard;
cupboard1 - cupboard2;
cupboard2 - cupboard3;
cupboard3 - lamp;
corner o lamp;
room o softtoy;
table | softtoy;
chair | softtoy;
cupboard o softtoy;
table | koala;
table | bear;
chair | lion;
room o toy;
cupboard o games1;
cupboard o pokemon;
cupboard o obelix;
cupboard o fred;
cupboard o raven;
cupboard o games;
table | frog;
table | cube;
table | rose



(a) subject 1(m) (b) subject 2(m) (c) subject 3(m) (d) subject 4(m) (e) subject 5(m)



(f) subject 6(m) (g) subject 7(m) (h) subject 8(m) (i) subject 9(m) (j) subject 10(m)

Figure B.2: These raw tree sets are generated from the descriptions about the playroom given during the main study.

B.3 MAIN STUDY: LIVING ROOM

This section lists the descriptions collected during the main study where the participants have seen a picture of the living room presented on a screen. They have been told to describe freely what they see in the room.

Participant 1(m):

Hm. Das sieht aus wie ein Wohnzimmer, würd' ich mal tippen. (Da sind viel zu viele Sachen drauf.) Äh, da steht eben auch in der Ecke 'n Eckregal mit ganz viel Tinneff drinne: mit Büchern, auch Kuscheltieren, 'ner Puppe, 'ner Musikanlage, teilweise mit so Glasfronten verdeckt, mit schönen Blumen drauf. Unten steht ganz viel Alkohol im Regal. Dann steht da links 'n großes gemütliches rotes Sofa neben, wo auch zwei Kissen drauf sind und 'ne Puppe. Davor steht ein Ikea-Tischlein mit Deko drauf, mit irgendwelchen Zitronen – nee, gar nicht – Orangenschalen, getrockneten. Im Hintergrund ist äh hängen auch zwei Bilder. Einmal eins mit irgendwelchen Bären in Kanada oder so. Daneben hängt 'n Bild mit, ich glaube, Vögeln drauf. Und darunter hängt noch 'n Fächer. Dann steht da auch, ist da auch so 'ne Stehlampe hinter dem Sofa. Und ich sehe zugezogene Gardinen. Rechts ist 'n Fenster, links ist wahrscheinlich auch 'n Fenster, wenn da Gardinen vorhängen. Und rechts ist noch 'ne Heizung, ist 'n Teppich. Sieht aus wie 'n Wohnzimmer. (Fertig.)

It looks like a living room. (There are too much things in it.) There is a cupboard in the corner with a lot of objects in it like: books, soft toys, a doll, a radio. Part of it is covered with flowered glass. The cupboard contains at the bottom a lot of bottles with alcohol. On the left side you can find a big red sofa with two pillows and a doll on it. In front of the sofa is table with a bowl on it. Two pictures hang in the back. One shows bears in Canada (picture1) and the other next to the first picture (picture1) shows some birds (picture2). A fan hangs below the second picture (picture2). A lamp stands behind the sofa. On the left and the right side are windows with closed curtains. Further, there is also a heater and a carpet.

corner o cupboard;	↓	cupboard o alcohol;	table | orangebowl;
cupboard o object;		cupboard - sofa;	back o picture1;
cupboard o books;		sofa | pillow1;	picture1 - picture2;
cupboard o softtoy;		sofa | pillow2;	picture2 - fan;
cupboard o doll;		sofa | doll;	sofa - lamp
cupboard o radio;		sofa - table;	

Participant 2(m):

Hier haben wir auch einen Raum. Ähm. In der Ecke des Raumes steht ein Regal. Das Regal hat ähm mehrere Glastüren. Hinter den Glastüren sind Gläser, Flaschen, CDs und Tassen. Ähm. Im Regal selbst sind noch Bücher, Bilder, eine Uhr, ein Radio, äh eine Puppe und mehrere Kuscheltiere. Und rund um das Regal ist eine Lichterkette. Vor dem Regal liegt ein Teppich mit einem Tisch

da drauf und auf dem Tisch ist eine Schale mit getrockneten Orangenscheiben. Ähm links daneben steht ein rotes Sofa mit zwei großen Kissen und einer Puppe. Und hinter dem Sofa haben wir eine Stehlampe. An der Wand hängen noch zwei Bilder und ein Fächer. Ja.

Here, we have a room. In the corner of the room is a cupboard. It has several doors made of glass. Behind the doors are glassware, bottles, CDs, and cups. In the cupboard are books, pictures, a clock, a radio, a doll and several soft toys. Fairy lights are mounted around the cupboard. In front of the cupboard lies a carpet with a table on it and a bowl is placed on the table. Left of the table is a red sofa with two large pillows (pillow1 and pillow2) and a doll on it. Behind the sofa is lamp. Two pictures (picture1 and picture2) and a fan are hanging at the wall.

corner o cupboard;	↓	cupboard o softtoy;	sofa |o pillow2;
cupboard o books;		cupboard - carpet;	sofa |o doll;
cupboard o picture;		carpet | table;	sofa - lamp;
cupboard o clock;		table | orangebowl;	wall o picture1;
cupboard o radio;		table - sofa;	wall o picture2;
cupboard o doll;		sofa |o pillow1;	wall o fan

Participant 3(m):

Ja, als Erstes sticht einmal das rote Sofa ins Auge, auf dem zwei gestreifte Kissen liegen und eine Puppe in Blau gekleidet mit Mützchen. Der Raum an sich ist weiß gestrichen. Es hängen zwei Bilder an der Wand, eins mit Bären im Fluss, und das andere könnten Vögel im Baum sein. Dann hängt da noch 'n Fächer an der Wand und hinter dem Sofa an derselben Wand steht noch 'ne Stehlampe. 'ne grau-weiß gestreifte Gardine links und rechts vom Bild. Ähm. Auf'm Fußboden liegt 'n Teppich, 'n beiger. Mit'm Bei mit'm Tisch drauf, in dem 'ne Schale mit Orangenschalen drauf steht. Und hinten an der Wand in der Ecke steht ein großes Regal mit Vitrinentüren. Da hängt 'ne Lichterkette drüber. Und in dem Regal stehen diverse Deko-Sachen und Kuscheltiere, Bücher, 'ne Uhr, 'n Radio, Flaschen, so 'ne kleine Minibar, CDs, Geschirr auch, ja, der Kölner Dom, glaub' ich. In diesem komischen Bild. Dann 'n Foto in einem Bilderrahmen. Ganz oben steht noch 'ne Kerze. Ja. Ja.

First, you see a red sofa with two stripped pillows (pillow1 and pillow2) and a blue doll lying on it (doll1). The room is painted in white. Two pictures are hanging on the wall one shows bears in a river (picture1) and the other birds in a tree (picture2). A fan is also hanging at the wall and a lamp stands behind the sofa. At the left and right side of the picture stripped curtains frame it. On the floor lies a beige carpet. A table with a bowl on it stands on the carpet. Back at the wall stands a big cupboard with cabinet doors in the corner. The cupboard is framed with fairy lights. Several decorative objects and soft toys, books, a clock, a radio, bottles, CDs, dishes, and a box can be found in the cupboard. Further, a funny picture (picture3) and a candle are in the cupboard.

sofa | pillow1;	↓	wall o lamp;	cupboard o softtoy;
sofa | pillow2;		floor | carpet;	cupboard o books;
sofa | doll1;		carpet | table;	cupboard o clock;
wall o picture1;		table | orangebowl;	cupboard o radio;
wall o picture2;		wall o cupboard;	cupboard o picture3;
wall o fan;		corner o cupboard;	cupboard o candle
sofa - lamp;		cupboard o decoration;	

Participant 4(m):

Das sieht aus wie 'n Wohnzimmer mit einer knallroten Couch, auf der 'ne Puppe sitzt. Hinter der Couch ist 'ne kleine Stehlampe. An der Wand hängen Bilder und ein Fächer. Und in der Zimmerecke steht ein Regal mit allerhand Kram drin. Und vor dem Sofa steht ein Tisch mit 'ner Schale drauf. Und auf'm Boden liegt ein heller Teppich. Und an den Fenstern sind Vorhänge, grau gestreift. (Fertig.)

It looks like a living room with a red sofa which has a doll on it. Behind the sofa is a small lamp. At the wall are pictures and a fan. In the corner stands a cupboard with a lot of objects in it. A table with a bowl on it stands in front of the sofa. A light-colored carpet is lying on the floor. Gray-stripped curtains are hanging in front of the windows.

livingroom |o sofa;	↓	wall o fan;	table | bowl;
sofa | doll;		corner o cupboard;	floor | carpet
sofa - lamp;		cupboard o object;	
wall o picture;		sofa - table;	

Participant 5(m):

Okay, wo soll ich anfangen? Also, das ist äh ein Raum. An der Wand hängt ein Bild, auf dem sind Bären zu sehen. Daneben ist ein weißes oder 'n ein Bild mit weißem Rahmen und zwei Vögeln, glaube ich, ich kann's nicht genau erkennen. Ähm davor befindet sich eine Lampe. An der Wand unter dem Bild hängt noch ein Fächer. Ähm an der hinteren Wand ist ein großer Schrank äh mit drei Fächern und – (Soll ich dir das alles sagen, was da drin steht?) Hm in dem ersten Teil ist Gesch – äh oder sind Gläser, Geschirr, vermute ich mal. Daneben sind Fotoalben, eine Kerze. Darunter sind auch Bücher oder Fotoalben. Daneben ist noch ein Stofftier, ein Elch. Darunter befindet sich ein Bild, darunter ein Weihnachts-Teddybär. Ähm darunter noch mal irgendwie Fotoalben oder Bücher und eine Kiste. Darunter – kann man nicht genau erkennen, ich glaub, eine Trommel. Dann ähm der rechte Teil des Schrankes: Oben ist ein Bild, äh ein Buch. Darunter befinden sich weitere Bücher, ein Wecker und eine Puppe. Darunter ein Radio – Kassettendeck, wie auch immer. Daneben ein Stofftier, ein Koalabär. Ist 'n, nee, ist 'n Koalabär? Pandabär. Dann unten ist ja noch mal so 'n so 'n Vitrinenteil mit ähm Geschirr, verschiedenem. Darunter ähm, kann nich' erkennen, ich vermute irgendwie Video-, DVD-, Kassettenhüllen, keine Ahnung.

Und unten im letzten Regal ähm befinden sich Flaschen. Dann ähm in der Mitte links des Raumes befindet sich ein rotes Sofa mit zwei Kissen. Darauf eine Puppe. Hm. Rechts daneben ein Tisch, darauf steht eine Schale mit Orangen, ja, so Orangen Orangen. Ähm. Am Boden befindet sich ein Teppich. Dann äh rechts im Raum sieht man noch ein Fenster beziehungsweise die Vorhänge. Darunter die – eine Heizung. Und links befinden sich auch noch mal diese Vorhänge. (Ja, ich glaub, das war's.)

Okay, where should I start? Well, it is a room. A picture showing bears (picture1) hangs at the wall. Next to it is a picture showing birds (picture2). Next to this picture is a lamp and below it at the wall a fan. At the backwall stands a big cupboard with three shelves – (Should I describe the content?). Well, in the first part (part1) are glasses. Next to them are albums and a candle. Below them are books and further albums and aside of the albums is an elk. Below the elk you can find a picture, below the picture a Christmas bear, and below the bear further books and a box. In the right part of the cupboard (part2) a book is placed on the top. Below this book are further books, a clock, and a doll. Below the doll is a radio and next to the radio a soft toy panda bear. The bottom of the cupboard contains dishes, CDs, and bottles. In the center of the room you can find a red sofa with two pillows (pillow1 and pillow2) and a doll on it. Well, on the right side of the sofa is a table with an orange bowl on it. A carpet is located on the floor. On the right side of the room is a window with a curtain and a heater below the curtain. On the left side of the room you can find the same curtain. (Well, that's it.)

wall o picture1; ↓
picture1 - picture2;
picture2 - lamp;
wall o fan;
picture2 - fan;
backwall o cupboard;
part1 o glasses;
glasses - albums;
glasses - candle;
candle - albums;
albums - elk;
elk - picture;
picture - bear;
bear - books;
part2 o books;
books - books;
books - clock;
books - doll;
doll - radio;
radio - panda;
room o sofa;
sofa |o pillow1;
sofa |o pillow2;
sofa | doll;
sofa - table;
table | orangebowl;
floor o carpet

Participant 6(m):

Also, ich sehe eine Puppe auf einem roten Sofa mit zwei großen rot-orange-gelb-grauen Sofakissen, ein äh Bild mit drei Braunbären drauf, einen Fächer, der an der Wand hängt. Über dem Fächer hängt ein Bild, ich nehme an, es sind zwei Vögel in einem Baum, auf jeden Fall mit grün und braun. Ähm unter den Braunbären ist eine Stehlampe mit weißem Schirm, klein. Dann gibt's da eine Schrankwand ähm mit einem Elchplüschtier drinne und einer Puppe und einer Uhr, ein'n Pandabär und diversen anderem Kram. Ähm. Ein Ikea-Tisch äh mit – (Wie heißt das noch gleich drauf?), ähm, Potpourri – Orangen-Potpourri. Vorhänge an den Fenstern und äh ein Heizkörper. Um den Schrank geht eine Lichterkette. Steht noch ein Bilderrahmen drin, ein Plüschteddy. Ähm, drei

Bücher, vier, fünf. (Ist das ausführlich genug oder noch ein bisschen mehr? Ja, ich würde sagen, im Großen und Ganzen wär's das.)

I see a doll on the red sofa with two big red-orange-yellow-gray pillows (pillow1 and pillow2). I see a picture showing bears (picture1) and a fan hanging at the wall. A picture showing birds (picture2) is hanging above the fan. Below picture1 is a lamp. Further, there is a cupboard with an elk soft toy, a doll, a clock, a panda bear, and diverse other objects in it. Further, there is an IKEA table with on orange bowl on it. You can see curtains at the window and a heater. Fairy lights are framing the cupboard. Moreover, the cupboard contains a picture frame, a teddy bear, and books. (Is it detailed enough? Well, then, that's it).

sofa | doll;
sofa | pillow1;
sofa | pillow2;
picture1 0 0;
wall o fan;
fan o picture2;

↓

picture1 - lamp;
cupboard o elk;
cupboard o doll;
cupboard o clock;
cupboard o panda;
cupboard o object;

table |o orangebowl;
cupboard o picture3;
cupboard o bear;
cupboard o books

Participant 7(m):

Also, in dem Raum steht vorne links im Bild ein rotes Sofa, auf dem liegen zwei gestreifte Kissen. Ja, die sind so rot-orange-weiß gestreift. Zwischen diesen Kissen sitzt eine Puppe mit langen blonden Haaren. Die hat einen blauen Pullover an und eine blaue Hose mit einem weißen Aufdruck und weiße Schuhe und auf dem Kopf trägt sie eine, ja, eine Mütze mit einem weißen Bommel. Genau. Hinter dieser Couch ähm steht links an der Wand eine Leselampe mit einem kleinen Lampenschirm. Der ist weiß. Genau. Hinten an der Wand hinter dem Sofa rechts von dieser Lampe ähm steht auch noch, ich glaub, das ist so was wie 'n Hula-Hoop-Reifen oder so. Man kann nur die obere Ecke erkennen, deswegen weiß ich's nicht ganz genau. Links von der Lampe sieht man noch 'n Stück Gardine. Die ist auch so grau-weiß gestreift. Rechts von der Gardine und über der Lampe ist ein Bild, auf dem sind ein paar Bären zu sehen und rechts von diesem Bärenbild ist noch 'n anderes Bild an der Wand, das ist in der Mitte grün. Ich denke auch, dass da Tiere drauf sind, das kann man aber nicht ganz genau erkennen. Und unter diesem Bild, das ich gerade beschrieben hab', hängt ein Fächer an der Wand. Genau. Rechts von dem Bild und dem Fächer in der Zimmerecke steht 'ne Vitrine. Ja, oder 'n Regal, ich weiß gar nicht, wie ich's benennen soll. Es hat zum einen ein paar Glastüren, zum anderen sind's offene Regalfächer, die da sind und n in diesem Regal steh'n halt ganz verschiedene Dinge: so Stofftiere und Bücher, noch 'n Radio, 'n Bilderrahmen, und auch 'n paar Alkoholflaschen und 'n bisschen Geschirr. Das ist ganz ganz gemixt, was da drin steht. Über diesem Regal hängt eine Lichterkette, sind, glaub' ich, ein paar Sterne dran befestigt, die leuchten könnten. Genau. Rechts von dieser Vitrine genau auf – an der Wand, auf die man so relativ frontal drauf guckt, ist rechts, wahrscheinlich vor einem Fenster, auch eine gestreifte Gardine. Genau. Die ist etwas kürzer als die Gardine am linken Bildrand, die geht nur bis über eine

Heizung und endet halt etwas höher. Und, wie gesagt, unter der Gardine ist eine Heizung. Genau. Auf dem Fußboden ist zum einen, ja, ich würd' sagen, das ist ein Holzfuß – Fußboden oder Laminat und vor dem vor beziehungsweise auch 'n Stück unter dem Sofa liegt ein, ja, zum größten Teil beiger Teppich. Dem sind so einige Ornamente, 'n paar ja rote Streifen, 'n paar orange Streifen und 'n paar blaue Kringel zu sehen. Genau. Und auf diesem Teppich steht ein viereckiger Holztisch, auf dem eine Glasschale steht und da sind ja so Deko-Orangen zu sehen. Ja. (Und mehr fällt mir zu dem Zimmer grad nicht ein.)

Well, in the room stands a red sofa where two striped pillows (pillow1 and pillow2) lie on it. They are red-orange-white striped. A doll with blond hairs (doll1) sits between the pillows. She wears a blue pullover, blue trousers, white shoes, and a bonnet. Behind the sofa left at the wall stands a lamp. Back at the wall, behind the sofa, and right of the lamp stands a hulahoop from which you can see only the upper part. Left of the lamp you can see a part of a curtain. It is gray-white striped. Right of the curtain and above the lamp is a picture showing bears (picture1) and right of picture1 is another picture which is green and shows animals (picture2). Below this picture (picture2) a fan is hanging at the wall. At the right of the fan and the picture in the corner of the room stands a cupboard. The cupboard has doors and is partially open. It contains diverse objects: soft toys, books, a radio, a picture frame (picture3), alcohol bottles, and glasses. The cupboard is framed with fairy lights. At the right side of the cupboard probably in front of the window a striped curtain is hanging and you can see a heater. On the floor lies a carpet and on the carpet stands a table with an orange bowl on it. (Can't think of more details for the room.)

room o sofa;
sofa | pillow1;
sofa | pillow2;
pillow1 - doll1;
doll1 - pillow2;
sofa - lamp;
wall o lamp;
wall o hulahoop;
sofa - hulahoop;
lamp - hulahoop;
lamp - curtain;

↓

curtain - picture1;
lamp - picture1;
picture1 - picture2;
wall o picture2;
picture2 - fan;
wall o fan;
fan - cupboard;
corner o cupboard;
cupboard o object;
cupboard o softtoy;
cupboard o books;
cupboard o radio;
cupboard o picture3;
cupboard o alcohol;
cupboard o glasses;
cupboard - curtain;
floor o carpet;
carpet | sofa;
carpet | table;
table | orangebowl

Participant 8(m):

Man sieht ein Wohnzimmer mit natürlich einem kaum auf-fallenden roten Sofa, äh einem Eckschrank mit Glasfront, äh einen kleinen Tisch, einen Läufer, äh ein paar Vorhängen und Bildern an der Wand. (Weitere Details, oder? Ah ja.) Auf dem roten Sofa sitzt eine blaue Puppe äh vor zwei Kissen. Dahinter steht eine kleine Leselampe. Das Regal ist gefüllt mit Stofftieren und Puppen, ein Radio, etwas Geschirr. Und auf dem kleinen Beistelltisch steht noch eine Schale mit Deko-Orangen, getrocknet. Nanu-Nana, drei-achtzig.

You can see a living room with a flashy red sofa, a cupboard, a small table, a carpet, curtains, and pictures at the wall. (Further details?) On the red sofa sits a blue doll (doll1) in front of two pillows (pillow1 and pillow2). Behind the sofa stands a lamp. The cupboard is filled with soft toys, dolls, a radio, and glasses. On the small table stands an orange bowl (from Nanu-Nana, three-eighty).

livingroom o sofa; ↓	wall o picture;	cupboard o softtoy;
livingroom o cupboard;	sofa | doll1;	cupboard o doll;
livingroom o table;	pillow1 - doll1;	cupboard o radio;
livingroom o carpet;	pillow2 - doll1;	cupboard o glasses;
livingroom o curtain;	sofa - lamp;	table | orangebowl

Participant 9(m):

Okay, wir haben hier einen Raum, in dem – nein, wir haben eigentlich eine Ecke von einem Raum, in dem eine rote Couch steht. Äh wir haben ein Regal, gefüllt mit Plüschtieren und Puppen, ähm 'n Stereorecorder. Äh außerdem gibt's dort Bilder zu sehen. Wir haben eine Wand außerdem, auch noch äh zwei eingerahmte posterähnliche Bilder. Ham einen Bär, drei Bären äh und äh einmal, was weiß ich denn, was, vielleicht so 'n paar Vögel, die, glaube ich, runter gucken. Kamera-Bild wurde nach oben aufgenommen in den Baum. So, auf der Couch sitzt äh 'ne Puppe und sind zwei Kissen zu sehen. Hinter der Couch äh ist eine Lampe, eine Leselampe, aber eine Stehlampe. Äh, dahinter an der Wand ist noch 'n Fächer. Ja. Vor der Couch steht ein kleiner Couchtisch mit getrockneten Orangenschalen. Es scheint Weihnachten zu sein. Äh wir haben außerdem noch äh is' noch die Heizung halb im Bild und ein zugezogenes Fenster mit einem lustigen Vorhang. Äh gut, ich könnt' jetzt noch weiter –. Ein Teppich liegt auf dem Boden. Könnt noch weiter auf das Regal, was in der Ecke steht, eingehen. Da ist eine L äh Leucht äh – (Wie heißen die noch mal?) hier, so 'ne äh so 'ne äh Leuchtschnur. So 'ne Lampendings (hm ja, weißt schon). Äh und äh die linke Seite ist zugezogen und eine Glasscheibe, die milchig ist. Äh mittig finden sich von oben nach unten äh Ordner oder Fotoalben mit dem Rücken äh zum zum äh Fotoaufnehmer, äh ein lustiger Geburtstagshut, ein Elch, Plüschelch mit weiteren Alben. Dann da drunter ein Foto von einer Person am Strand mit rotem Pulli, mehr kann man leider nicht erkennen, weil's zu klein ist. Äh dann kommt ein lustiger weißer Weihnachts-Teddy, äh gefolgt von äh einem Bild vom Kölner Dom wahrscheinlich, äh in Silber gehalten, und weiteren Buchrücken. Äh dann kommen wir zum rechten Teil des äh Regals. Dort ist die – das obere Fach fast leer, außer von einem – mit einem Buch, wo "Addams" drauf steht auf'm Rücken. Dann kommt eine Puppe ähm und äh eine Uhr sowie weitere Bücher mit Buchrücken äh zu sehen. Äh dann ein Pandabär mit und schon vorher erwähnte Stereoanlage, Ghettoblaster unten schön versteckt ist das TV äh unter – hinter ja ein weiterer Glasscheibe, die auch milchig ist, aber diesmal kann man 'n bisschen mehr durchgucken. Dort steht das Kaffee-Service von Oma. Dann CDs, ganz viele. Und Spirituosen. Ja. Ich glaube prinzipiell, bis auf den japanischen Fächer, der noch an der Wand steht, habe ich diesen Raum beschrieben.

Okay, I see a room. In the corner of the room stands a red sofa. I can see a cupboard filled with soft toys and dolls, a radio, and pictures. You can see a wall with two pictures at it, one showing bears (picture1) and one showing birds (picture2). On the sofa sits a doll and there are two pillows (pillow1 and pillow2). Behind the sofa stands a lamp and behind the lamp at the wall hangs a fan. In front of the sofa stands a small table with an orange bowl on it. It seems to be Christmas. Furthermore, a part of a heater is visible on the photo and the windows are covered with funny curtains. Well, a carpet lies on the floor and I can describe the cupboard in the corner in more details. It is framed with fairy lights and on the left side it is covered with a door. It contains albums, a candle, and an elk. Below the elk is a picture showing persons (picture3) and next to the picture a white Christmas bear with a box beside it. In the cupboard are books with a doll and a clock next to them followed by a panda bear and a radio. Behind the door dishes, CDs, and alcohol are visible. So except from the fan at the wall I have described the room exhaustively.

room o sofa;
cupboard o softtoy;
cupboard o doll;
cupboard o radio;
cupboard o picture;
wall o picture1;
wall o picture2;
sofa | doll;
sofa | pillow1;
sofa | pillow2;
↓
sofa - lamp;
lamp - fan;
wall o fan;
sofa - table;
table | orangebowl;
floor o carpet;
corner o cupboard;
cupboard1 o albums;
cupboard1 o candle;
cupboard1 o elk;
elk - picture3;
picture3 - bear;
bear - box;
cupboard2 o books;
books - doll;
books - clock;
doll - panda;
doll - radio

Participant 10(m):

Also. In dem Raum steht ein rotes Sofa. Auf dem Sofa liegen zwei Kissen. Davor sitzt eine Puppe. Hinter dem Sofa ist 'ne Lampe. An der Wand links hängt ein Foto von mehreren Bären, daneben hängt ein kleineres Bild mit irgendwelchen anderen Tieren. In der hinteren Ecke des Bildes steht ein Schrankwand, Regal, teilweise verglast. Da drin stehen ein paar Bücher, paar Stofftiere, einige Flaschen unten, ja. Vor dem Sofa steht 'n kleiner quadratischer Holztisch, da drauf steht 'ne Schale mit so trockenen Orangen. Man sieht, dass auf der rechten Seite ein Fenster sein muss, da is'n Vorhang und da drunter ist 'ne Heizung. Ja. Fußboden ist mit Teppichboden belegt, da drauf liegt noch ein beiger Teppich. An der Wand hängt noch 'n Fächer. Auch auf der linken Seite ist – scheint 'n Fenster zu sein, auch da gibt's einen Vorhang. (Ja. Das war's.)

Well, a red sofa stands in the room. On the sofa are two pillows (pillow1 and pillow2). In front of the pillows sits a doll. A lamp stands behind the sofa. A picture showing several bears (picture1) is hanging left at the wall. Next to this picture is a smaller picture showing other animals (picture2). You can find a cupboard in the corner. In the cupboard are some books, some soft toys, and

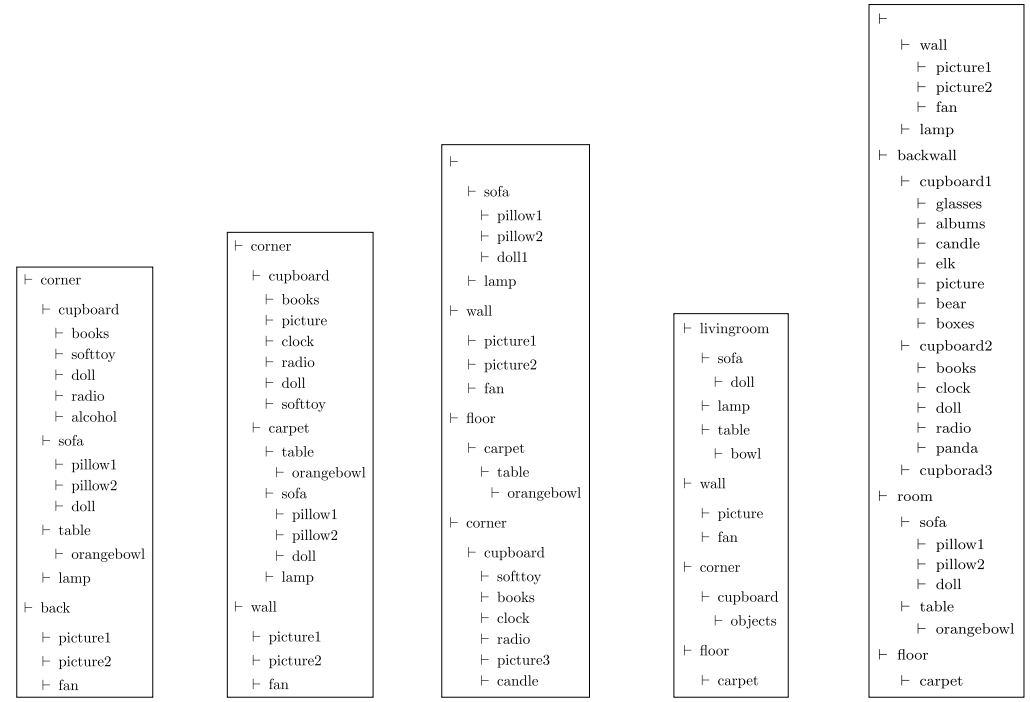
some bottles. In front of the sofa stands a small table with an orange bowl on it. You can see that there must be a window on the right side as you can see a curtain and below it a heater. The floor is covered with a carpet. A fan hangs at the wall. There seems to be a window on the left side of the room as there exists also a curtain. (That's it.)

room o sofa;
sofa | pillow1;
sofa | pillow2;
pillow2 - doll;
pillow2 - doll;

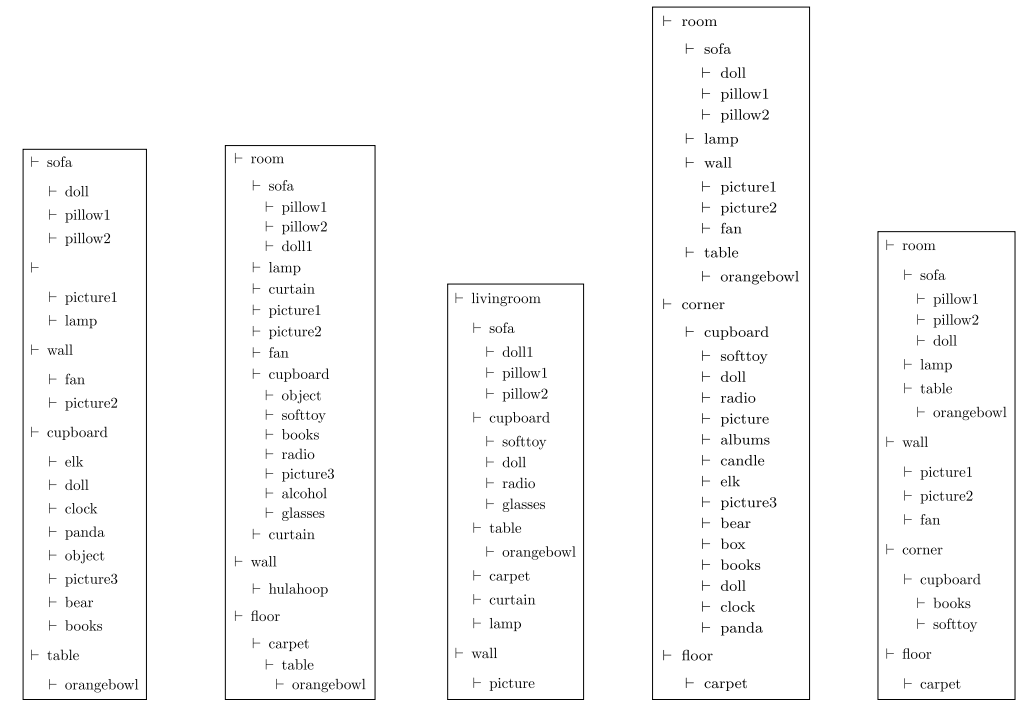
↓

sofa - lamp;
wall o picture1;
picture1 - picture2;
corner o cupboard;
cupboard o books;

cupboard o softtoy;
sofa - table;
table | orangebowl;
floor o carpet;
wall o fan



(a) subject 1(m) (b) subject 2(m) (c) subject 3(m) (d) subject 4(m) (e) subject 5(m)



(f) subject 6(m) (g) subject 7(m) (h) subject 8(m) (i) subject 9(m) (j) subject 10(m)

Figure B.3: These raw tree sets are generated from the descriptions about the living room given during the main study.

BIBLIOGRAPHY

[Ada94] R. Adams and L. Bischof. Seeded Region Growing. *Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[And05] H. Andreasson, R.h Triebel, and W. Burgard. Improving Plane Extraction from 3D Data by Fusing Laser Data and Vision. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2656–2661, Edmonton, Canada, 2005. IEEE.

[ASM00] *Practical Guide to Image Analysis*. ASM International, 2000.

[Bea02] M. S. Beauchamp, K. E. Lee, J. V. Haxby, and A. Martin. Parallel Visual Motion Processing Streams for Manipulable Objects and Human Movements. *Neuron*, 34:149–159, 2002.

[Bee07] P. Beeson, M. MacMahon, J. Modayil, A. Murarka, B. Kuipers, and B. Stankiewicz. Integrating Multiple Representations of Spatial Knowledge for Mapping, Navigation, and Communication. In *Proceedings of the Symposium on Interaction Challenges for Intelligent Assistants*, AAAI Spring Symposium Series, 2007. AAAI Technical Report SS-07-04.

[Bei97] J. Beis and D. G. Lowe. Shape Indexing using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, San Juan, Puerto Rico, 1997. IEEE.

[Bel61] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[Ber03] M. Berthold and D. J. Hand. *Intelligent Data Analysis*. Springer, 2nd edition, 2003.

[Bes92] P. J. Besl and N. D. McKay. A Method for Registration of 3D Shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[Beu10] N. Beuter, A. Swadzba, F. Kummert, and S. Wachsmuth. Using Articulated Scene Models for Dynamic 3D Scene Analysis in Vista Spaces. *3D Research*, 1(3), 2010.

[Bla95] G. Blais and M. Levine. Registering Multiview Range Data to Create 3D Computer Objects. *Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.

[Bog79] L. Boggess. Spatial Operators in Natural Language Understanding: The Prepositions. In *Proceedings of the Southeast Regional Conference*, pages 8–11. ACM, 1979.

[Bos08] A. Bosch, A. Zisserman, and X. Muñoz. Scene Classification using a Hybrid Generative/Discriminative Approach. *Transactions on Pattern Analysis and Machine Intelligence*, 30(4):712–727, 2008.

[Bre81] W. F. Brewer and J. C. Treyens. Role of Schemata in Memory for Places. *Cognitive Psychology*, 13:207–230, 1981.

[Bre07] M. Brenner, N. Hawes, J. Kelleher, and J. Wyatt. Mediating Between Qualitative and Quantitative Representations for Task-Orientated Human-Robot Interaction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2072–2077, Hyderabad, India, 2007. AAAI Press.

[Bre08] M. D. Breitenstein, E. Sommerlade, B. Leibe, L. van Gool, and I. Reid. Probabilistic Parameter Selection for Learning Scene Structure from Video. In *Proceedings of the British Machine Vision Conference*, Leeds, UK, 2008. British Machine Vision Association.

[Bro99] G. J. Brostow and I. A. Essa. Motion Based Decomposition of Video. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 8–13, Kerkyra, Corfu, Greece, 1999. IEEE.

[Bus02] P. Buschka and A. Saffiotti. A Virtual Sensor for Room Detection. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 1, pages 637–642, Lausanne, Switzerland, 2002. IEEE.

[Bux03] H. Buxton. Learning and Understanding Dynamic Scene Activity: A Review. *Image and Vision Computing*, 21:125–136, 2003.

[Can02] H. Cantzler, R. B. Fisher, and M. Devy. Improving Architectural 3D Reconstruction by Plane and Edge Constraining. In *Proceedings of the British Machine Vision Conference*, pages 43–52, Cardiff, UK, 2002. British Machine Vision Association.

[Cob01] D. Cobzas and H. Zhang. Planar Patch Extraction with Noisy Depth Data. In *Proceedings of the International Conference on Recent Advances in 3D Digital Imaging and Modeling*, pages 240–245, Quebec City, Canada, 2001. IEEE.

[COG04] COGNIRON. The Cognitive Robot Companion, 2004. (FP6-IST-002020), http://www.cogniron.org.

[Csá03] P. Csákány and A. M. Wallace. Representation and Classification of 3D Objects. *Systems, Man, and Cybernetics*, 33(4):638–647, 2003.

[Dee08] H. M. Dee, R. Fraile, D. C. Hogg, and A. G. Cohn. Modelling Scenes using the Activity within Them. In *Proceedings of the International Conference on Spatial Cognition VI*, Lecture Notes in Artificial Intelligence, pages 394–408, Berlin, Heidelberg, 2008. Springer.

[Del05] E. Delage, H. Lee, and A. Y. Ng. Automatic Single-Image 3D Reconstructions of Indoor Manhattan World Scenes. In *Proceedings of the International Symposium on Robotics Research*, 2005.

[Divo9] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An Empirical Study of Context in Object Detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1271–1278, Miami, FL, USA, 2009. IEEE.

[Doro7] P. Dorninger and C. Nothegger. 3D Segmentation of Unstructured Point Clouds for Building Modelling. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, pages 191–196, Munich, Germany, 2007.

[Eps98] R. Epstein and N. Kanwisher. A Cortical Representation of the Local Visual Environment. *Nature*, 392:598–601, 1998.

[FFo5] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 524–531, San Diego, CA, USA, 2005. IEEE.

[Fin99] G. A. Fink. Developing HMM-based Recognizers with ESMERALDA. *Lecture Notes in Artificial Intelligence*, 1692:229–234, 1999.

[Fis81] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[För87] W. Förstner and E. Gülch. A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features. In *ISPRS Intercommission Workshop*, pages 281–305, Interlaken, Switzerland, 1987.

[Freo8] C. Freksa. Zur interdiziplinären Erforschung räumlichen Denkens. *Kognitive Psychologie – Ausgewählte Grundlagen- und Anwendungsbeispiele*, pages 87–108, 2008.

[Frio7] J. Fritsch and S. Wrede. An Integration Framework for Developing Interactive Robots. In Davide Brugali, editor, *Software Engineering for Experimental Robotics*, volume 30 of *Springer Tracts in Advanced Robotics*, pages 291–305. Springer, Berlin, 2007.

[Fua97] P. Fua. From Multiple Stereo Views to Multiple 3D Surfaces. *International Journal of Computer Vision*, 24(1):19–35, 1997.

[Galo5] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. A. Fernández-Madrigal, and J. González. Multi-Hierarchical Semantic Maps for Mobile Robotics. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 3492–3497, Edmonton, Canada, 2005. IEEE.

[Gap95] K.-P. Gapp. Angle, Distance, Shape, and their Relationship to Projective Relations. Technical report, University of Saarland, CRC 314, 1995.

[Gib50] J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, USA, 1950.

[Gra97] O. Grau. A Scene Analysis System for the Generation of 3D Models. In *Proceedings of the International Conference on Recent Advances in 3D Digital Imaging and Modeling*, pages 221–228, Ottawa, Canada, 1997. IEEE.

[Gua07] L. Guan, J.-S. Franco, and M. Pollefeys. 3D Occlusion Inference from Silhouette Cues. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MN, USA, 2007. IEEE.

[Häh03] D. Hähnel, W. Burgard, and S. Thrun. Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. *Robotics and Autonomous Systems*, 44(1):15–27, 2003.

[Han08] M. Hanheide and G. Sagerer. Active Memory-based Interaction Strategies for Learning-enabling Behaviors. In *Proceedings of the International Symposium on Robot and Human Interactive Communication*, Munich, Germany, 2008.

[Har03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[Hay03] E. Hayman and J.-O. Eklundh. Statistical Background Subtraction for a Mobile Observer. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 67–74, Nice, France, 2003. IEEE.

[Hed08] G.-P.M. Hedge and C. Ye. SwissRanger SR-3000 Range Images Enhancement by a Singular Value Decomposition Filter. In *Proceedings of the International Conference on Information and Automation*, pages 1626–1631, Zhangjiajie, China, 2008. IEEE.

[Hed09] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the Spatial Layout of Cluttered Rooms. In *Proceedings of the International Conference on Computer Vision*, pages 1849–1856, Kyoto, Japan, 2009. IEEE.

[Hen99] J. M. Henderson and A. Hollingworth. High-level Scene Perception. *Annual Review of Psychology*, 50:243–271, 1999.

[Hen04] J. M. Henderson and F. Ferreira. *The Interface of Language, Vision, and Action: Eye Movements and the Visual World*, chapter Scene Perception for Psycholinguists, pages 1–58. Psychology Press, 2004.

[Hen07] J. M. Henderson, C. L. Larson, and D. C. Zhu. Cortical Activation to Indoor versus Outdoor Scenes: An fMRI Study. *Experimental Brain Research*, 179:75–84, 2007.

[Hen08] J. M. Henderson, C. L. Larson, and D. C. Zhu. Full Scenes Produce More Activation than Close-up Scenes and Scene-diagnostic Objects in Parahippocampal and Retrosplenial Cortex: An fMRI Study. *Brain and Cognition*, 66(1):40–49, 2008.

[Her85] A. Herskovits. Semantics and Pragmatics of Locative Expressions. *Cognitive Science*, 9:341–378, 1985.

[Her94] D. Hernández. *Qualitative Representation of Spatial Knowledge*, volume 804 of *Lecture Notes in Artificial Intelligence*. Springer, 1994.

[Het01] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3D Object Recognition from Range Images using Local Feature Histograms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 394–399, Kauai, HI, USA, 2001. IEEE.

[Hir85] S. C. Hirtle and J. Jonides. Evidence of Hierarchies in Cognitive Maps. *Memory & Cognition*, 13:208–217, 1985.

[Hoi06] D. Hoiem, A. A. Efros, and M. Herbert. Putting Objects in Perspective. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2137–2144, New York, NY, USA, 2006. IEEE.

[Hoi07] D. Hoiem, A. A. Efros, and M. Hebert. Recovering Surface Layout from an Image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[Hoi08] J. Hois, M. Wünstel, J. A. Bateman, and T. Röfer. Dialog-based 3D-Image Recognition using a Domain Ontology. In *Spatial Cognition*, volume 4287 of *Lecture Notes in Computer Science*, pages 107–126, Freiburg, Germany, 2008. Springer.

[Hor81] Berthold K.P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[Hub04] D. F. Huber, A. Kapuria, R. Donamukkala, and M. Hebert. Parts-based 3D Object Classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 82–89, Washington, DC, USA, 2004.

[Hut79] J. Huttenlocher and C. C. Presson. The Coding and Transformation of Spatial Information. *Cognitive Psychology*, 11:375–394, 1979.

[Jam10] M. Jamieson, A. Fazly, S. Stevenson, S. Dickinson, and S. Wachsmuth. Using Language to Learn Structured Appearance Models for Image Annotation. *Transactions on Pattern Analysis and Machine Intelligence*, 32(1):148–164, 2010.

[JL80] P. N. Johnson-Laird. Mental Models in Cognitive Science. *Cognitive Science*, 4:71–115, 1980.

[Joa99] T. Joachims. Making Large-scale SVM Learning Practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press, 1999.

[Joa02] T. Joachims. *Learning to Classify Text using Support Vector Machines*. PhD thesis, Cornell University, 2002.

[Joh99] A. E. Johnson and M. Herbert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[Käh08] O. Kähler, E. Rodner, and J. Denzler. On Fusion of Range and Intensity Information using Graph-Cut for Planar Patch Segmentation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):365–373, 2008.

[Kim05] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-Time Foreground-Background Segmentation using Codebook Model. *Real-Time Imaging*, 11:172–185, 2005.

[Kim06] S. Kim and I. S. Kweon. Scene Interpretation: Unified Modeling of Visual Context by Particle-based Belief Propagation in Hierarchical Graphical Model. In *Proceedings of the Asian Conference on Computer Vision*, volume 3852 of *Lecture Notes in Computer Science*, pages 963–972, Hyderabad, India, 2006. Springer.

[Kit98] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On Combining Classifiers. *Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[Kla09] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, and R. Klette. Moving Object Segmentation using Optical Flow and Depth Information. In *Proceedings of the Symposium on Advances in Image and Video Technology*, pages 611–623, 2009.

[Koi03] K. Koile, K. Tollmar, D. Demirdjian, H. Shrobe, and T. Darrell. Activity Zones for Context-aware Computing. In *Proceedings of the International Conference on Ubiquitos Computing*, volume 2864 of *Lecture Notes in Computer Science*, pages 90–106, Seattle, WA, USA, 2003. Springer.

[Kru78] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Number 07-011 in Sage University Paper Series on Quantitative Application in the Social Sciences. Sage Publications, Beverly Hills and London, 1978.

[Kui00] B. Kuipers. The Spatial Semantic Hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[Kun02] L. I. Kuncheva. Theoretical Study on Six Classifier Fusion Strategies. *Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002.

[Kun04] L. I. Kuncheva. *Combining Pattern Classifiers*. John Wiley & Sons, 2004.

[Lak06] R. Lakaemper and L. J. Latecki. Using Extended EM to Segment Planar Structures in 3D. In *Proceedings of the International Conference on Pattern Recognition*, pages 1077–1082, Hong Kong, China, 2006. IEEE Computer Society.

[Lam97] L. Lam and C. Y. Suen. Application of Majority Voting to Pattern Recognition: An Analysis of its Behavior and Performance. *Transactions on Systems, Man, and Cybernetics*, 27(5):553–568, 1997.

[Lan01] R. Lange and P. Seitz. Solid-State, Time-of-Flight Range Camera. *Journal of Quantum Electronics*, 37(3):390–397, 2001.

[Laz06] S. Lazebnik, C. Schmid, and J. Ponce. 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*, pages 2169–2178, New York, NY, USA, 2006. IEEE.

[Lee77] D. T. Lee and C. K. Wong. Worst-case Analysis for Region and Partial Region Searches in Multidimensional Binary Search Trees and Balanced Quad Trees. *Acta Informatica*, 9(1):23–29, 1977.

[Lee05] S. Lee, D. Jang, E. Kim, S. Hong, and J. Han. A Real-Time 3D Workspace Modeling with Stereo Camera. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 2140–2147, Edmonton, Canada, 2005. IEEE.

[Lee09] D. Lee, M. Hebert, and T. Kanade. Geometric Reasoning for Single Image Structure Recovery. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2136–2143, Miami, FL, USA, 2009. IEEE.

[Lev02] D. T. Levin, D. J. Simons, B. L. Angelone, and C. F. Chabris. Memory for Centrally Attended Changing Objects in an Incidental Real-World Change Detection Paradigm. *British Journal of Psychology*, 93:289–302, 2002.

[Log96] G. D. Logan and D. Sandler. A Computational Analysis of the Apprehension of Spatial Relations. In P. Bloom, M. Peterson, L. Nadek, and M. Garrett, editors, *Language and Space*, pages 493–526. MIT Press, 1996.

[Lor97] A. Lorusso, D. W. Eggert, and R. B. Fisher. A Comparison of Four Algorithms for Estimating 3D Rigid Transformations. *Machine Vision and Applications*, 9:295–307, 1997.

[Low99] D. G. Lowe. Object Recognition from Local Scale-invariant Features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157, Kerkyra, Corfu, Greece, 1999. IEEE.

[Luc81] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, Canada, 1981.

[Lüt09] I. Lütkebohle, J. Peltason, L. Schillingmann, B. Wrede, S. Wachsmuth, C. Elbrechter, and R. Haschke. The Curious Robot – Structuring

Interactive Robot Learning. In *Proceedings of the International Conference on Robotics and Automation*, pages 4156–4162, Kope, Japan, 2009. IEEE.

[Mako3] D. Makris and T. Ellis. Automatic Learning of an Activity-based Semantic Scene Model. In *Proceedings of the Conference on Advanced Video and Signal Based Surveillance*, pages 183–188, Miami, FL, USA, 2003. IEEE.

[Maro2] C. Martin and S. Thrun. Real-Time Acquisition of Compact Volumetric Maps with Mobile Robots. In *Proceedings of the International Conference on Robotics and Automation*, Washington, DC, USA, 2002. IEEE.

[Mat99] P. Matsakis and L. Wendling. A New Way to Represent the Relative Position between Areal Objects. *Transactions on Pattern Analysis and Machine Intelligence*, 21(7):634–643, 1999.

[Mavo6] N. Mavridis and D. Roy. Grounded Situation Models for Robots: Where Words and Percepts Meet. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 4690–4697, Beijing, China, 2006. IEEE.

[Mito9] A. Mittal, A. Monnet, and N. Paragios. Scene Modeling and Change Detection in Dynamic Scenes: A Subspace Approach. *Computer Vision and Image Understanding*, 113(1):63–79, 2009.

[Miy94] K. Miyajima and A. Ralescu. Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations. *Fuzzy Sets and Systems*, 65:225–236, 1994.

[Mölo5] T. Möller, H. Kraft, J. Frey, M. Albrecht, and R. Lange. Robust 3D Measurement with PMD Sensors. In *Proceedings of the 1st Range Imaging Research Day at ETH*, 2005.

[Mon93] D. R. Montello. Scale and Multiple Psychologies of Space. In *Lecture Notes in Computer Science: Spatial Information Theory A Theoretical Basis for GIS*, volume 716, pages 312–321, 1993.

[Moro1] R. Moratz, K. Fischer, and T. Tenbrink. Cognitive Modeling of Spatial Reference for Human-Robot Interaction. *International Journal of Artificial IntelligenceTools*, 10(4):589–611, 2001.

[Mozo5] Ó. M. Mozos, C. Stachniss, and W. Burgard. Supervised Learning of Places from Range Data using AdaBoost. In *Proceedings of the International Conference on Robotics and Automation*, pages 1730–1735, Barcelona, Spain, 2005. IEEE.

[Mozo7] Ó. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised Semantic Labeling of Places using Information Extracted from Sensor Data. *Robotics and Autonomous Systems*, 55(5):391–402, 2007.

[Muno9a] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 975–982, Miami, FL, USA, 2009. IEEE.

[Muno9b] D. Munoz, N. Vandapel, and M. Hebert. Onboard Contextual Classification of 3D Point Clouds with Learned High-order Markov Random Fields. In *Proceedings of the International Conference on Robotics and Automation*, pages 4273–4280, Kobe, Japan, 2009. IEEE.

[Muro4] D. Murray and J. J. Little. Segmenting Correlation Stereo Range Images using Surface Elements. In *Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission*, pages 656–663, Washington, DC, USA, 2004. IEEE Computer Society.

[Naro5] A. Narasimhamurthy. Theoretical Bounds of Majority Voting Performance for a Binary Classification Problem. *Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1988–1995, 2005.

[Neuo8] B. Neumann and R. Möller. On Scene Interpretation with Description Logics. *Image and Vision Computing: Special Issue on Cognitive Vision*, 26(1):82–101, 2008.

[Nüco8] A. Nüchter and J. Hertzberg. Towards Semantic Maps for Mobile Robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[Ogg04] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. BLanc. An All-Solid-State Optical Range Camera for 3D Real-Time Imaging with Sub-Centimeter Depth Resolution (SwissRanger). In *Proceedings of the International Society for Optical Engineering – Specific Applications: Sensors and Medical Optics*, volume 5249, 2004.

[Oli94] P. Olivier, T. Maeda, and J. Tsujii. Automatic Depiction of Spatial Descriptions. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, pages 1405–1410, Seattle, WA, USA, 1994. AAAI Press.

[Olio1] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[Pelo9] J. Peltason, F. H. K. Siepmann, T. P. Spexard, B. Wrede, M. Hanheide, and E. A. Topp. Mixed-Initiative in Human Augmented Mapping. In *Proceedings of the International Conference on Robotics and Automation*, pages 3175–3182, Kobe, Japan, 2009. IEEE.

[Peuo4] P. Peursum, S. Venkatesh, G. West, and H. H. Bui. Using Interaction Signatures to Find and Label Chairs and Floors. *Pervasive Computing*, 3(4):58–65, 2004.

[Phi03] R. Philippsen and R. Siegwart. Smooth and Efficient Obstacle Avoidance for a Tour Guide Robot. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 446–451, Taipei, Taiwan, 2003. IEEE.

[Pic04] M. J. Pickering and S. Garrod. Toward a Mechanistic Psychology of Dialogue. *Behavioral and Brain Sciences*, 27(2):169–190, 2004.

[Pir04] F. Pirri. Indoor Environment Classification and Perceptual Matching. In *Proceedings of the International Conference on Knowledge Representation*, pages 73–84, Whistler, Canada, 2004. AAAI Press.

[Pos06] I. Posner, D. Schröter, and P. M. Newman. Using Scene Similarity for Place Labelling. In *Proceedings of the International Symposium on Experimental Robotics*, volume 39 of *Springer Tracts in Advanced Robotics*, pages 85–98, Rio de Janeiro, Brazil, 2006. Springer.

[Pro06] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen. A Discriminative Approach to Robust Visual Place Recognition. In *Proceedings of the International Conference on Intelligent Robots and Systems*, Beijing, China, 2006. IEEE.

[Pro10a] A. Pronobis, B. Caputo, P. Jensfelt, and H. I. Christensen. A Realistic Benchmark for Visual Indoor Place Recognition. *Robotics and Autonomous Systems*, 58:81–96, 2010.

[Pro10b] A. Pronobis, Ó. M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal Semantic Place Classification. *International Journal of Robotics Research*, 29(2–3):298–320, 2010.

[Qua09] A. Quattoni and A. Torralba. Recognizing Indoor Scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 413–420, Miami, FL, USA, 2009. IEEE.

[Rab06] T. Rabbani, F. A. van den Heuvel, and G. Vosselman. Segmentation of Point Clouds using Smoothness Constraints. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 36, pages 248–253, Dresden, Germany, 2006.

[Ran07] A. Ranganathan and F. Dellaert. Semantic Modeling of Places using Objects. In *Proceedings of the Robotics: Science and Systems*, Atlanta, GA, USA, 2007.

[Reg01] T. Regier and L. A. Carlson. Grounding Spatial Language in Perception: An Empirical and Computational Investigation. *Journal of Experimental Psychology: General*, 130(2):273–298, 2001.

[Ren02] R. A. Rensink. Change Detection. *Annual Review of Psychology*, 53:245–277, 2002.

[RS88] G. Retz-Schmidt. Various Views on Spatial Prepositions. *Artificial Intelligence Magazine*, 9(2):95–105, 1988.

[Rum86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. *Parallel Data Processing*, 1:318–362, 1986.

[Rus08] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Learning Informative Point Classes for the Acquisition of Object Model Maps. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, pages 643–650, Hanoi, Vietnam, 2008. IEEE.

[Sam02] H. Samet and A. Kochut. Octree Approximation and Compression Methods. In *Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission*, pages 460–469, Padova, Italy, 2002. IEEE.

[San02] B. C. S. Sanders, T. C. Nelson, and R. Sukthankar. A Theory of the Quasi-Static World. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 1–6, Quebec, Canada, 2002. IEEE Computer Society.

[Sax08] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2008.

[Sch66] Peter Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. University of North Carolina.

[Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[Sch07] J. Schmidt, C. Wöhler, L. Krüger, T. Gövert, and C. Hermes. 3D Scene Segmentation and Object Tracking in Multiocular Image Sequences. In *Proceedings of the International Conference on Computer Vision Systems*, Bielefeld, Germany, 2007.

[Sco83] D.W. Scott and J. R. Thompson. Probability Density Estimation in Higher Dimensions. In *Proceedings of the Symposium on the Interface*, pages 173–179, 1983.

[She05] Y. Sheikh and M. Shah. Bayesian Modeling of Dynamic Scenes for Object Detection. *Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[Sim98] D. J. Simons and D. T. Levin. Failure to Detect Changes to People in a Real-World Interaction. *Psychonomic Bulletin and Review*, 5:644–649, 1998.

[Sku04] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial Language for Human-Robot Dialogs. *Transactions on Systems, Man, and Cybernetics*, 34(2):154–167, 2004.

[Soc00] Gudrun Socher, Gerhard Sagerer, and Pietro Perona. Bayesian Reasoning on Qualitative Descriptions from Images and Speech. *Image and Vision Computing*, 18(2):155–172, 2000.

[Som10] G. Somanath and C. Kambhamettu. Abstraction and Generalization of 3D Structure for Recognition in Large Intra-Class Variation. In *Proceedings of the Asian Conference on Computer Vision*, volume 3 of *Lecture Notes in Computer Science*, pages 1793–1806, Queenstown, New Zealand, 2010. Springer.

[Spe06] T. Spexard, S. Li, B. Wrede, J. Fritsch, G. Sagerer, O. Booij, Z. Zivkovic, B. Terwijn, and B. Kröse. BIRON, where are you? – Enabling a Robot to Learn New Places in a Real Home Environment by Integrating Spoken Dialog and Visual Localization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 934–940, Beijing, China, 2006. IEEE.

[Sta99] C. Stauffer and W. E. L. Grimson. Adaptive Background Mixture Models for Real-Time Tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, Fort Collins, CO, USA, 1999. IEEE.

[Sta02] I. Stamos and P. K. Allen. Geometry and Texture Recovery of Scenes of Large Scale. *Computer Vision and Image Understanding*, 88(2):94–118, 2002.

[Stu09] J. Sturm, V. Predeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard. Learning Kinematic Models for Articulated Objects. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1851–1856, Pasadena, CA, USA, 2009. AAAI Press.

[Stu10] J. Sturm, K. Konelige, C. Stachniss, and W. Burgard. Vision-based Detection for Learning Articulation Models of Cabinet Doors and Drawers in Household Environments. In *Proceedings of the International Conference on Robotics and Automation*, pages 362–368, Anchorage, AK, USA, 2010. IEEE.

[Swa06] A. Swadzba. Estimation of Camera Motion from Depth Image Sequences. Master's thesis, Institute of Pattern Recognition, University of Erlangen-Nürnberg, 2006.

[Swa07] A. Swadzba, B. Liu, J. Penne, O. Jesorsky, and R. Kompe. A Comprehensive System for 3D Modeling from Range Images Acquired from a 3D ToF Sensor. In *Proceedings of the International Conference on Computer Vision Systems*, Bielefeld, Germany, 2007.

[Swa08a] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer. Tracking Objects in 6D for Reconstructing Static Scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, USA, 2008. IEEE.

[Swa08b] A. Swadzba, A. Vollmer, M. Hanheide, and S. Wachsmuth. Reducing Noise and Redundancy in Registered Range Data for Planar Surface Extraction. In *Proceedings of the International Conference on Pattern Recognition*, pages 1–4, Tampa, FL, USA, 2008. IEEE Computer Society.

[Swa08c] A. Swadzba and S. Wachsmuth. Categorizing Perceptions of Indoor Rooms using 3D Features. In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 5342 of *Lecture Notes in Computer Science*, pages 744–754, Orlando, FL, USA, 2008. Springer.

[Swa09] A. Swadzba, C. Vorwerg, S. Wachsmuth, and G. Rickheit. A Computational Model for the Alignment of Hierarchical Scene Representations in Human-Robot Interaction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1857–1863, Pasadena, CA, USA, 2009. AAAI Press.

[Swa10a] A. Swadzba, N. Beuter, S. Wachsmuth, and F. Kummert. Dynamic 3D Scene Analysis for Acquiring Articulated Scene Models. In *Proceedings of the International Conference on Robotics and Automation*, pages 134–141, Anchorage, AK, USA, 2010. IEEE.

[Swa10b] Agnes Swadzba and Sven Wachsmuth. Indoor Scene Classification using combined 3D and Gist Features. In *Proceedings of the Asian Conference on Computer Vision*, volume 6493 of *Lecture Notes in Computer Science*, pages 725–739, Queenstown, New Zealand, 2010. Springer.

[Tal83] L. Talmy. How Language Structures Space. In Jr. H. L. Pick and L. P. Acredolo, editors, *Spatial Orientation: Theory, Research and Application*, pages 225–282, 1983.

[Thr00] S. Thrun, W. Burgard, and D. Fox. A Real-time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 321–328, San Francisco, CA, USA, 2000. IEEE.

[Thr02] I. M. Thronton and D. Fernandez-Duque. Converging Evidence for the Detection of Change without Awareness. *Progress in Brain Research*, 140:99–118, 2002.

[Tom98] V. D. Tomaso, V. Lombardo, and L. Lesmo. A Computational Model for the Interpretation of Static Locative Expressions. In P. Olivier and K.-P. Gapp, editors, *Representation and Processing of Spatial Expressions*, pages 73–90, 1998.

[Top08] E. Topp and H. Christensen. Detecting Structural Ambiguities and Transitions during a Guided Tour. In *Proceedings of the International Conference on Robotics and Automation*, pages 2564–2570, Pasadena, CA, USA, 2008. IEEE.

[Tor03a] A. Torralba. Contextual Priming for Object Detection. *International Journal of Computer Vision*, 53(2):153–167, 2003.

[Tor03b] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based Vision System for Place and Object Recognition. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 273–280, Nice, France, 2003. IEEE.

[Tor09] A. Torralba, B. C. Russell, and J. Yuen. LabelMe: Online Image Annotation and Applications. Technical report, MIT CSAIL, 2009.

[Tri05] R. Triebel, W. Burgard, and F. Dellaert. Using Hierarchical EM to Extract Planes from 3D Range Scans. In *Proceedings of the International Conference on Robotics and Automation*, pages 4437–4442, Barcelona, Spain, 2005. IEEE.

[Tri07] R. Triebel, R. Schmidt, Ó. M. Mozos, and W. Burgard. Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2225–2230, Hyderabad, India, 2007. AAAI Press.

[Tve98] B. Tversky and P. U. Lee. How Space Structures Language. In *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, volume 1404 of *Lecture Notes in Computer Science*, pages 157–176, London, UK, 1998. Springer.

[Tve99] B. Tversky, J. Kim, and A. Cohen. Mental Models of Spatial Relations and Transformations from Language. In *Mental Models in Discourse Processing and Reasoning*, pages 239–258. Elsevier Science B. V., 1999.

[Ull96] S. Ullman. *High-Level Vision: Object Recognition and Visual Cognition*. MIT Press, Cambridge, MA, 1996.

[Ull08] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and H. I. Christensen. Towards Robust Place Recognition for Robot Localization. In *Proceedings of the International Conference on Robotics and Automation*, pages 530–537, Pasadena, CA, USA, 2008. IEEE.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[Vas07a] S. Vasudevan, S. Gächter, V.T. Nguyen, and R. Siegwart. Cognitive Maps for Mobile Robots – An Object based Approach. *Robotics and Autonomous Systems*, 55(5):359–371, 2007. From Sensors to Human Spatial Concepts.

[Vas07b] S. Vasudevan, S. Gächter, and R. Siegwart. Cognitive Spatial Representations for Mobile Robots – Perspectives from a User Study. In *Proceedings of the International Conference on Robotics and Automation Workshops*, Roma, Italy, 2007.

[Vas07c] S. Vasudevan, A. Harati, and R Siegwart. A Bayesian Approach to Conceptualization and Place Classification: Using the Number of

Occurrences of Objects to Infer Concepts. In *Proceedings of the European Conference on Mobile Robotics*, Freiburg, Germany, 2007.

[Vaso7d] S. Vasudevan and R. Siegwart. A Bayesian Approach to Conceptualization and Place Classification: Incorporating Spatial Relationships (Distances) to Infer Concepts. In *Proceedings of the International Conference on Intelligent Robots and Systems Workshops*, San Diego, CA, USA, 2007.

[Viso9] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. Mackworth. Automated Spatial-Semantic Modeling with Applications to Place Labeling and Informed Search. In *Proceedings of the Canadian Conference on Computer and Robot Vision*, pages 284–291, Kelowna, Canada, 2009. IEEE.

[Vogo4] J. Vogel and B. Schiele. A Semantic Typicality Measure for Natural Scene Categorization. In *Lecture Notes in Computer Science: Pattern Recognition – DAGM Symposium*, pages 195–203, Tübingen, Germany, 2004. Springer.

[Vor97] C. Vorwerg, G. Socher, T. Fuhr, G. Sagerer, and G. Rickheit. Projective Relations for 3D Space: Computational Model, Application, and Psychological Evaluation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 159–164, Providence, Rhode Island, 1997. AAAI Press / The MIT Press.

[Voro9] C Vorwerg and G. Rickheit. Verbal Room Descriptions Reflect Hierarchical Spatial Models. In *Annual Meeting of the American Psychological Society*, 2009. Poster.

[Waco2] S. Wachsmuth and G. Sagerer. Bayesian Networks for Speech and Image Integration. In *Proceedings of the AAAIConference on Artificial Intelligence*, pages 300–306, Edmonton, Canada, 2002. AAAI Press.

[Waco9] S. Wachsmuth, M. Hanheide, F. Siepmann, and T. P. Spexard. ToBI - Team of Bielefeld: The Human-Robot Interaction System for RoboCup@Home 2009. Technical report, Bielefeld University, 2009.

[Wac10] S. Wachsmuth, F. Siepmann, D. Schulze, and A. Swadzba. ToBI – Team of Bielefeld: The Human-Robot Interaction System for RoboCup@Home 2010. Technical report, Bielefeld University, 06/2010 2010.

[Wal8o] D. L. Waltz. Understanding Scene Descriptions as Event Simulations. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 7–11, Morristown, NJ, USA, 1980. Association for Computational Linguistics.

[Walo3] C. Wallraven, B. Caputo, and A. Graf. Recognition with Local Features: The Kernel Recipe. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 257–264, Nice, France, 2003. IEEE.

[Wan06] X. Wang, K. Tieu, and E. Grimson. Learning Semantic Scene Models by Trajectory Analysis. In *Proceedings of the European Conference on Computer Vision*, volume 3953 of *Lecture Notes in Computer Science*, pages 110–123, Graz, Austria, 2006. Springer.

[Wan09] G. Wang, D. Hoiem, and D. Forsyth. Building Text Features for Object Image Classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1367–1374, Miami, FL, USA, 2009. IEEE.

[Wan10] H. Wang, S. Gould, and D. Koller. Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding. In *Proceedings of the European Conference on Computer Vision*, volume 6314 of *Lecture Notes in Computer Science*, pages 497–510, Hersonissos, Heraklion, Crete, Greece, 2010. Springer.

[Wei03] J. Weingarten, G. Gruener, and R. Siegwart. A Fast and Robust 3D Feature Extraction Algorithm for Structured Environment Reconstruction. In *Proceedings of the International Conference on Advanced Robotics*, Coimbra, Portugal, 2003.

[Wei04] J. Weingarten, G. Gruener, and R. Siegwart. A State-of-the-Art 3D Sensor for Robot Navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3, pages 2155–2160, Sendai, Japan, 2004. IEEE.

[Wen07] A. Wendel and A. Pinz. Scene Categorization from Tiny Images. In *Proceedings of the Workshop of the Austrian Association for Pattern Recognition*, 2007.

[Woj10] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes. In *Proceedings of the European Conference on Computer Vision*, volume 6314 of *Lecture Notes in Computer Science*, pages 467–481, Hersonissos, Heraklion, Crete, Greece, 2010. Springer.

[Wre06] S. Wrede, M. Hanheide, S. Wachsmuth, and G. Sagerer. Integration and Coordination in a Cognitive Vision System. In *Proceedings of the International Conference on Computer Vision Systems*, Manhattan, New York City, USA, 2006. IEEE.

[Wre08] S. Wrede. *An Information-driven Architecture for Cognitive Systems Research*. PhD thesis, Bielefeld University, 2008.

[Yu08] S. X. Yu, H. Zhang, and J. Malik. Inferring Spatial Layout from A Single Image via Depth-ordered Grouping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, USA, 2008. IEEE.

[Yua09] F. Yuan, A. Swadzba, R. Philippsen, O. Engin, M. Hanheide, and S. Wachsmuth. Laser-based Navigation Enhanced with 3D Time-of-

Flight Data. In *Proceedings of the International Conference on Robotics and Automation*, pages 2844–2850, Kobe, Japan, 2009. IEEE.

[Zen08] H. Zender, O. Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard. Conceptual Spatial Representations for Indoor Mobile Robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.

[Zie10] L. Ziegler, F. Siepmann, M. Kortkamp, and S. Wachsmuth. Towards an Informed Search Behavior for Domestic Robots. In *Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots Workshops: Domestic Service Robots in the Real World*, Darmstadt, Germany, 2010.

[Zin03] T. Zinßer, J. Schmidt, and H. Niemann. A Refined ICP Algorithm for Robust 3D Correspondence Estimation. In *Proceedings of the International Conference on Image Processing*, volume 2, pages 695–698, Barcelona, Spain, 2003. IEEE.

[Zwa98] R. A. Zwaan and G. A. Radvansky. Situation Models in Language Comprehension and Memory. *Psychological Bulletin*, 123:162–185, 1998.

[Zwa99] R. A. Zwaan. Situation Models: The Mental Leap into Imagined Worlds. *Current Directions in Psychological Science*, 8(1):15–18, 1999.

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

BIC	Bayesian Information Criterion
CRFH	Composed Receptive Field Histograms
EM	Expectation Maximization
FPM	Fua's Patch Merging
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
ICP	Iterative Closest Points
IDI	Information-Driven-Integration
MRF	Markov Random Field
NN	Neural Network
PCA	Principal Component Analysis
PDF	Probability Density Function
RBF	Radial Basis Function
RANSAC	RANdom SAmple Consensus
RG	Region Growing
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SVD	Singular Value Decomposition
SVM	Support Vector Machines
VIPP	Virtual Image Plane Projection
VS	Voxel Sampling
AM	Active Memory
BIRON	BIelefeld Robot CompaniON
BonSAI	BirON Sensor Actuator Interface
LED	Light-Emitting Diode
ToBI	Team of BIelefeld
ToF	Time-of-Flight
fMRI	functional Magnetic Resonance Imaging
MTG	Middle Temporal Gyrus
PPA	Parahippocampal Place Area
STS	Superior Temporal Sulcus
COLD	COsy Localization Database
IDOL	Image Database for rObot Localization
INDECS	INDoor Environment under Changing conditionS
fps	frame per second
mm	millimeter