Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Prof. Dr.-Ing./Univ. Tokio Martin Buss

Semantic Mapping for Autonomous Robots in Urban Environments

Christian W. Landsiedel

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende:		Prof. Dongheui Lee, Ph.D.
Prüfende der Dissertation:		
	1.	Prof. Dr.-Ing. habil. Dirk Wollherr
	2.	Prof. Michael Beetz, Ph.D.

Die Dissertation wurde am 13.11.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 23.4.2018 angenommen.

Foreword

Writing this thesis would not have been possible without the input and support by a group of people, to all of whom I am deeply thankful. Dirk Wollherr, my advisor for this thesis, I want to thank for his guidance and his deliberation in giving me the freedom to tackle challenging issues in my own way. Likewise, I am thankful to Martin Buss, who chaired the Chair for Automatic Control Engineering, for creating a unique working environment. I am obliged to the staff at LSR, namely Larissa Schmid, Brigitta Renner, Karin Rosenits and Wolfgang Jaschik, for all kinds of help with practical and organisational matters.

Much of the work presented in this thesis has its roots in the work done in the IURO project. I fondly remember the fun and effective work together with the IURO team at LSR, namely with Daniel Carton, Sheraz Khan, David Lenz, Nikos Mitsou, Roderick de Nijs, Sebastian Soentges, and Annemarie Turnwald. There are numerous other colleagues with whom I had much appreciated both technical discussions and engaging conversations in the coffee kitchen. Thank you for those, Laith Alkurdi, Daniel Althoff, Philine Donner, Ken Friedl, Stefan Friedrich, Volker Gabler, Milad Geravand, Gerold Huber, Robert Jenke, Andreas Lawitzky, Markus Schill, and Moritz Stötter.

I also much enjoyed the direct research collaboration with my colleagues. I am much indebted to Roderick de Nijs for discussions and fruitful work on pseudo-Boolean inference, and likewise to Sheraz Khan for working together with me on the topic of spatial relations for point cloud-based objects. I also had the great pleasure of collaborating with Matthew Walter and Verena Rieser on a survey paper about semantic mapping, and I thank them for their substantial input and feedback on my part of that paper.

During my time at the Chair for Automatic Control, I also had the opportunity to supervise the work of motivated and engaged students. Among those who worked with me, I want to especially acknowledge Mustafa Sezer for his help with the registration of the Munich 3D point could data set, and Joachim Neu and Huaijiang Zhu for their thesis work.

Most of all, my thankfulness goes out to my parents and my sister, who were always there to give steadfast support even in difficult times, and to Kadi, who provided the complex balance of focus and distraction I needed to complete this thesis.

Contents

List of Abbreviations

CM Chamfer Matching.

CNF conjunctive normal form.

DBN Dynamic Bayesian Network.

DCM Directional Chamfer Matching.

FDCM Fast Directional Chamfer Matching.

GNSS Global Navigation Satellite System.

GPS Global Positioning System.

HRI Human-Robot Interaction.

ILP integer linear program.

IURO Interactive Urban Robot.

KB knowledge base.

MAP maximum-a-posteriori.

MCMC Markov Chain Monte Carlo.

MLN Markov Logic Network.

MPE most probable explanation.

MRF Markov Random Field.

OSM OpenStreetMap.

PBMLN Pseudo-Boolean Markov Logic Network.

QPBO Quadratic Pseudo-Boolean Optimization.

QSR Qualitative Spatial Reasoning.

SVM Support Vector Machine.

Abstract

The increasing autonomy and interactivity of mobile robotic technologies require the representation of environment information on multiple levels. While quantitative geometric information is indispensable for navigation and motion planning, understanding and planning complex tasks and interacting with humans require knowledge represented at a higher level, which is able to express the semantics of the environment, and to reason about its spatial structure in a qualitative way that is easily relatable to humans. *Semantic Maps* combine these different features into a common environment representation. Robotic systems with different degrees of autonomy are expected to be commercially deployed in urban environments in the near future. These environments are typically larger and less structured than the more commonly studied indoor environments, and so they pose specific needs to the semantic representation. This thesis studies aspects of semantic mapping which are particular to urban settings. The presented methods are motivated by work done in a robotics project addressing interactive scenarios in urban environments, and by an extensive review of hybrid robotic mapping technologies, qualitative spatial representation and reasoning techniques, and their cognitive origins.

The first part of this thesis concerns the extraction of qualitative spatial relations between objects from point cloud data. The proposed approach relies on *Markov Logic Networks*, a probabilistic logic modelling technique, to model higher-order consistency relationships between spatial relations. The proposed approach includes the description a new inference technique for Markov Logic Networks, which relies on a purely algebraic pseudo-Boolean representation of the logical formulation, which enables the use of highly efficient inference techniques.

Next, the combination of point cloud data and semantic data from the online mapping repository OpenStreetMap for the problem of scene interpretation is considered. Precise sensor data in the form of point clouds is fused with street network information in order to estimate street geometries in an urban environment. The street network information is used to leverage a probabilistic model of the geometries of neighbouring street segments. Thus, the benefits of combining data from different sources for the creation of semantic map annotation are shown.

Finally, another aspect of combining sensor-level data with semantic data from another source is explored by using building outlines from a coarse semantic map to localize a robot in an unknown urban scene. The technique builds on the generic *chamfer matching* template matching technique, which is extended to include visibility analysis in the cost function to model the characteristics of the laser range finder providing the input data. Since the method is independent of the provenience of the input data, the formulation can be expected to generalize to other forms of input data, such as 3D point cloud data from monocular or stereo cameras. The method is shown to produce state-of-the-art results on two large, diverse datasets from different environments, and illustrates the power of semantic data from diverse sources in the localization task.

Zusammenfassung

Die zunehmende Autonomie und Interaktivität von mobilen Robotern erfordert die Darstellung von Umgebungsinformationen auf mehreren Ebenen. Während quantitative geometrische Informationen für die Bewegungsplanung unentbehrlich sind, verlangen die Planung komplexer Aufgaben und die Interaktion mit Menschen Wissen auf einer höheren Ebene, die die Semantik der Umwelt ausdrücken kann und die qualitatives Schließen über räumliche Strukturen erlaubt. Semantische Karten kombinieren diese verschiedenen Merkmale zu einer gemeinsamen Umgebungsdarstellung. Robotiksysteme mit unterschiedlichen Autonomiegraden sollen in naher Zukunft kommerziell in städtischen Umgebungen eingesetzt werden. Diese Umgebungen sind typischerweise größer und weniger strukturiert als die häufiger untersuchten Innenräume, und stellen deswegen besondere Bedürfnisse an die semantische Darstellung. Diese Arbeit untersucht Aspekte von semantischen Karten, die für städtische Einsatzgebiete besonders wichtig sind. Die vorgestellten Methoden werden motiviert durch Ergebnisse aus einem Robotik-Projekt, das interaktive Szenarien in städtischen Umgebungen behandelt, und durch eine umfassende Übersicht über in der Robotik eingesetzte hybride Kartierungstechniken, qualitative räumliche Repräsentationen und Schließmethoden, sowie deren kognitive Ursprünge.

Der erste Teil dieser Arbeit betrifft die Extraktion von qualitativen räumlichen Beziehungen zwischen Objekten aus Punktwolken. Der vorgeschlagene Ansatz stützt sich auf Markov Logic Networks, eine probabilistische Logikmodellierungstechnik, um Konsistenzbedingungen höherer Ordnung zwischen räumlichen Beziehungen zu modellieren. Der vorgeschlagene Ansatz beinhaltet die Beschreibung einer neuen Inferenztechnik für Markov Logic Networks, die auf einer rein algebraischen pseudo-booleschen Darstellung der logischen Formulierung beruht, die den Einsatz hocheffizienter Inferenztechniken ermöglicht.

Als nächstes wird die Kombination von Punktwolken und semantischen Daten aus dem Online-Kartendienst OpenStreetMap für das Problem der Szeneninterpretation betrachtet. Sensordaten in Form von Punktwolken werden mit Straßennetzinformationen verbunden, um Straßengeometrien in einer städtischen Umgebung abzuschätzen. Die Straßennetzinformation wird in einem probabilistischen Modell der Geometrien benachbarter Straßensegmente verwendet. So werden die Vorteile der Kombination von Daten aus verschiedenen Quellen für die Erstellung einer semantischen Kartenannotierung dargestellt.

Schließlich wird ein weiterer Aspekt der Kombination von Sensorebenen-Daten mit semantischen Daten aus einer anderen Quelle erforscht, indem Gebäudegrundrisse aus einer groben semantischen Karte verwendet werden, um einen Roboter in einer unbekannten urbanen Szene zu lokalisieren. Die Technik baut auf der generischen Methode *chamfer matching* auf, die um eine Sichtbarkeitsanalyse in der Kostenfunktion erweitert wird, die die Eigenschaften eines optischen Sensors modelliert. Es wird gezeigt, dass die Methode auf zwei großen Datensätzen aus verschiedenen Umgebungen Ergebnisse auf dem Stand der Technik liefert und so die Vorteile der Kombination von semantischen Daten aus verschiedenen Quellen für die Lokalisierungsaufgabe veranschaulicht.

1

Introduction

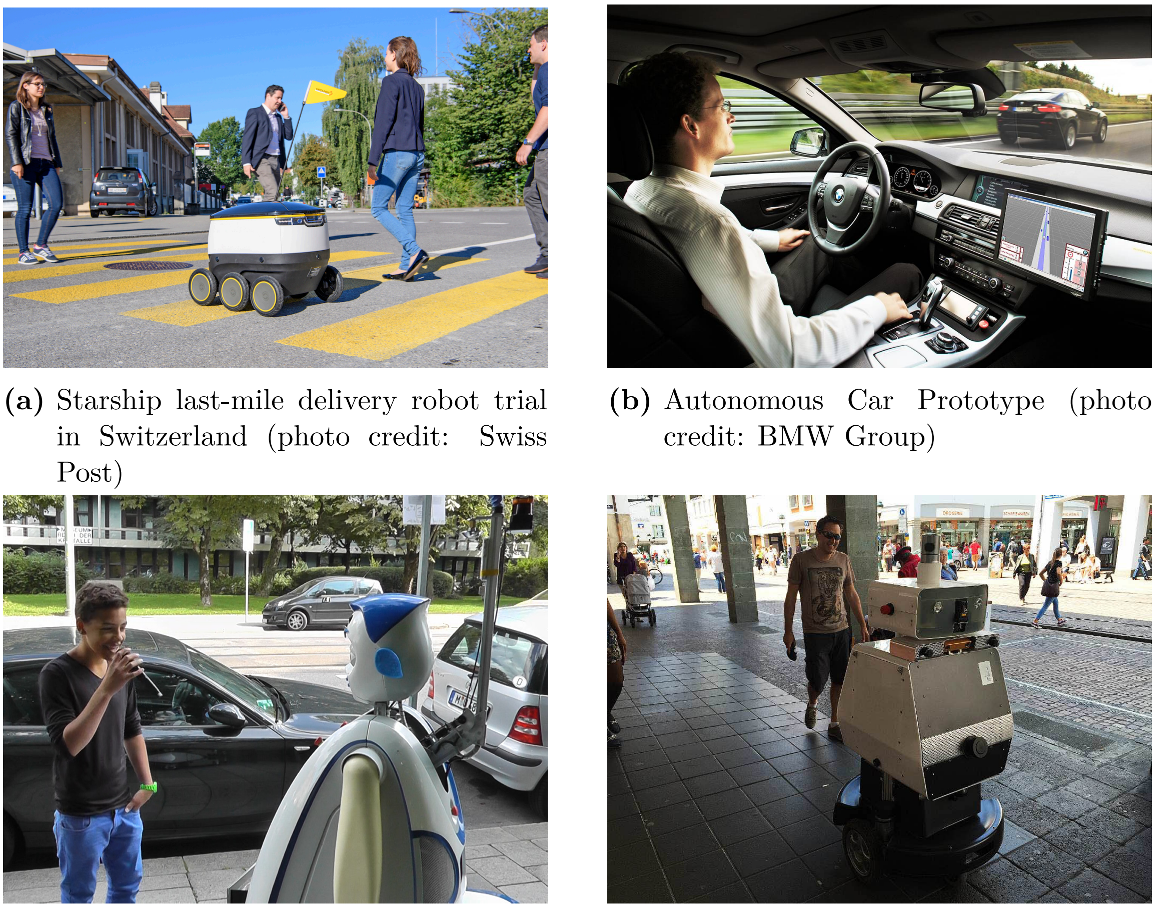
1.1 Semantic Mapping for Autonomous Robots in Urban Environments

It is highly likely that urban environments will be among the first places where consumers and end users come into contact with highly advanced interactive robots. In the assessment of the situation of the robotics industry and research in Europe put forward in the *Robotics 2020—Multiannual Roadmap for Robotics in Europe* [34], almost all application areas for consumer service robotics have components where significant interaction between users and robots happen in an urban setting. For instance, in the rapidly growing domain of healthcare robotics, robotic personal assistants for assistant living will be developed which help humans with limited mobility, autonomy or sensoric abilities with navigating, moving and acting in cities. The application area of autonomous vehicles covers a wide array of services, which, if put to full use, will require autonomous vehicles to have a high level of interactivity and understanding of their environment. Furthermore, logistics scenarios for goods, such as the last-mile delivery of packages, are an application area where robots will operate in the near future in close proximity with humans in everyday city life. Marketing and tourism pose further challenging applications for highly interactive robots acting in inner city environments.

These applications require untrained users to interact with robots without intermediaries. To be usable by and useful to the general public, these robots need simple, intuitive interfaces that enable them to do complex tasks. Both task-related interaction and task execution require knowledge on a semantic level.

Various recent examples from media and research publications illustrate these developments. Different robotics companies have begun to tackle last-mile delivery of goods as an commercial, industrial application. To give one example that received much media attention, the Estonian company *Starship*[1] has developed a robotic platform (Figure 1.1a) for this task which is now being tested under real-usecase constraints in inner cities in

[1] `starship.xyz`



(a) Starship last-mile delivery robot trial in Switzerland (photo credit: Swiss Post)

(b) Autonomous Car Prototype (photo credit: BMW Group)

(c) The Interactive Urban Robot (IURO)

(d) Obelix [96] (photo credit: Badische Zeitung)

Figure 1.1: Examples for robot prototypes operating in urban environments

Europe and the United States. Autonomous Driving is another application which is omnipresent in the media, and where companies such as BMW (Figure 1.1b) see realistic business cases for deploying cars with full or partial autonomy in everyday driving situations, including those in urban environments. Such environments have also received special attention in the robotics research community. Projects like Obelix (Figure 1.1d) and IURO (Figure 1.1c), the latter of which has been closely connected with the motivation for the work presented in this thesis as detailed in Section 1.2, have deployed autonomous robots in the pedestrian space of inner cities to gain insight both about the technical challenges of this environment and the social implications of robots interacting with other users of urban space.

Semantic Mapping, the overarching topic of this thesis, is an integral part of the systems in all these applications. Its goal is to extend the purely geometric environment data, which is indispensable for robotic navigation, by additional *semantic* information, which is for example derived from object classification, from interaction or from common-sense knowledge. Including this type of data in the environment representation of a robot allows it to interpret, reason and communicate with humans about tasks in a high-level way.

Robotic systems can benefit from semantic data in various ways. More advanced reasoning capabilities may reduce the dependence on sensor data or external information sources (like GPS functionality), for example when information can be improved or validated by employing common-sense knowledge. Semantic-level data also allows easier interoperability between systems, and rich information sources which provide man-made and made-for-humans information sources, such as online databases of location-based or common-sense information, can be tapped. Exploiting the richness and diversity of different information sources is important, since the combination of data from various sources will lead to new capabilities. For instance, large advances in computer vision research notwithstanding, the semantic classification of objects and scenes from sensor data, taking into account the context, is still a very demanding and error-prone task. Nevertheless, similar or at least helpful information may already be present in a human-annotated form for human use, or the information may be arrived at through interaction. A wide range of Human-Robot Interaction (HRI) applications are enabled by equipping robots with semantic data, since grounded, situated interaction is only possible given location-specific information in a format that is suitable for humans. Multimodal interfaces, including natural language as a very powerful communications channel, operating on different types of semantics can be used for task specification or clarification.

The importance of Semantic Mapping technology for future commercial use of robotics technology is also illustrated by it being specifically mentioned as a major milestone to the commercialization of service robots in a recent roadmap for Robotics in the United States [26]. Semantic knowledge is also listed as a requirement for the commercialization of a large set of the applications surveyed in the *Robotics 2020—Multiannual Roadmap for Robotics in Europe* [34].

1.2 Motivations from the IURO Project

A large part of the research compiled in this thesis is motivated by experiences gained in the course of the IURO project between 2011 and 2014. While the aim of this thesis is to describe approaches to extend semantic mapping capabilities for a large class of robots operating in urban environments, the scope and goal of the IURO endeavour provide a good context for the general problem of semantic mapping. For this reason, an outline of the project will be given in the following.

The main goal of the project was to create a complete robotic system that is capable of navigating in the pedestrian space of an urban environment, and in doing so only relies on information sources immediately available in its environment. In addition to whatever can be perceived with the robot's sensors, pedestrians with local knowledge in the vicinity were identified as another valuable information source, mainly for route and navigation information. However, other information sources, like precomputed and stored maps or

GPS navigation, would have required the use of external data sources and connections, and were not used in the IURO system. From this setting, the main research areas for the project were identified, which are briefly summarized in the following.

Local and Global Navigation The pedestrian areas of inner-city streets are the main workspace of the IURO robot. Local navigation in this environment requires the ability to avoid static and moving objects, as the space is both shared with pedestrians and cyclists, and unstructured, since obstacles of very different kinds can be encountered along the way. Paramount to the safety of the system is its capability to reliably detect the boundaries of the sidewalk area, and move only within that space. On a global level, navigation solely based on route instructions gathered from humans requires that the given descriptions can be verified against each other, mapped to sequences of landmarks and navigation actions, and these can be grounded in the environment and executed.

Situation Interpretation and Environment Modelling The environment representation of the robot relies on processing data from its sensors. One aspect of this is the assessment whether an areas is safely traversable and free from obstacles. Furthermore, landmarks such as crossings and traffic lights, which are used for navigation and for grounding route instructions, need to be detected in the environment. The way this data is represented to the robot needs to unify the very different aspects of processed sensor data and communication with humans to allow navigating based on route instructions given by humans and comparing and verifying descriptions against each other, and against the environment. Developing building blocks for environment representations satisfying these requirements is one of the main aims of this thesis.

Spoken Language Dialogue System and Action Planning The IURO robot's main source of route information is the interaction with pedestrians. For this communication to be effective, the robot is equipped with a multimodal interface. First and foremost, it is able to both produce and understand spoken language in a dialogue system designed for the efficient and effective elicitation and verification of route instructions. Furthermore, a touchscreen in the body of the robot provides a touch interface for route instruction and for displaying route visualizations for verification. A pointing device behind the head of the robot and its arms can be used to illustrate directions and to clarify the frame of reference the robot is using. The tendency of the robot to advance on the route to its goal, or to obtain or verify new route information, is controlled by a action selection module, which chooses between a set of behavior options based on the value of new information for the estimated success of task completion.

Social Aspects of Proactive Human-Robot Interaction An important novelty in the interaction scenario of the IURO project is that the robot is the initiator of the interaction, since the robot is in need of route information and thus has to ask bystanders or passers-by for help. This type of interaction with interaction partners who are not necessarily experienced with robots or similar technologies requires that social aspects are incorporated into the interaction design. Considerations from social robotics were taken into account for the design of the robot hardware, the design of the spoken language dialogue strategy, and also the navigation behaviors of the robot, with the main goal being to create a friendly impression of the robot that elicits the willingness to help and avoids intimidation in its interaction partners.

The IURO project provided a challenging application for semantic mapping research. The interpretation of route instructions given in natural language requires an internal representation of the environment that bridges the gap between metric information, which is necessary for navigation, and a semantic, symbol-level representation, which is necessary for natural language interaction. Semantic information must be understood from route descriptions as well as verbalized in order to verify the robot's belief about the route to take. Thus, all sources and sinks of information must be mapped to a common environment representation. Although not within the scope of the IURO project, the topic of increasing the diversity of information sources lends it self to being extended to other, possibly online, information source for humans, such as OpenStreetMap or ontology projects that seek to capture common-sense knowledge.

This thesis explores aspects of the large problem of semantic mapping for autonomous, interactive robots in urban environments in detail. The question of how to represent sensor-level data on a human-adapted level is addressed in Chapter 3, where qualitative labels for spatial relations between objects, which can be used in interaction, are integrated into a metric environment representation. The topic of using other data sources, which are originally meant for human use, to aid signal-level processing, is the topic of Chapter 4. This chapter concerns itself with the use of coarse geometric data from the open-source, human-annotated OpenStreetMap to aid the extraction of road geometries from sensor-level point cloud data, which can then be used for higher-level navigation. The third part of this thesis deals with the idea of data sparsity, which is the motivation for Chapter 5. There, it is explored how little information is necessary for a robot to localize on a map that hasn't been visited before, using only building outlines as very salient and robust semantic features for place recognition.

1.3 Thesis Overview, Contributions and Published Work

This thesis covers three major aspects of semantic mapping for autonomous, interactive robots in urban environments:

(i) Integration of qualitative spatial relations with metric 3D point cloud object data using probabilistic logic

(ii) Combination of metric point cloud data with coarse data from crowdsourced maps to improve scene interpretation in the form of estimation of road geometry

(iii) Localization of a robot in unknown urban environments based only on building outlines extracted from a crowdsourced map

These topics are motivated from the experience gained with autonomous robot in urban environments during the course of the IURO project as described above. The outline of the thesis, its individual contributions, and the publications that have been the result of the work in the respective areas are described in the following.

Chapter 2 presents an extensive review over the research in spatial representation for robotics and semantic mapping, with a particular focus on representations for interactive robotic applications. It presents the origins of robotic environment representations in theories about the cognitive abilities and strategies humans use to model space and navigate in it. Furthermore, an overview over qualitative spatial representations and reasoning techniques is given. Finally, different approaches to creating maps for the use by robots are laid out, differentiating by the way geometric information is abstracted, and with specific focus on the integration of semantic data in the map. This review presents a new take on the large amount of existing research on these topics with the specific focus of interactive robots. It was published as part of a review paper [209] that grew out of discussions and collaboration at a workshop at IROS 2015 with the topic of "Spatial Reasoning and Interaction for Real-World Robotics". The work presented in Chapter 2 is built on the part of the review that was written by the author of this thesis.

Chapter 3 is concerned with the robust extraction of qualitative spatial relations from metric point cloud data. The described method uses Markov Logic Networks (MLNs), a probabilistic logic model that has previously been used in different robotics contexts. The chapter first describes probabilistic logic models, and then proceeds to describe a fast inference method for this type of models. This method builds on the conversion from the original logical formulation of the problem to a purely algebraic one with only pairwise interactions in a process called *quadratization*, for which different, highly efficient inference methods based on maximum-flow graph-theoretic computations are available. The method is shown to achieve state-of-the-art results on typical problems from the probabilistic logic literature. The chapter then proceeds to apply the MLN methodology to the problem of spatial relation estimation in a qualitative spatial representation that is geared towards interpreting and verbalising scenes in urban environments. The performance of the method is evaluated on a dataset of challenging urban scenes. The quadratization-based inference method for MLNs was developed in collaboration with Roderick de Nijs and published in [205]. The work on spatial relation estimation is part of the work published in [214].

Chapter 4 illustrates the extension of metric data with semantic information from a different information source, and the benefits the combination of different data sources can offer. In the presented work, a semantic map is extended with information about the street network and street geometry of an urban environment. This information is inferred both from sensor-level point cloud data, and from coarse semantic data about the street network from the crowdsourced, open-source OpenStreetMap database. The street network information is used to leverage a probabilistic model of the geometries of neighbouring street segments. Also in this chapter, an in-depth evaluation on a large point cloud data set is provided, along with an evaluation of the computational properties of the presented algorithms. The method was published in [211].

In Chapter 5, a different aspect of using sparse semantic data from OpenStreetMap is explored by using building outlines extracted from the map as single data source to localize the robot in an unknown urban scene. The method builds on the generic *chamfer matching* template matching technique, which is extended to include visibility analysis in the cost function to model the characteristics of the laser range finder providing the input data. Since the method is independent of the provenience of the input data, the formulation can be expected to generalize to other forms of input data, such as 3D point cloud data from monocular or stereo cameras. The method is evaluated on two large, diverse point cloud datasets of different urban environments, and shown to produce state-of-the-art results in comparison with a baseline method from literature and with the generic chamfer matching approach. The presented approach has been published in [210].

The thesis concludes with a summary of the presented work and the encountered challenges, as well as an outlook on future research directions in semantic mapping for urban environments, in Chapter 6.

2

Concepts in Spatial Reasoning and Robotic Mapping

This chapter presents a review of research in the major fields relevant for this discussion of semantic mapping and spatial reasoning. It first gives an overview over the cognitive theories that have been developed about spatial representations that humans use and that have influenced models used in robotics. Furthermore, an introduction to qualitative spatial representations and the corresponding qualitative reasoning methods is given. Finally, different approaches to creating maps for the use of robots on different levels of abstraction, taking into account quantitative and qualitative geometric as well as semantic information are highlighted.

The work presented in this chapter was published as part of a review paper [209].

2.1 Overview

Truly universal helper robots capable of coping with unknown, unstructured environments must be capable of spatial reasoning, i.e., establishing geometric relations between objects and locations, expressing those in terms understandable by humans. It is therefore desirable that spatial and semantic environment representations are tightly interlinked. Precise 2D and 3D robotic mapping and the generation of accurate, consistent metric representations of space are highly useful for navigation and exploration, but they do not capture symbol-level information about the environment. The latter is, however, essential for reasoning, and enables interaction via natural language, which is arguably the most common and natural communication channel used and understood by humans.

This chapter aims to give an overview about the research on both quantitative and qualitative representations of space for the use of robotic systems. Naturally, this research has been inspired by the way humans and animals represent space and navigate in known and unknown environments. Section 2.2 gives a brief overview over the findings from cognitive science that have inspired and been incorporated in approaches for spatial representation for robots. Humans often reason about space in qualitative terms, and this kind of reasoning and the appropriate representation systems have been formalized in the field of Qualitative Spatial Reasoning (QSR). An overview over the representation systems and reasoning techniques developed in this scientific discipline is given in Section 2.3.

Robotic knowledge about spatial relationships is encoded in maps. They are essential for basic tasks in navigation and localization, but they also constitute the domain for task planning and situation understanding. For the latter tasks, purely geometric information is not sufficient, and additional semantic information needs to be represented to enable robots with these capabilities. Interaction with humans, for example for task specification such as in a route description scenario, requires that the environment representation is available in terms that are easily relatable for human interaction partners. Section 2.4 presents different mapping methods from the literature that address some of these requirements.

Section 2.5 concludes the chapter with a short discussion of the current state of semantic mapping research in a robotics context and the challenges that are being faced.

2.2 Cognitive Models

In order to create functional and efficient abstractions of space for intelligent robots, research has often looked to insights on the way humans and animals organize their spatial knowledge. Spatial representations for technical systems that are close to a human understanding of space are often easier to design and interpret, and facilitate information exchange with humans. In the following section, basic terms and distinctions from the study of human and animal spatial cognition are highlighted that have shown to also be helpful spatial models for technical systems. These ideas from cognitive studies have influenced research especially in hierarchical hybrid and semantic maps, which are discussed in Section 2.4.3 and Section 2.4.4.

Two basic paradigms that have been used to describe human spatial cognition are those of *route* and *survey knowledge* [33, 168]. Route knowledge represents space on a person-to-object basis, with the perspective of the visual system, while survey knowledge represents object-to-object relations at a global, world-centered view [61]. Survey knowledge is often also referred to by the term *cognitive map* [181]. On the lowest level, there is also *location knowledge*, which identifies a single location by a salient configuration of objects, which should be robust against change to reduce the uncertainty of the mental environment model [32, 92].

Corresponding to these levels of spatial knowledge, *frames of reference* are defined. The *egocentric* frame takes the person-centered view, and the *allocentric* frame designates the world-centered view. Additional useful designations of frames are the relative, intrinsic, and extrinsic frames, which stand for a person-centered, object-centered, or global view, respectively [185].

Both route and survey knowledge are acquired when moving through an environment. Once learned, route and survey levels of spatial representation are tied to navigational tasks that they are most useful for. Route knowledge is used when navigating along a known path between identifiable places, where the navigational decisions have to be made at decision points to identify the correct continuation of the paths. On the other hand, survey knowledge is needed for pre-meditated navigational planning, where an unknown route to a target in a known or partially known environment has to be determined before actually executing the plan [195]. Insights on the different forms of spatial representations in cognitive models have influenced the research on hybrid maps for technical systems, in which environments are represented at multiple hierarchically organized levels.

While the ability to perform these tasks shows that both levels of knowledge are accessible, humans generally do not acquire full survey knowledge by exploration. Instead, they store topological relationships along with coarse, imprecise spatial relations between places that enable some Euclidean reasoning, for example about shortcuts through unexplored areas [33]. Experiments have shown that humans do not perform very well on recreating exact Euclidean measurements for known large-scale environments, with recalled distances distorted and affected by properties such as the number of landmarks on a route, and angles between alternative paths generally regressing towards right angles [184].

The non-Euclidean nature of the cognitive spatial model is further illustrated by the observations that recalled spatial relations may depend on an (imagined) vantage point, and that symmetric relations tend to be recalled asymmetrically depending on the properties of the involved objects. Cognitive load expended on retrieving spatial relations is also a factor that allows some insight into the mental spatial representation, which can be seen in some spatial relations being faster to recall than others, and in the fact that recalled spatial arrangements are more accurate when more information is asked for than when only partial information is inquired [184]. Tversky [184] calls the ensuing representation *spatial mental models*, eschewing the term ‘cognitive map’, since its properties are rather different from a standard Euclidean map. The notion of the representation being not fully Euclidean, but topological with added imprecise general spatial relations is corroborated by experiments in Virtual Reality, where participants have no problems navigating in worlds that are physically impossible [87]. For technical systems, formalisms that do not rely on quantitative Euclidean geometry have been explored with qualitative spatial representations as discussed in Section 2.3 and topological maps, which are introduced in Section 2.4.2.

A further important characteristic in the discussion of mental spatial models is their hierarchical nature. Nonhierarchical models rely on all spatial elements being stored at the same level, while hierarchical theories postulate that different areas or aspects of space are organized in different branches of a hierarchy. Hierarchical models can be strongly or partially hierarchical, where the latter permits additional attributes between elements of different branches. Experiments have shown that human spatial memory is likely to be organized partially hierarchically [75, 121].

Another distinction that has proven useful in discussing human spatial cognition is the dichotomy of *propositional* and *imagistic* representations in human cognition [76]. Imagistic representations are common as spatial representation such as maps, sketches, and figures. On the other hand, propositional representations are closer to the way spatial arrangements are expressed with language, and can be computed from imagistic representations.

2.3 Qualitative Spatial Representation and Reasoning

Traditionally, formal mathematical reasoning about space primarily used the tools of topology and Euclidean or Cartesian geometry. While this type of reasoning about metric quantities is essential to many aspects of robotics, the disciplines of robotics and Artificial Intelligence have also developed an interest in a qualitative, symbolic system of reasoning about space. Arguably, a quantitative representation of space is closer to the cognitive and, in particular, the linguistic ways of representing space. Thus, it can bridge the gap between physical space where robots operate, and common-sense space, which is commonly addressed by language. Deliberate quantization can also bring robustness against noise and parameter errors. Dealing with metric values can also bring a degree of unwanted precision in the presence of uncertainty or in interaction scenarios. Finally, qualitative reasoning can be beneficial in terms of memory and computational complexity.

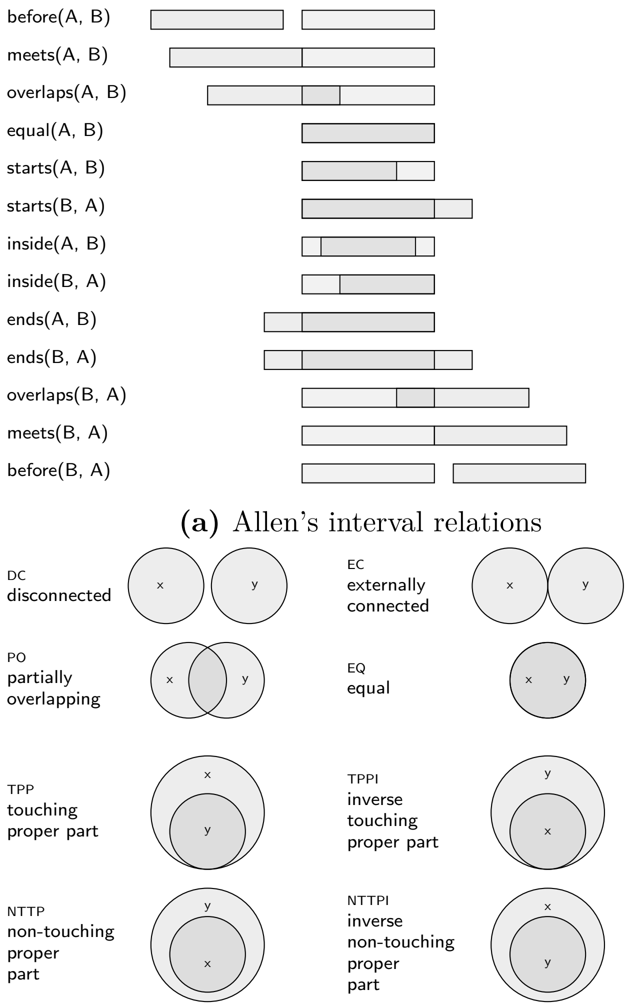
2.3.1 Qualitative Representations of Space

The aspects of space that need to be represented by a specific representation depend on the application, and many different formalisms for different requirements have been developed in the Qualitative Spatial Reasoning community.

A spatial representation consists of a set of basic spatial entities, and the relations that can be defined between them. Basic entities can be points, lines, line segments, rectangles, cubes, or arbitrary regions of any dimension. The *dimensionality* of the basic entities and the space that is being modeled depends on the modeling depth and the application as well: As a practical example, a road is one-dimensional for trip planning, two-dimensional when planning overtaking behavior, and three-dimensional when trying to estimate the curb position.

For brevity, the focus here is on representations used or usable for robotics. The basics of reasoning systems will be mentioned; more in-depth treatments can be found in the review articles by Vieu, Cohn and Renz, and Chen et al. [31, 39, 188].

Mereotopology An important set of qualitative spatial representations is based on the topology, i.e., relations of connectedness and enclosure, and mereology, i.e., the relations of parthood, of basic entities. These are known as mereotopological representations. In the following, some important instances of these formalisms will be briefly introduced.



(a) Allen's interval relations

(b) RCC basic relations

Figure 2.1: Mereotopological calculi

As a very basic reasoning system, the *point calculus* for scalar values defines the relations <, =, >. The *interval calculus* [2] extends the reasoning in a single dimension to intervals, originally for reasoning about intervals in time. The 13 resulting binary relations are illustrated in Figure 2.1a.

This type of reasoning is extended to two dimensions to form the *Rectangle Algebra* [13, 66]. For this type of representation, shapes are projected to the axes of an extrinsically defined coordinate system, and the relations between the resulting intervals are constructed separately for each axis. It can be noted that this representation not only conveys topological information, but also has an orientation component. The spatial representation defined by the interval calculus has been analyzed for cognitive adequacy by Knauff [88], who has shown that this representation aligns well with cognitive models.

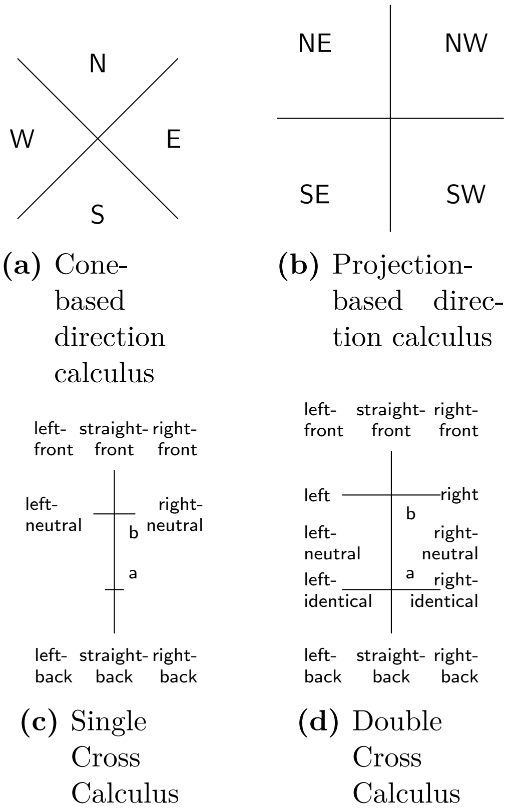
Another important representation, which builds entirely on the notion of connectedness between regions, is the *Region Connection Calculus* (RCC) [145]. The canonical set of eight relations between two regions that can be defined using connectedness, which is known as RCC-8, is shown in Figure 2.1b. Based on this set of base relations, different reasoning systems are possible depending on the intricacies of handling open and closed sets. Easier calculi are possible when border regions are not considered explicitly for reasoning [39]. A reduced set of base relations that does not take the boundary of regions into account comprises the five relations EQ, PO, PP (subsumes TPP and NTPP), PPI (subsumes TPPI and NTPPI), and DR (subsumes DC and EC). The cognitive plausibility of RCC-8 has been evaluated by Renz and Nebel [151] with the result that test subjects cluster pairs of regions according to the topological information it represents; thus showing its cognitive adequacy.

RCC can also serve as a good example for the concept of *conceptual neighborhoods* [54]. These define a system of neighborhood for relations in a reasoning system, as opposed to a system of neighborhood of objects. The conceptual neighborhood of a relation contains all those relations that can be reached directly through transformations of one of the involved objects. For example, in RCC-8, the cognitive neighborhood of the EC relation consist of DC and PO, but none of the other five relations. The conceptual neighborhood depends on the transformations that are allowed, but can usually be used to limit the complexity of reasoning in a system, in particular if the reasoning entails the movement of objects.

Orientation calculi Mereotopological relations are important for qualitative modelling of space; however, the information they can represent is limited. In the following, some simple representations that focus on orientation and direction between two objects, a primary object and a reference object, are introduced. For reasoning about orientation, a *frame of reference* is necessary. This can be either extrinsic, such as the cardinal directions given by a compass, or intrinsic to the problem. Frank [53] presents two spatial calculi based on cardinal directions: a cone-based one, where the angular direction towards the reference object is rounded to the nearest cardinal direction, and a projection-based one, which overlays the two pairs of half-planes associated with the two pairs of opposing cardinal directions. Both divide the plane by two lines intersecting in the reference point. They are illustrated in Figure 2.2a and Figure 2.2b, respectively. A generalization of this representation to an arbitrary number of lines is the Star calculus [150].

The single cross calculus and the double cross calculus [55, 192] are example for relative orientation representation, where orientation is given as a ternary relation between a point on the plane, the *referent*, and the oriented line segment defined by the *origin a* and the *relatum b*. These representations are illustrated in Figures 2.2c and 2.2d.

The Cardinal direction calculus (CDC) [62] is a representation for relations between two extended regions in the plane. For the primary region, the minimum bounding rectangle is determined. The continuations of its edges separate the plane into nine sections, and the relation to the reference region is given by the set of sections the reference region



(a) Cone-based direction calculus

(b) Projection-based direction calculus

(c) Single Cross Calculus

(d) Double Cross Calculus

Figure 2.2: Orientation calculi. 2.2a and 2.2b are binary calculi; 2.2c and 2.2d are ternary. For the latter two, the origin is denoted by a, the referent by b, and the relatum can be any point in the plane.

intersects with. This is usually written as a 3 × 3 Boolean matrix, where each element indicates the non-emptiness of the corresponding intersection.

Other relations: size, distance and shape More predicates can be introduced to describe other aspects of objects or tuples of objects like relative or absolute distance, size, or shape. Distance and size properties are often based on quantization into a small number of categories like *far* or *close* or relative to other objects, as in a predicate $\mathit{Closer}((o_1, o_2), (o_3, o_4))$, which compares the distances of two pairs of objects. Reasoning about the shape of objects is a more recent development in qualitative spatial reasoning. The high complexity of most approaches and formalisms has led to only very simple formalisms being adapted into robotics applications, mostly based on representing objects by their centroid as a single point, their convex hull or a minimum bounding rectangle.

More recent work has focused on combining reasoning mechanisms from different calculi to jointly reason about different aspects of a spatial arrangement, e.g. topology and orientation using RCC-8 and RA or CDC simultaneously [38]. Another approach at combining reasoning about orientation and distance is the ternary point configuration (TPCC) calculus [125], which separates the plane into eight radial segments based on orientation with respect to the origin-relatum line segment, and additionally qualifies distance of the relatum to the referent as greater or smaller than the distance between origin and referent.

2.3.2 Qualitative Spatial Reasoning

Qualitative Spatial Reasoning is tightly connected with methods and results from mathematical logic. Reasoning systems can be formulated as *axiomatic systems*, which are generally first-order [188]. Due to the high complexity of reasoning in axiomatic systems, most spatial reasoning systems are defined as *relational algebras* or *calculi* [108]. These define a finite set of qualitative relations as described for various representation systems above. Usually, this set of base relations is required to be jointly exhaustive and pairwise disjoint (JEPD). If there are multiple possible base relations between a pair or tuple of objects, their relation is described by the disjunction of the individual base relations, which is generally denoted by the union of these relations. The full set of possible relations is the power set of the base relations, but it can also be restricted further, e.g., to ensure tractability. In addition to the relations, two important operations need to be defined to enable reasoning with a spatial algebra. For a binary calculus, the *converse* operation defines the relation S that holds for the pair (x, y) if relation R holds for the pair (y, x). The *composition* operator defines the relation for the pair (x, z) if the relations for pairs (x, y) and (y, z) are known. For many calculi, compositions of pairs of base relations are given in *composition tables*, which allow to determine the composition of arbitrary relations as the union of the compositions of the contained base relations by a simple table lookup. For ternary calculi, corresponding ternary operators have to be defined.

Different spatial reasoning problems can be posed. An important reasoning problem is the question whether there is an arrangement of objects that fulfills a set of given relations, which is known as *consistency checking* or *satisfiability.* From a computational standpoint, the consistency checking problem is a convenient choice, since many other decision or counting problems can be converted to this problem with polynomial complexity, and it has been studied for a long time for general-purpose logical formulations. Among these related tasks is the problem of finding one or all variable assignments that conform with a given constraint network, removing redundant constraints, or deciding whether a constraint network can be realized in a particular dimension, for example on a plane. For a propositional algebra, the consistency checking problem can be posed as a constraint satisfaction problem, and the corresponding methods from literature can be applied. The more restricted structure of spatial problems, as compared to general problems in logical formulations, allows to make simplifications to the reasoning process, which make the reasoning more efficient than general logical inference.

In many cases, the operators defined for the relations can be used in constraint propagation algorithms such as *path-consistency* and *algebraic-closure* [39] to decide the consistency of a constraint network. For decidable calculi, the complexity of inference is an important characteristic. For most calculi, deciding consistency is NP-complete, for example for the interval and rectangle algebras, as well as for RCC-8 and RCC-5 [39]. Reasoning problems in relation systems that are able to distinguish left from right is NP-hard [115, 198], since in these cases, local consistency algorithms such as algebraic closure cannot decide consistency of a global scenario.

Research has been directed towards improving the practical applicability of the algorithms based on local consistency by trying to identify (maximal) tractable subsets of existing calculi, which make the backtracking search in algebraic-closure based constraint processing more efficient. This can, for example, be done by searching for subsets that can be expressed using Horn clauses.

While generic constraint programming and logical inference tools can be used for spatial reasoning with the formalisms mentioned above [196], a number of specialized software toolboxes specifically for QSR have been developed. Among them are SPARq (Spatial Reasoning done Qualitatively) [199], GQR (Generic Qualitative Reasoner) [59], PelletSpatial [175], CHOROS [35] and the Qualitative Algebra Toolkit (QAT) [40].

Wolter and Wallgrün [199] list some applications other than satisfiability checking via constraint processing that have practical relevance. Among these is *qualification*, the translation of a quantitative description of a scenario to a qualitative one considering rounding errors and noise, and the process of producing a (cognitively valid) rendition of a qualitative scenarion, e.g., for visualization. The qualification problem has also been addressed in the context of machine learning. For example, the work presented in Chapter 3, which describes a system based on Markov Logic Networks that estimates relations between objects in an annotated map of an urban environment, can be seen as an instance of a qualification problem. The approach put forward by Sjöö et al. [173] relies on a Graphical Model to determine the relations ‘On’ and ‘In’ between everyday objects. Support relations between objects in household scenes are the result of the estimation process performed by Silberman et al. [169].

2.4 Mapping in Robotics

For an overwhelming majority of robotic tasks, robots need to develop and keep a representation of their surroundings based on sensor readings and possibly prior knowledge. Independent of the actual properties of this representation, this field of research is known as mapping. This section will give a brief overview of the different types of maps used in robotics, with a focus on representations that are wholly or partially qualitative in nature, and those that have a semantic component.

2.4.1 Metric maps

Learning and maintaining a metric map is central to many robotic tasks that rely on navigation. Based on early work by Smith and Cheeseman [174] and Leonard and Durrant-Whyte [103], the probabilistic formulation of the problem of building a globally consistent

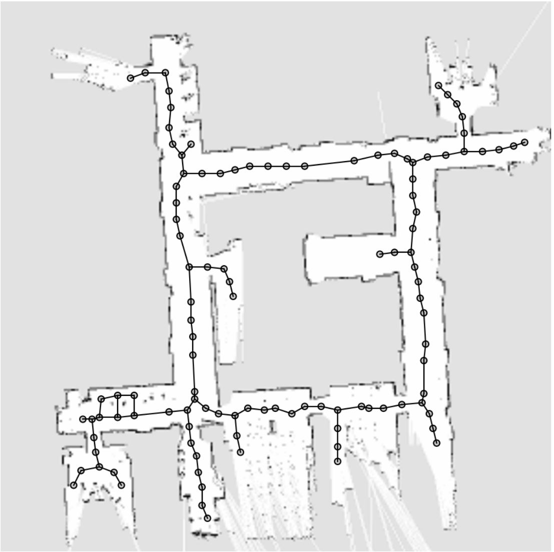


Figure 2.3: Metric map with overlaid topological structure. The metric map is generated with a SLAM algorithm on laser data. For the topological map, the structure of the environment is extracted from the metric map as the Voronoi graph, and edges are placed at junctions of the Voronoi graph as well as at constant intervals between junctions.

map has become known as the Simultaneous Localization and Mapping (SLAM) problem. Data from a very diverse range of sensors such as cameras, sonars, laser sensors, odometry, and GPS needs to be integrated over the course of potentially long exploration runs of a robot. A basic distinction between SLAM approaches is whether they are filter-based or graph-based. Filter-based SLAM emphasizes the temporal aspect of consecutive sensor measurements, while the graph-based variant emphasizes the spatial aspect by adding spatial constraints between robot poses where landmarks are jointly visible [65]. The underlying representation for the metric map can vary independently of the SLAM formalism, from landmark-based formalisms that store the positions of salient features in the environment to low-level representations like occupancy grids [64], surface maps [183], or raw sensor measurements like point clouds. A central challenge in SLAM is the data association problem of aligning real-world features across multiple sensor measurements. A good solution is important when the robot revisits a location where it has been before *(closing the loop)*, where a wrong association of features leads to an inconsistent map.

An example of a metric map generated using a SLAM algorithm from laser sensor data is shown in Figure 2.3.

2.4.2 Topological maps

Topological maps represent environments using a graph, the nodes of which represent *places* in free space, and edges denote traversability or connection in free space between pairs of nodes. There are different approaches to defining the notion of places. One common approach is to define nodes in the topological map for each distinct part of the environment, separated by gateways such as doors or entryways. Other approaches define nodes every time the robot has traveled a fixed, specified distance, or use the structure of the *Generalized Voronoi Graph* [9]. An example of a topological map overlaid over a metric map of the same environment is given in Figure 2.3.

Like the problem of loop closure in metric mapping, topological mapping also faces the problem of identifying a place that is being revisited by the robot. This is known as the correspondence problem, which is made difficult in environments where possible matching candidate places look exactly or approximately the same in the available sensor data, which is known as *perceptual aliasing*.

An axiomatic theory and full ontological definition of topological maps was presented by Remolina [149]. Map learning is accomplished in a purely logical fashion by using nested abnormality theories, which use causal, topological and metrical properties of the environment to determine the topological map as the minimal map that explains the robot's percepts.

For topological maps, the space of maps is combinatorial, but still much smaller than the space of all possible metric maps. Thus, multi-hypothesis or probabilistic methods that keep a distribution over all possible hypotheses are possible. The probabilistic topological map [147] keeps a distribution over all possible topologies using a Rao-Blackwellized particle filter. Wallgrün [191] presents a topological mapping algorithm that exclusively relies on qualitative spatial reasoning to keep track of multiple hypotheses about the structure of the environment. Two different qualitative reasoning calculi are compared on the task of building a consistent map from sparse qualitative connection information, using various constraints on the spatial structure of the resulting network to reduce the size of the search space. An extensive review of SLAM in topological maps is presented by Boal et al. [20]

A topological map is also a convenient and efficient representation of environments for route-based navigation. The *route graph* [194] is a topological map designed for this purpose. Its nodes are *places* connected by *courses*, which together make up *route segments* and entire *routes*. Elements of the route graph can be labelled with additional information to convey categories such as the medium of transport to be used on a particular route segment.

2.4.3 Hybrid maps

Each type of map has its own strengths, and the term hybrid maps describes approaches that combine different representations to form a stronger overall environment representation. Buschka and Safiotti [25] define a hybrid map as a tuple of maps, where usually one is metric and one is topological. The benefit of the hybrid maps comes from links between the two, which maps objects from one map to objects of the other. Other combinations are possible, however. Some advantages of this combination of different maps are improved loop closure, lower complexity, improved localization, easier planning and high-level reasoning, and the possibility to define a system state on different levels. A particular benefit for hybrid maps can be the possibility to relax the requirement for global consistency of metric representations, and keep a consistent topological representation instead, which can have computational advantages.

An early instance of hybrid maps that has received much attention is the *Spatial Semantic Hierarchy* (SSH) [93, 94], which is inspired by cognitive studies about human spatial representations. It models space on four levels, where each level depends on information from the levels below: The lowest level is the *control level*, which defines a dynamical system where *distinctive state*, known poses in the environment, can be reached by hill-climbing, and trajectories between these states or their attractor regions can be followed. Sensor percepts, so-called *views*, allow the unique identification of these states. On the *causal level*, a finite state automaton is defined, in which state transitions correspond to movements between places. The states and edges of this automaton map to places and paths on the *topological level*. Finally, the *metric level* stores a geometric representation, such as occupancy grids, for each place, which can be combined to form a global metric map. Not all levels must be present or available at all times, depending on whether the region has been explored, availability of computation resources, sensor data etc. The hierarchical structure of the SSH is illustrated in Figure 2.4.

This formalism was extended with ideas from the SLAM community to form the *hybrid SSH* [95], where local maps are used instead of views to identify places locally. This allows more tolerance for noise and dynamics in the environment in small-scale space (within the sensor horizon), but does not require loop closure in large-scale space, where the topological representation can be used. Beeson et al. [16, 17] integrate semantic aspects in the hybrid SSH by reasoning about gateways and integrating the approach with a natural language interface.

2.4.4 Semantic Maps

While metric and topological maps only describe the spatial arrangement of an environment, additional information is necessary for many robotics tasks. Semantic maps broaden the scope of the elements represented in a map to instances of objects, their categories and possible attributes, and to common-sense knowledge about entities represented in the map [99]. This is particularly beneficial in applications where a higher-level understanding of scenarios is necessary, and when applications require human-robot interaction.

Semantic mapping requires that information about the objects in an environment is available for reasoning. Like the spatial information represented in metric maps, this information is often inferred from typical sensor data, coming from 2D and 3D sensors including sonars, LIDAR scanners, monocular, stereo and omnidirectional camera setups and RGB-D sensors. For building semantic maps, high-level techniques like character recognition [28], interaction with humans or databases of common-sense knowledge are used as additional modalities. While the use of non-technical information sources, such as

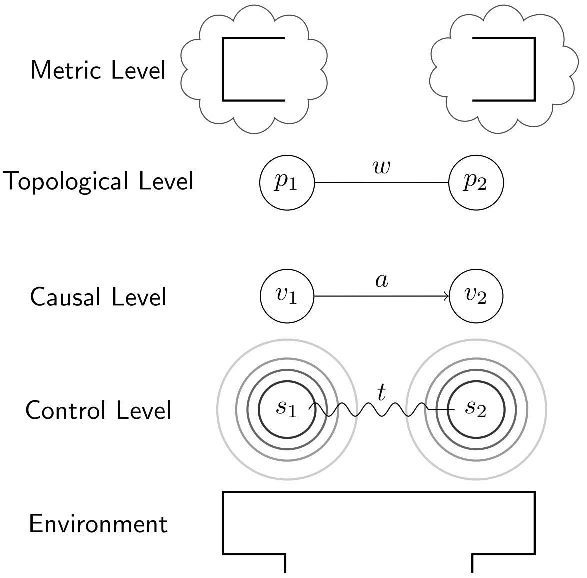


Figure 2.4: Illustration of the Spatial Semantic Hierarchy. The environment is represented by two distinctive states s on the control level, which each have a region of attraction and are connected by a trajectory t. On the causal level, the distinctive states can be identified from sensor percepts with the views v, and transitioning from one state to the other is possible by taking action a. On the topological level, the environment has two places p, which are connected by a path w. On the metric level, a local metric representation for each place can be stored.

human-machine interaction through natural language, are an active research topic [208, 182, 193], this section focuses on the use of purely technical sensor streams for semantic map building. Advanced perception algorithms for object detection, segmentation and classification have been adapted from the robotic perception and computer vision literature and developed specifically for semantic mapping. This area of research is out of scope for this overview, which instead focuses on the mapping and representational aspects of semantic mapping. An overview of perception approaches to semantic mapping is given by Kostavelis and Gasteratos [90].

There is a broad range of different types of semantic information in maps, depending on factors like the intended application, the sensor repertoire of the robot, and the type of environment that is being mapped. A broad categorization can be made between maps that add semantic attributes to objects in the map, maps that categorize regions, and those that add semantic categories to sensor percepts on the trajectory of an exploration of the environment.

Object-based Semantic Maps

The first category of semantic mapping approaches relies on techniques for scene interpretation to label objects in the robot's sensor stream and localize them using a metric environment representation. In this vein, Limketkai et al. [109] label line segments in a metric map as *wall*, *door* or *other* using a relational Markov Network that uses unary and pairwise as well as higher-order spatial relations between objects as input. Nüchter and Hertzberg [134] use a constraint network expressing common properties of spatial arrangements of planes in buildings to classify points from a point cloud into different categories (ceiling, wall, floor, etc.). Additionally, other objects like humans and printers are detected and classified, forming a semantically annotated 3D point cloud. A more perception-oriented approach is presented by Meger et al. [122], where objects detected and classified based on camera images are mapped into their locations in a global occupancy grid. A place categorization method based on object co-occurrence statistics and clustering of objects to places based on spatial distance and a Bayesian criterion for the number of clusters is presented by Viswanathan et al. [189]. In order to capture additional information in the map, and to improve object recognition results, it can be useful to not only enter object information, but also relations between objects explicitly into the map [105].

Region-based Semantic Maps

Many semantic mapping approaches discretize space to a topological map on some level of their hierarchy of maps into areas of conceptual meaning, which are often called *places*. One distinguishing factor between semantic mapping approaches is the way places (or generally nodes in a corresponding topological formulation) are separated.

Some mapping systems recognize that distinct places are usually separated by gateway structure like doors, and devise ways of identifying these structures. The work by Vasudevan et al. [186] builds on a method to recognize objects and doors. A probabilistic relative object graph tracks object positions relative to the place they are found in, and allows to compute probabilistic spatial relations between them. These graphs are used for place recognition, and classification of places is based on the types of objects present in the scene. An extension of the approach [187] uses spatial information even in the reasoning about place categories, where the category of objects, the number of occurrences and simple spatial relationships between them are taken into account when classifying rooms into different categories. Places can have hierarchical structure, so places that afford particular functions, such as a 'printer area' or a 'couch area' can be contained in a more general place of type 'office'. A more detailed subcategorization of gateways is part of the approach by Rituerto et al. [155], which distinguishes the categories *door*, *stairs*, *elevator* and *jamb*. Ranganathan and Dellaert [148] use objects to model and recognize places, where object classifiers are learnt in a supervised manner for recognition. For localization, the camera pose is reconstructed based on the positions of the objects recognized in the environment.

Other work on segmenting metric maps of indoor environments into semantically meaningful clusters is based on a semi-supervised scheme employing a Markov process model [110], spectral clustering on a graph that encodes visibility between randomly sampled free space points in its edges [24], clustering based on mutual information [111] and fitting models of basic room shapes in a Markov Chain [113].

Pronobis et al. [143] present an approach for semantic mapping where low-level classifiers are used to determine properties of areas such as room shape, size, or the existence of certain objects, which are then used to determine room types in a probabilistic reasoning step through inference in a chain graph. Later work [141] includes this technique in a complete semantic mapping system for indoor environments. It accepts multimodal sensor input, including input from humans via natural language, which is treated as a separate sensor modality with an appropriate sensor model. For mapping an environment, first a global metric and topological map are built. Places are created at constant distance intervals on the trajectory of the robot, which are further clustered into rooms separated by door places.

Semantic Maps from Segmenting the Robot Trajectory and from User Interaction

Environments can also be segmented into semantically distinct regions in an online process by recognizing significant changes in the surroundings of the robot while it is exploring the environment. Mozos et al. [126] use a boosting classifier in combination with a hidden Markov model to segment the trajectory of the robot into contiguous segments, where the surrounding environment corresponds to a place. The same classifier together with probabilistic smoothing techniques is used to cluster an occupancy grid into areas of semantic meaning.

Sünderhauf et al. [176] create a semantic occupancy grid by classifying camera data with a convolutional neural network and propagating the classification results along laser beams similar to the probability update in a standard occupancy grid. A number of other approaches rely on classifying and segmenting environments based on the stream of images from the robot's sensors. A topic modeling approach is used by Murphy and Sibley [127], while Ranganathan & Dellaert [146] use an information-theoretic approach. A string encoding of appearance features is used for segmentation of places by Tapus and Siegwart [177].

A segmentation of an environment can also be determined through user interaction. Thrun et al. [180] determine distinctive places by having users push a button to communicate that the robot has arrived at a distinctive place. Nieto-Granda et al. [130] define the assignment of places to the environment as a mixture-of-Gaussians distribution, where the centers of the individual components are taught by human interaction partners during a tour of the surroundings.

Ontologies and High-Level Reasoning

High-level reasoning about the map and its elements requires the robot's understanding of task-relevant concepts as they are used in human reasoning and in language in their own right, and their connection to the corresponding sensor impressions, which is one aspect of the *symbol grounding problem* [71]. A common trait to many approaches that combine metric or topological mapping with reasoning on higher-level concepts is the introduction of an ontology, where world knowledge is stored in a taxonomy and sensor experience from the map is encoded to domain knowledge, which are then linked based on overlapping semantic attributes. Zender et al. [202] present one instance of such an approach, where ontological reasoning complements a multi-level spatial map to form a conceptual representation of an indoor environment. The ontology is handcrafted to represent different room types and the typical objects present in them. Grounding instances of places and detected objects in the environment allows to refine knowledge about the environment, and to generate a linguistic representation of a scene, for example for clarification dialogues. Hawes et al. [72] builds on this mapping approach to build a system that can identify, reason about and autonomously fill gaps in its knowledge about the environment, both its structure and conceptual knowledge as well as semantic knowledge such as room categories.

The multi-hierarchic semantic map for indoor environments presented by Galindo et al. [57] maintains hierarchical representations both for spatial and for semantic knowledge, where the latter takes the form of an ontology. The bottom level of the spatial hierarchy is made up of an occupancy grid, which is segmented into rooms using image processing techniques to form a topological map. Based on properties of the rooms and objects found in them, regions can be classified and anchored to the corresponding concepts in the ontology, and further reasoning can be performed based on the world knowledge stored there. Tenorth et al. [179], Pangercic et al. [135] and Riazuelo et al. [152] introduce semantic mapping approaches which link objects detected in the environment to a large database of common-sense, probabilistic knowledge including high-level attributes like affordances or object articulations, which allows to execute high-level plans like ‘clear the table’. A different type of world knowledge is tapped by works that use the large-scale structure of buildings to determine the function of rooms by their typical topology or by conditioning classifiers on the type of building [10, 116, 117].

Outdoor Semantic Mapping

While the research in semantic mapping has primarily been directed towards the application in indoor environments, outdoor environments have been addressed as well, using a similar array of techniques. Lang et al. [99] apply a multilevel spatial representation along with ontological reasoning to urban outdoor environments. Multiple other methods to add semantic labels to metric maps of urban road environments have been presented, e.g., [42, 73, 139, 140, 165, 166]. Singh and Košecká [170] put the focus on their work on using semantic features extracted from images to cluster street segments into similar categories for segmentation, and using these features for the detection of intersections in cluttered inner cities [170]. Semantic categories can also be used to determine the dynamic properties of parts of the map, which helps to keep track of changes when revisiting places where certain objects of dynamic classes have moved, while static objects can be assumed to remain in the same place over time [45]. Semantic categories can also be useful for autonomous airborne vehicles [164], for example for avoiding obstacles such as as trees or buildings, and for marking certain object classes as possible targets or points of interest. An exemplary use for semantic maps in an outdoor context is given by Drouilly, Rives, and Morisset [43], who extend the idea of route planning based on a task description containing objects as landmarks to an urban domain.

While the mentioned approaches present techniques for describing outdoor environments with semantic attributes in general, a large share of the scientific effort is directed specifically towards the application of autonomous driving. A topological description of large-scale outdoor environments, augmented with semantic information relevant to the task of for off-road driving, is defined by Bernuy and Ruiz de Solar [18]. Wolf and Sukhatme [197] create a terrain map of a robot's driving surface that is annotated with semantic labels, and includes traversability information [197]. In addition to common appearance features for static environments, observed dynamics are included as activity measurements to distinguish different environments in that work. A similar approach [98] uses CRF-based terrain classification both for traversability analysis and for localization. Grimmett et al. [63] developed an hybrid map for the application of automated car parking that combines metric information for navigation with semantic information about parking space locations, pedestrian crossings and safe driving speeds [63]. Special focus is put on the map being adaptible, which means that it is able to keep information consistent when the map is updated partially on the arrival of new information. Further information about urban environments can be gathered by observing the behavior of pedestrians. Qin et al. [144] present work in which places with pedestrian activity, such as pedestrian crossings or subway exits, are classified based on observed pedestrian trajectories. Semantic categories assigned to objects can also be used for localization in urban scenarios [4]. Chapter 5 describes in detail a system that localizes a robot in building maps based on building shapes alone, and gives an in-depth discussion of the scientific work related to that problem.

2.5 Conclusion

This chapter has attempted to give an overview over the different aspects of the scientific field of semantic representations of space in robotics. As it can be seen from the large variety of topics involved in this discussion, it is a highly multidisciplinary field, which takes inputs from engineering, mathematical modeling and reasoning, cognitive science and psychology, cartography and multiple other fields. Its applications extend over the realm of robotics to intelligent services for logistics and transportation, pedestrian and vehicle navigation, and location-based services, which all benefit from map-based environment information.

As the overview has shown, there is no single, unique semantic mapping solution which fits all applications in this spectrum, and it is hard to imagine that there will be any in the future. Instead, many application-specific solutions have been developed, which fit the need of the problem in terms of the contained information, the structure and source of the input data, the internal representation and the way that information is presented to other systems. In particular, tailored solutions are necessary to retrieve the needed semantic information from the environment. Improvements in this respect are to be expected from the development of powerful, standardized methods for scene understanding, object recognition and classification with the recent scientific advances in machine learning research, in particular in deep learning. This lack of generalizable, easily applicable solutions for semantic mapping, and particular the lack of a standard, widely usable 'off the shelf' software, causes a relatively high development effort for robotic systems that are to use semantic knowledge about the environment. This is in contrast to the development in metric mapping approaches, where the advances in the SLAM community have produced software packages that are readily available and can be used with little customization for a wide array of applications. Nevertheless, semantic mapping has enabled robots to achieve tasks that would not have been possible otherwise, especially in the area of establishing a common grounded environment representation for robots and human interaction partners. It has been shown multiple times that higher-level, semantic knowledge can be beneficial for the performance of traditionally lower-level algorithms like metric mapping or classification.

3

Estimating Spatial Relations with Probabilistic Logic for Human-Robot Interaction in Urban Environments

Since spatial and semantic reasoning are tightly linked to the sensor perception information, it is desirable that all types of information are integrated in a joint environment model. This chapter presents the interplay of a novel environment representation called Semantic Rtree (SRTree) and Markov Logic Networks for reasoning about qualitative spatial relations between objects. The SRTree is a semantic occupancy grid based on the hierarchical Rtree data structure that models the occupancy of each grid cell and assigns a class label to it. The main advantages of the proposed approach are 1) a hierarchical representation of large scale outdoor urban environments, which 2) captures both quantitative (metric) and qualitative (semantic) aspects of the environment and allows reasoning in a single data structure, and 3) the capability of dealing with higher-order spatial relations. The proposed methods are experimentally evaluated on a large scale 3D point cloud dataset of downtown Munich enhanced by RGB image data.

In addition, this chapter also presents a novel inference method for the most probable explanation (MPE) task in Markov Logic Networks (MLNs), which is based on a conversion of the original logic formulation to a purely algebraic pseudo-Boolean formulation. Subsequently applying a quadratization method allows the application of efficient inference methods such as Quadratic Pseudo-Boolean Optimization (QPBO). Experiments on standard problems from the Statistical Relational Reasoning literature show that the approach performs very well with respect to other state-of-the-art inference engines.

The work on spatial relation estimation is part of the work published in [214], and the quadratization-based inference method for MLNs was published in [205].

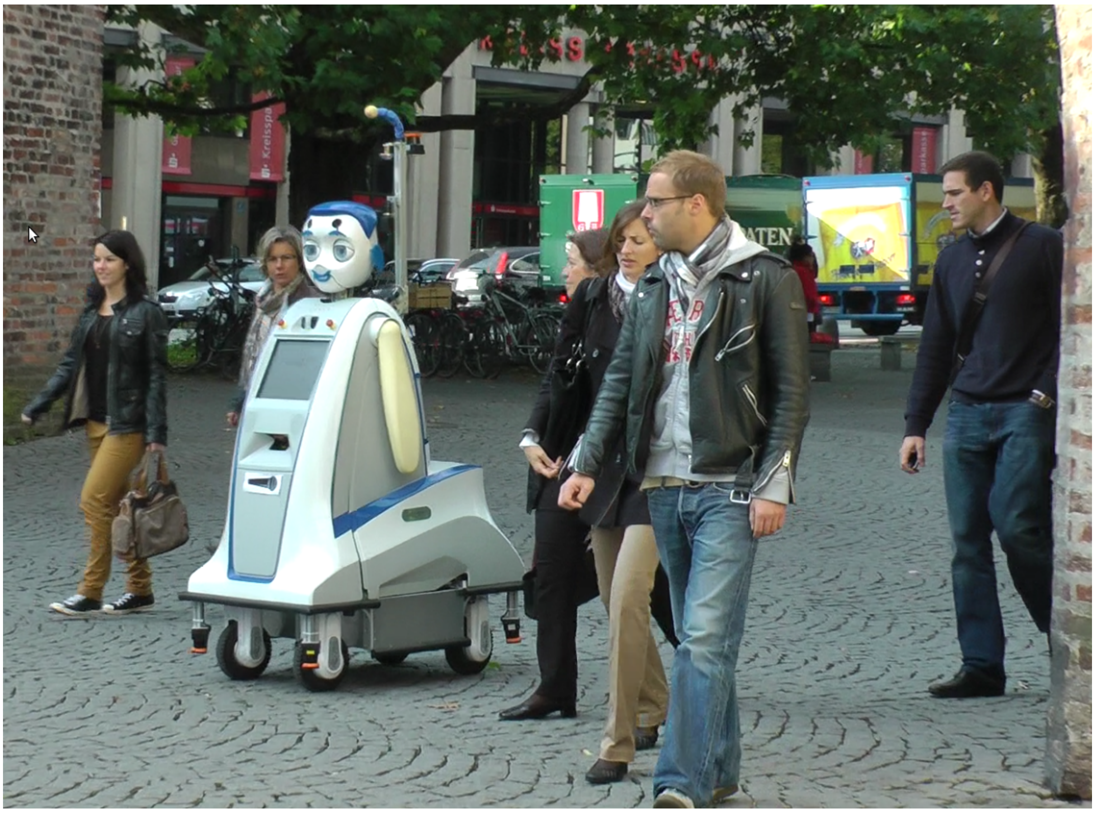


Figure 3.1: The IURO robot in unstructured urban environment.

3.1 Motivation, Problem Statement, Related Work

As described in Section 1.2, the Interactive Urban Robot (IURO) project was formed to address some of the challenges towards the goal of robots as universal helpers being able to autonomously act in unstructured, dynamically changing environments. The goal of this project was to create a robot that is both able to navigate in an unknown urban environment and interact with human passers-by in order to retrieve information. The robot can be given a designated goal location in a city and successfully finds its way to this location without the use of prior map knowledge or Global Positioning System (GPS), obtaining and interpreting directions by asking pedestrians for the way. Building an autonomous interactive robot requires the design of a cognitive architecture that ties together functional modules for navigation, environment perception and interaction. The interaction has to be natural and intuitive for the humans, as they are picked autonomously by the robot, have not had prior contact with robotics technology, and are not instructed prior to the interaction.

In this scenario, a fundamental ability of the robot is to integrate environment information from multiple sources—particularly from sensors like cameras and laser range finders as well as verbal and gesture information from interactions with humans—into a concise environment representation. To this end, the robot must be capable of spatial reasoning, i.e. establishing geometric relations between objects, in terms understandable by humans. It is therefore desirable that metric and semantic environment representations are tightly interlinked.

The field of 3D robotic mapping has received a lot of attention in the scientific community. The most commonly used approach is an occupancy grid, which divides space into grid cells and estimates the probability of occupancy of each cell [46, 77, 160, 161]. These representations can be useful for navigation and exploration; however, they do not capture symbol-level information about the environment. To develop autonomous interactive robots, robots must be capable of understanding the semantics and relationships between the objects in the environment. Recently, the focus of the robotics community has shifted towards semantic representations [123, 131, 142] and object relation modelling in semantic maps [3, 101, 109, 135, 137, 179] as outlined in Chapter 2. The majority of the works mentioned rely on point clouds or operate on the level of objects to represent the semantics of the environment, which is not suited for navigation. Occupancy grids are suitable for navigation; however, they do not capture the environment semantics. Hence, there is a requirement to generate a hybrid representation that combines the advantages of occupancy grids and semantic environment representations. This chapter presents the SRTree, which is capable of generating a probabilistic occupancy representation for the task of navigation and exploration and additionally captures symbolic information about the environment. This representation can be useful in scenarios in which a robot is required to navigate in an environment while also allowing object-level reasoning.

As an instance of such reasoning tasks, the typical Human-Robot Interaction (HRI) scenarios encountered in the IURO project deal with higher-order spatial relationships. For instance, natural language route instructions frequently contain identifiers like ‘turn at the *next crossing*’, ‘the building *left of* the statue’ or ‘the crossing *behind* the traffic light’. Correctly interpreting these instructions requires the retrieval of the correct referent objects from the robot’s internal environment representation. Hence it is important for the robot to be aware of the environment semantics and the relationships between objects present in the environment.

The main contributions of this chapter are:

- The description of a novel inference method for Markov Logic Networks (MLNs) based on a pseudo-Boolean algebraic formulation

- A semantic occupancy grid (SRTree) that models the occupancy probabilities and assigns a class label to each grid cell

- The capability of dealing with higher-order spatial relationships using Markov Logic Networks

- A large scale colored point cloud dataset of downtown Munich annotated with 10 different class labels, which is made publicly available to the research community

The main advantage of the proposed approach is that it is capable of dealing with higher-order spatial relations and generates a hierarchical representation of large scale outdoor urban environments. The SRTree provides the foundation for this higher-order spatial reasoning through its hierarchic structure which ensures fast access to the occupancy grid, the environment semantics and allows storing of spatial relations within the same structure.

This chapter is organized as follows: Section 3.2 gives an introduction into probabilistic logic with MLNs and then proceeds to describe a novel inference mechanism that can be applied to them. The MLN formalism is then used to reason about qualitative spatial relations in the SRTree environment representation, as it is explained in Section 3.3. The dataset that is used for the experimentation is described in Section 3.4.1, while the experiments themselves are put forward in Section 3.4. Section 3.5 concludes the chapter with a discussion of the results and possible extensions of the method.

3.2 Efficient Inference in Markov Logic Networks using a Pseudo-Boolean Formulation

3.2.1 Motivation for Probabilistic Logic

As mentioned specifically for the field of spatial reasoning in Section 2.3.2, machine intelligence problems have often benefited from reasoning on a symbolic level. On the other hand, experience with systems operating in the real world shows that the stringency and hardness of the rules employed in deterministic symbolic reasoning is often unable to deal with the imprecision and uncertainty that is always contained in the data these systems operate with. For this reason, extensions of classic symbolic techniques using logical reasoning systems with probabilistic methods have proven successful in robotics and machine intelligence applications. In order to solve a particular task or equip a technical system with domain expertise, knowledge bases (KBs) are built which allow domain experts to describe important concepts and relationships. Observed data from a real-world system is then used to associate these rules with weights that reflect their validity in the noisy technical process. One particularly successful formalism combining logical and probabilistic reasoning are Markov Logic Networks (MLNs) [153]. They combine the powerful language of first-order logic with the flexibility of Probabilistic Graphical Models. The following section gives an overview over the basics of MLNs, and then proceeds to describe an algebraic formulation of these models that is based on the theory of pseudo-Boolean functions, which is shown to have computational benefits on a wide range of problems.

3.2.2 First-Order Logic and General Formulation of Markov Logic Networks

First-Order Logic

First-Order Logic is a language that describes relations between *objects* in the world. A full description of its aspects is out of scope for this thesis, and detailed treatments can be found for example in [41]. Objects in the world are part of a *domain*, which can be interpreted as a semantic type or categorization of the object. First-Order Logic refers to objects through *terms*, which can either be *constants* referring to a particular object, *logical variables*, which can stand for one element from a group of objects, or *logical functions*, which map tuples of terms to other objects. First-Order Logic *formulas* are formed from *predicates*, which describe relationships between terms, the connectives $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow, =$, and the existential and universal quantifiers $\exists$ and $\forall$. A *sentence* is a formula where all variables are bound, i.e. all variables are governed by a quantifier. A variable that is not bound by a quantifier is a *free variable*. Predicates that are applied on a specific tuple of terms, the *arguments* of the predicate, are called *atoms* or *literals*, while their negations are called *negative literals*. Atoms that only have constants as their arguments are called *ground atoms*.

In first-order logic inference, truth values $\top$ and $\bot$ are assigned to atoms based on partial knowledge about the objects in the world and the relationships present between them. First-Order Logic is not decidable, so it is not always possible to reach a definite conclusion about whether a satisfying assignment to the variables in a sentence exists or not.

Markov Networks

Markov Networks are an efficient way of specifying factored probability distributions. A distribution over a set of variables $\boldsymbol{x}$ that can be expressed as a product of factors

$$P(\boldsymbol{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \phi_C(\boldsymbol{x}_C),$$

where $\boldsymbol{x}_C$ are subsets of the complete set of variables, which are referred to as *cliques*, and the *partition function* Z is a normalization constant, belongs to the family of distributions that are expressible as Markov Networks. Markov Networks can be efficiently represented as an undirected graph which has a node for each variable, and nodes for variables co-occurring in a clique are connected with an edge. The nonnegative clique functions ϕ_C describe the influence of their arguments in the probability distribution and they are commonly chosen to be from the exponential family. Computations with and analysis of Markov Networks are often performed in the log domain, where the expression

$$-\sum_{C \in \mathcal{C}} \log \phi_C(\boldsymbol{x}_C) \tag{3.1}$$

is referred to as the energy function. Its minimizing arguments define the maximum-a-posteriori (MAP) variable assignment of the distribution. Apart from this, the formulation allows a host of different statistical computations. Aside from the MAP query, the marginal distributions for the random variables $\boldsymbol{x}$ are a common query in Markov Networks.

Markov Logic Networks

Markov Logic Networks use first-order logic functions as templates for the clique functions of a Markov Network. A MLN M_L, as defined by Richardson and Domingos [153], is a set of pairs (F_i, w_i), where F_i are formulas in first-order logic and w_i are associated weights. Together with knowledge about the domains of the variables contained in the KB and the objects contained in these, a *ground Markov Logic Network* can be defined, which is equivalent to a Markov Network with a special structure.

For the formal definition of MLNs that follows it is helpful to extend this concept to allow further restrictions on the set of objects a logical variable in a formula can be mapped to. These can be formalized as *constraints* C. For the discussion that follows, two types of constraints are useful: substitution constraints, which specify that logical variables $\boldsymbol{t}$ should only be mapped to the tuples of constants contained in a set P, and equality constraints, which specify that different logical variables should be mapped to the same constants (i.e. $(X = Y)$), or only to different constants (i.e. $(X \neq Y)$). The combination of a formula $F(\boldsymbol{t})$ in logical variables $\boldsymbol{t}$, its associated weight w and constraints for its logical variables C is called a *parfactor* [138] $g = (C, F, w)$.

The process through which the ground MLN is obtained is called *grounding*. It entails substituting each variable in each of the formulas in the database by all the objects in its domain, such that the formula only contains ground atoms for that particular substitution. This formula is then added as a clique function, which in the MLN context are called *factors*, to the network. The binary random variables of the network are thus the ground atoms resulting from the combination of formulas and domains in the knowledge base. To formalize this process, it is useful to introduce the substitution operator θ. It maps terms from a set $\mathcal{T}$ to terms from a different set $\mathcal{T}'$ according to a mapping $\mathcal{T} \to \mathcal{T}'$. The substitution of terms in formula f with the substitution θ is written as $f\theta$. A substitution where the set of terms that is being mapped to consists solely of constants is called a *ground substitution*. With these definitions, the set of all possible groundings of a set of logical variables $\boldsymbol{t}$ under the constraints C can be defined as $gr(\boldsymbol{t} : C)$. Now, the Markov Network resulting from grounding the logical variables in the knowledge base consisting of the parfactors $G = \{g_i\}$ can be written as

$$P(\boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_{g \in G} \sum_{\theta \in gr(\boldsymbol{t}_g : C_g)} F_g(\boldsymbol{t}_g)\theta \right). \tag{3.2}$$

Similar to the concept of conditioning in general statistical inference, it is possible to introduce knowledge about the state of some random variables into the model. In MLNs, this is done by specifying *evidence*. It consists of sets of tuples of objects for which the truth value assigned to a specific predicate, which has them as its arguments, is known. The sets of objects for which the truth value of predicate P is $\top$ or $\bot$ are denoted by $P_\top$ and $P_\bot$, respectively. Thus, if it is known that P is $\top$ when it is applied on the constants o, then $P(o) \in P_\top$. If evidence is available for all possible groundings of the predicate P, the predicate is *fully observed*. Otherwise, there are some groundings of the predicate for which no truth value is known *a priori*, which are contained in the set of unobserved groundings P_U.

There are different methods for obtaining the weights associated with the formulas. In engineered systems, they can be specified by a domain expert. If data that exhibits the relationships the MLN is supposed to model is available, then weights can be learnt to reflect the degree to which the given rules describe the data. The original formulation of MLNs [153] described a weight learning technique based on optimizing the pseudo-likelihood of labeled data under the model using gradient descent. Several other weight learning methods [78, 79, 114, 162] have been developed since.

Example 3.2.1. To illustrate the concepts introduced up to this point, a very small toy example for a MLN is created. A common example for MLN models describe the social aspects of smoking. It contains the formula $F = \mathit{Friends}(X,Y) \wedge \mathit{Smokes}(X) \rightarrow \mathit{Smokes}(Y)$, which means that friends have similar smoking habits. It contains the predicates *Smokes* and *Friends*, and the logical variables contained in the formula are X and Y, which are bound to a domain containing persons. The sensible restriction that this formula should not apply when X and Y both refer to the same person can be expressed by the constraint $C = (X \neq Y)$. If the domain for grounding the MLN contains the persons $\{A, B\}$, the groundings for the parfactor (F, w, C) are $\{(X,Y) \rightarrow (A,B), (X,Y) \rightarrow (B,A)\}$.

Inference in Markov Logic Networks

Multiple inference queries can be posed to the probability distribution (3.2) defined by the MLN. Most common inference goals are the marginal distributions of all or a subset of the random variables, and the variable assignment that maximizes the probability, conditioned on some evidence. This assignment is known as the maximum-a-posteriori (MAP) or most probable explanation (MPE) solution, and the discussion in this chapter is concerned with this instance of the inference problem. Due to the high treewidth of the ground models, this inference problem is in general intractable. The following gives an overview over recent methods for MAP inference in MLNs.

MaxWalkSat [83] is a stochastic search algorithm for weighted maximum satisfiability problems, of which the MAP query to an MLN is an instance. It can be implemented in a *lazy* way, which means that it is not necessary to keep the whole grounded network in memory during inference, as it is implemented in the Alchemy package [153]. The Tuffy system [132] extends this approach by using a relational database for the necessary grounding operations and by parallelizing inference in different weakly connected parts of the network, thus achieving higher scalability. A partitioning of the network in parts found using minimum cuts can also be used to parallelize the inference process, e.g.,in an importance sampling framework [15].

Contrary to these search- and sampling-based approaches, a different line of research has formulated the inference problem in MLNs as an integer linear program (ILP). In this framework, a relaxation of the integrality constraint results in a linear program, which provides a solution that can be converted to an approximate solution of the original problem by rounding. The optimal cost of the relaxed problem provides an optimistic estimate of the exact value of the problem. A critical factor for the complexity of the resulting inference problem is the number of constraints that are included in the problem. Riedel [154] presents a cutting plane algorithm for MLNs, which adds constraints to the problem in an iterative fashion if the current solution does not satisfy them, and disregards constraints that are satisfied. The RockIt system [133] improves scalability of the LP-based inference by identifying structurally similar constraints on the first-order level and parallelizes the problem into several smaller ILPs.

Yet a different group of approaches tries to leverage the first-order definition of the problem to avoid propositionalizing the full network. This class of algorithms performs so-called *lifted* inference [86]. Algorithms for performing lifted inference on MAP problems have seen significant improvements in recent years [7, 124, 129, 163]. Since, however, these algorithms are limited to perform on a limited set of classes of tractable problems with specific constraints on the types of rules and evidence, they are not discussed in depth in this work.

3.2.3 Pseudo-Boolean Formulation of Markov Logic Networks

Pseudo-Boolean functions

Pseudo-Boolean functions are functions that map Boolean variables $\boldsymbol{x} \in \mathbb{B}$ to real values. Let $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ be a vector of Boolean variables. From this, the set of corresponding literals $\boldsymbol{L} = \{x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}, x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\}$ can be defined, where the superscript (1) stands for a positive literal and the superscript (0) for a negative one. A general pseudo-Boolean function with M terms can be written as a weighted polynomial in literals

$$\phi(\boldsymbol{x}) = \sum_{i=0}^{M} a_i m_i(\boldsymbol{x}_i)$$

where a_i are real-valued coefficients and $m_i(\boldsymbol{x}_i)$ are *monomials* of Boolean variables

$$m_i(\boldsymbol{x}_i) = \prod_j x_{i,j}^{(\gamma_{i,j})}.$$

As before, the superscripts $\gamma_{i,j}$ indicate whether literal $x_{i,j}$ appears positive or negative in monomial m_i. The *order* or *degree* of a pseudo-Boolean function is the maximum number of literals in any single one of its monomials.[1]

With the *negation* substitutions for literals $x^{(0)} = 1 - x^{(1)}$ and $x^{(1)} = 1 - x^{(0)}$, it is possible to transform the polynomial representation of a pseudo-Boolean function ϕ into a different one with the same value table. Notable among them is the representation which contains only positive literals. It is called the *multinomial* representation.

Using these transformations, it is also easily possible to transform the pseudo-Boolean function into a representation where all coefficients a_i are positive by applying the negation substitution on a single variable in all terms $a_i m_i(\boldsymbol{x}_i)$ where $a_i < 0$. Thus, for each term with a negative coefficient, two weighted monomials with positive coefficients are created, one of the order of the original term and one with an order decreased by 1. Such a representation, which is not unique, is called a *posiform*, and has some special properties. Most importantly, the constant term a_0 in a posiform representation represents a lower bound for the value of the associated function.

A subclass of pseudo-Boolean functions, the class of so-called *submodular functions*, has received special interest in the context of optimization of pseudo-Boolean functions. Submodularity is usually defined for set functions, which map sets of elements $S \subseteq V$ to real values. A set function f that is equivalent to a pseudo-Boolean function g can be derived by defining S as the set of indices of variables taking the value 1 in a Boolean variable assignment $\boldsymbol{x}$. Then, the values of the set function can be determined as $f(S) := g(\boldsymbol{x})$. The definition of submodularity is then

$$f(X) + f(Y) \geq F(X \cup Y) + f(X \cap Y).$$

Submodular functions take a special role in the optimization of pseudo-Boolean functions since they can be minimized in polynomial time [22]. However, the recognition problem, i.e. the problem of deciding whether a given pseudo-Boolean function is submodular or not, is intractable for functions of degree of 4 or higher. For quadratic pseudo-Boolean functions, however, the recognition problem is trivial: A quadratic pseudo-Boolean function is submodular if all its quadratic terms have non-positive coefficients.

Conversion between Formulas in First-Order Logic and Pseudo-Boolean functions

In order to create a pseudo-Boolean representation of the energy function (3.1) of an MLN, it is necessary to convert the parfactors to pseudo-Boolean potential functions. This is possible for general functions by creating a value table and creating a monomial for each entry in the table, and then applying algebraic simplifications [22]. However, since the formulas of the MLN are first-order logic sentences, they can be converted to conjunctive normal form (CNF), which can directly be translated to pseudo-Boolean by applying

$$\bigvee_{u \in U} u = 1 - \bigwedge_{\bar{u} \in U} \bar{u}$$

[1] The term *order* has a different meaning in the contexts of pseudo-Boolean functions and first-order logic. Here, it is generally assumed that the context is clear enough to avoid confusion between the two.

to each clause U containing literals u. All conjunctions can then be replaced by multiplications to obtain a polynomial in the literals of the original logical function. All monomials coming from MLN formula F_i are then multiplied with the coefficient $\log(w_i)$.

The result of applying this procedure to all parfactors of a MLN results in a new probability distribution

$$P(x) = \frac{1}{Z} \exp \left(\sum_{g \in G} \sum_{\theta \in gr(L_g : C_g)} \phi_g(a_g)\theta \right),$$

where ϕ_g is the pseudo-Boolean representation of the first-order logic formula F_g weighted with w_g. The distribution is equivalent to (3.2), but the representation is presented here is purely arithmetic and can be manipulated as such. It is referred to as Pseudo-Boolean Markov Logic Network (PBMLN).

Example 3.2.2. This example describes the PBMLN created from the problem defined in the toy Example 3.2.1. The pseudo-Boolean representation of F is $\phi = 1 - s_y + f_{x,y} s_x s_y$, where s_x and s_y are binary variables for *Smokes*(X) and *Smokes*(Y), respectively, and f_{x_y} is the binary variable for *Friends*(X, Y). The ground PBMLN is given by

$$P(s_A, s_B, f_{A,B}, f_{B,A}) = \frac{1}{Z} w \exp\left(1 - s_B + f_{A,B} s_A s_B + 1 - s_A + f_{B,A} s_B s_A\right)$$

with the ground binary variables defined accordingly.

The formulation of MLN problems in pseudo-Boolean form allows an analysis of the tractability of MLN problems. Tractability in this context describes the property of computation time being polynomial in the number of variables, as opposed to general MLN inference problems, where the computational complexity is exponential in the number of variables. The known classes of tractable pseudo-Boolean functions can be carried over to the MLN domain to identify tractable instances of MLN inference problems [205].

Quadratic Pseudo-Boolean Optimization

Tractability analysis is not the only technique that can be transferred from the study of pseudo-Boolean functions to PBMLNs. It is also worthwhile to study the application of optimization methods from the pseudo-Boolean literature to PBMLNs problems. An interesting approach that has provided good results on image processing problems because of its computational efficiency is Quadratic Pseudo-Boolean Optimization (QPBO) [22, 89]. It computes an exact solution for second-order pseudo-Boolean functions which belong to the *submodular* class of functions. Functions which are not submodular or of an order higher than two can not be solved exactly in general. However, for non-submodular functions it is possible to find a submodular relaxation of this function. The minimal function value for this relaxation obtained with QPBO is a lower bound to the real minimal value. Furthermore, QPBO guarantees to find the optimal submodular relaxation of a function so that the lower bound it provides is the maximal one among all submodular relaxations. This bound is known as the *roof dual bound.* It can be efficiently computed using a *max-flow* computation on a network constructed from a posiform representation of the pseudo-Boolean function.

In the case of non-submodular functions, QPBO is unable to compute the minimizing assignment to all variables, since the full minimizer is unknown. However, some variables in the solution can be determined to be weakly or strongly *persistent*, which means that their value is known to hold in some or all minimizing variable assignments. Further analysis of the QPBO solution using the *probing* [21] and *improving* [89] procedures can lead to identifying a larger set of optimal variable assignments, and obtain a suboptimal variable assignment for the full set of variables, which can be useful as an approximate solution.

Order Reduction of Pseudo-Boolean functions

For higher-order pseudo-Boolean functions (of order higher than two), it is possible to compute an *order reduction*, which is a new quadratic function in the original set of variables with additional *slack* variables. This function ρ is guaranteed to take the same values as the original function ϕ in its original variables $\boldsymbol{x}$ when its value is minimized over the slack variables $\boldsymbol{w}$

$$\phi(\boldsymbol{x}) = \min_{\boldsymbol{w}} \rho(\boldsymbol{x}, \boldsymbol{w})$$

Example 3.2.3. The function $\phi(x_1, x_2, x_3) = -x_1x_2x_3$ has an order reduction

$$\min_{w} \rho(x_1, x_2, x_3, w) = \min_{w} -x_1w - x_2w - x_3w + 2w$$

with the single slack variable w. The correctness of this transformation can be shown by computing the minimal value of ρ for all assignment to (x_1, x_2, x_3) and verifying that it takes the value -1 for $(1, 1, 1)$ and is equal to 0 for all other assignments. Now, the minimal value of ϕ can be computed through minimizing the quadratic function ρ in the augmented set of variables.

An order reduction satisfying this property exists for all functions [157]. Different methods for computing them have been proposed in the literature.

Ishikawa [81] presents a general technique ISH for quadratizing pseudo-Boolean functions, which works on each higher order term separately. It distinguishes between terms with positive and negative coefficients, where terms with negative coefficients can be quadratized with a single slack variable and terms with positive coefficients of order n result in a minimization over quadratic terms with approximately $\frac{n}{2}$ slack variables. An interesting approach that may be used to reduce the number of non-submodular terms in the quadratic function by leveraging the interaction between different terms is presented by Gallagher, Batra, and Parikh [58]. Their *asymmetric quadratization* ASM for third order terms with coefficient $a > 0$ is defined by

$$ax_1x_2x_3 = \min_w a(w - x_3w - x_3w + x_1w + x_2x_3).$$

It can be seen that the right-hand side of the equation is not symmetric in the original variables, but the left-hand side is. Thus, three different quadratizations can be defined depending on the choice of the order of the variables. Each variant contains a positive quadratic term which does not contain the slack variable w. If the variables of this term are present in other terms of the function, the corresponding terms can be combined, and in some cases the non-submodular term can be avoided. This prompts the question of which quadratization variant to choose for which higher order term in the original function. Gallagher, Batra, and Parikh [58] approach this question by posing a separate optimization problem for this choice to obtain a quadratic representation of the problem that is amenable to efficient minimization using QPBO.

Another order reduction method FIX that takes advantage of interactions between reductions of different higher order terms was introduced by Fix et al. [51]. It identifies higher order terms that have a common subset of variables, and transforms them to a sum of negative higher order terms and possibly positive quadratic terms.

Finally, *generalized roof duality theory*, which defines a different way of obtaining a higher order submodular relaxation of a pseudo-Boolean function, can also be used to obtain a quadratization GRD of a pseudo-Boolean function [205].

Quadratize-Solve-Simplify-Repeat (QSSR)

As it has been noted above, the QPBO algorithm generally does not provide a full solution for a pseudo-Boolean minimization problem. However, with the notion of persistent assignments, a partial assignment of values to some of the variables in the problem is often found. This can be used to iteratively decrease the size of the higher order minimization problem by first obtaining a quadratization of the problem, obtaining a persistent solution and setting those variables for which an assignment was found to their optimal value in the original problem. Optionally, the probing algorithm can be executed on the quadratized problem as well in order to obtain a larger set of persistent values. Then, the process can be repeated on the new, smaller higher order problem. This procedure can be iterated until no new persistencies are found. We denote this succession of *quadratizing – solving – simplifying – repeating* as QSSR algorithm. A similar of iteratively solving a relaxed problem and fixing the variables in the original problem is also applied by [82] in the context of computing the generalized roof duality bound.

Preprocessing of PBMLNS Based on Logical Structure

Further preprocessing steps are possible in the PBMLN formulation to make the subsequent inference steps more efficient. For one, information about evidence to be introduced can be used to reduce the size of the problem, and to inform the order reduction steps. Consider the example parfactor

$$g = (\emptyset, P(X)P(Y)S(X,Y) + P(Z)S(X,Y).$$

For the arguments of ground atoms of the predicate S, some values are known, i.e. $S_T \neq \emptyset$. These groundings can be subsumed in a new parfactor

$$g' = ((X,Y) \in S_T, P(X)P(Y) + P(Z).$$

As the example shows, the predicates with groundings from evidence do not show up in the simplified parfactor. If the simplified parfactor function evaluates to 0 given the known value of the evidence atom, the parfactor can be omitted from the simplified model completely, thus reducing the number of parfactors that need to be grounded [167]. Additionally, the function becomes simpler when the variables X and Y are mapped to the same constant under a substitution θ, i.e. $X\theta = Y\theta$. In this case, the original parfactor becomes

$$g'' = ((X = Y), P(X)S(X,X) + P(Z)S(X,X).$$

Finally, for substitutions where $X\theta = Z\theta$, the parfactor can be simplified to

$$g''' = ((X = Z), (((P(Y) + 1)P(X)S(X,Y)))).$$

These simplifications are important not only because they decrease the total number of grounding substitutions that have to be considered, but also because they may reduce the order of the pseudo-Boolean function defining the parfactor. In this case, as for the first part of the example given above, it may be possible to apply a simpler order reduction, or, if the simplified function is quadratic, no order reduction is necessary at all for the respective ground formulas. Thus, the number of slack variables necessary in the full quadratized optimization problem is reduced.

Another aspect of grounding a PBMLN which helps to reduce the number of slack variables is the fact that groundings of different parfactors may result in the same ground atoms [138]. It can be beneficial to split up the parfactors such that the terms of ground atoms resulting from each grounding each parfactor are strictly disjoint. For example, consider a PBMLN containing two parfactors $g_1 = (\emptyset, w_1P(X)Q(X))$ and $g_2 = (\emptyset, w_2P(X)Q(Y))$. Grounding g_2 with a substitution θ where $X\theta = Y\theta$ results in a multinomial that is also contained in the groundings of g_1. In order to determine the unique terms contained in the ground network already before grounding, which is more efficient than combining the terms on the ground level, the parfactors can be split up. For this example, this results in an equivalent model with the two parfactors $g_1' = (\emptyset, (w_1{+}w_2)P(X)Q(X))$ and $g_2 = ((X \neq Y), w_2P(X)Q(Y))$ with unique groundings.

3.2.4 Results on Standard Examples

The effect of different quadratization methods on inference performance was evaluated for different standard problems known from literature. This section first presents a comparison of the performance of the MaxWalkSat inference algorithm, depending on the preprocessing of the problem, and then proceeds to compare inference using QPBO on PBMLNs with various other inference engines.

Inference on PBMLN using the MaxWalkSAT algorithm

Social Network Model The social network model described in Richardson and Domingos [153] is a common testbed for MLN algorithms. The model captures some of the patterns related to friendship relationships, smoking habits and the probability of developing cancer. The version of the model that we use is characterized in Table 3.1. Evidence is generated in a similar way as described by Singla and Domingos [171], where a fixed percentage of humans have 10 known friends and known smoking habits. The task is then to infer the existence of other friendships relationships as well as the smoking habits and the cancer occurrences in the total population.

Table 3.1: The employed social network model with weights as specified in Singla and Domingos [171]. The last formula is included in the original model [153]

Nr.	Formula	w
1	$\neg Friends(x,y)$	4.6
2	$\neg Smokes(x)$	1.4
3	$\neg Cancer(x)$	2.3
4	$Smokes(x) \Rightarrow Cancer(x)$	1.5
5	$Smokes(x) \wedge Friends(x,y) \Rightarrow Smokes(y)$	1.1
6	$Friends(x,y) \Rightarrow Friends(y,x)$	4

In this case study, we focus on the social network model and explore the influence of different representations of the logical formula on the performance of the MaxWalkSat algorithm.

While the PBMLN formulation specifies the factors of the grounded MLN as monomials, the original formulation of the MaxWalkSat algorithm requires MLN formulas to be represented in clausal form. Thus, we use a different formulation of the algorithm, which operates on the network of ground monomials directly. For the modified MaxWalkSat algorithm, the states of variables in unsatisfied monomials, as opposed to unsatisfied clauses, are candidates for being flipped. The change in state to be taken is decided by the maximum reduction in cost that can be effected by flipping a candidate variable. As in the canonical form of the algorithm, these greedy steps alternate with random flips of variables. The cost function optimized in this procedure is (3.2), because the ground PBMLN is logically equivalent to the ground MLN.

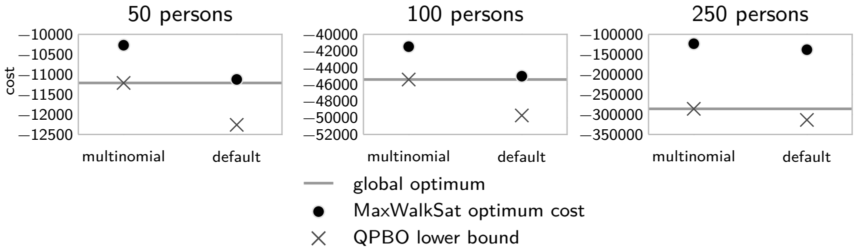


Figure 3.2: Minimum costs resulting from running MaxWalkSat on two different representations of the social network model for at most 20000 iterations and QPBO lower bounds computed for different reparameterized versions of the social network model.

In order to get an understanding of the effect of different pseudo-Boolean representations on the performance of MaxWalkSat, two different instances of the social network model are solved. In one, the logical formulas are converted to pseudo-Boolean form following the steps described in Section 3.2.3 and then used directly (*default*). For the other model (*multinomial*), the pseudo-Boolean formulas are subsequently converted to multinomial representation, i.e.no negated variable are contained in the final representation. For each instance of the problem, MaxWalkSat is run for 20000 iterations, and the best result of 50 randomly initialized Markov chains is used. Experiments were run on machines with 4 CPUs running at 3.3 GHz and 16 GB RAM. Multiple MaxWalkSat Markov chains were run in parallel. We use the QPBO implementation of Kolmogorov and Rother [89][2] to compute lower bounds to the optimal costs. The resulting optimal costs over the runtime of the algorithm are given in Figure 3.2, along with the QPBO lower bounds for each of the representations.

In addition to applying the inference algorithms to different pseudo-Boolean representations of the problem, we also compare the performance of MaxWalkSat on factor graphs created with the PBMLN approach with factor graphs that follow the approach put forward for MLNs specified purely on the first-order level. For the latter, the ground factor graphs consist of a factor for each conjunction of ground clauses of the CNF of the formulas in the KB. In the original MLN implementation [153], this construction is further relaxed by taking the sum of the truth values of multiple clauses in each formula instead of their conjunction in case the CNF of the formula consists of multiple clauses. We compare the development of the cost over iterations of the MaxWalkSat algorithm for both formulations (denoted by *Monomials* and *Clauses*, respectively) for instances of the social network model of varying size in Figure 3.3.

From these results, it can be seen that the representation of the problem influences the performance of the MaxWalkSat inference algorithm. In Figure 3.2, it is apparent that the local search converges for all problems, but the convergence speed can be influenced by the choice of pseudo-Boolean representation of the problem. Rewarding the compactness

[2] available at `http://pub.ist.ac.at/~vnk/software.html`

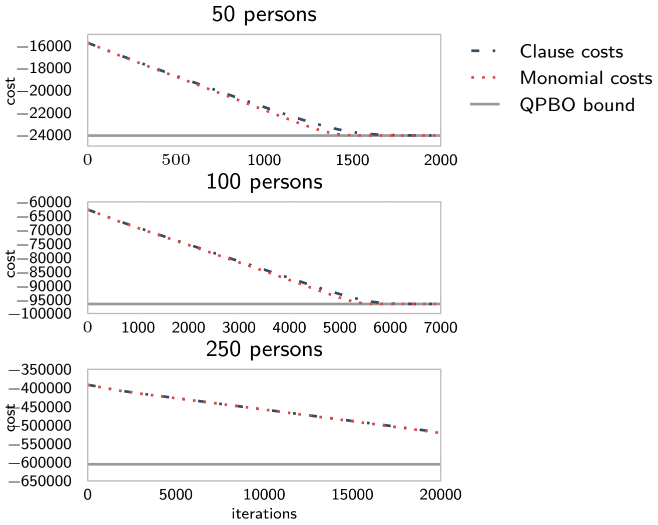


Figure 3.3: Trajectory of costs resulting from running the MaxWalkSat algorithm in clausal form and in the compact representation of the monomial form on different network sizes of the social network model

of the problem is a good choice of representation for MaxWalkSat, since a compact factor graph has fewer terms in the computation of the cost of the current state. This creates a performance benefit, as it can be seen by comparing the minimum cost attained after a fixed number of iterations with the *multinomial* representation against the canonical formulation, which produces a less sparse ground network. Figure 3.3 shows that the PBMLN formulation is also advantageous in comparison with doing inference on a factor graph built directly from the CNF representation of the formulas in the KB. While both methods approximately reach the global minimum for smaller instances of the problems, convergence is faster for the problem formulated in monomials optimized for compactness. For the larger problem with 250 humans, MaxWalkSat does not get close to the optimum within the allotted number of iterations in either of the problem formulations.

Inference on PBMLN using QPBO

The presented PBMLN inference approach using different quadratization methods are evaluated by their impact on the performance of the QPBO algorithm and its extensions. The performance of the QPBO-based inference is also evaluated on problems for which no quadratization is required. Finally we compare our overall pipeline to existing inference engines.

Datasets We evaluate our approach on various standard MLNs and datasets as well as additional problems. The characteristics of these problems are summarized in Table 3.2. A first set of datasets is similar to the ones employed in the evaluation of state-of-the-art engines Tuffy [132] and RockIt [133]. The link prediction problem on the UWCSE dataset (LP) tries to find relations between faculty members and students. The relational classification (RC) on the Cora dataset determines the category of research papers. The information extraction (IE) problem models how to obtain dataset records from parsed sources. The webKB dataset is used to predict to which university department a website belongs, given its hyperlink relations and contained words (KB). The entity resolution (ER) problem on the Cora dataset is obtained from the Alchemy website. The goal of this problem is to identify citations referring to the same paper. Because no trained model is available for this problem, it is trained with Alchemy using the first of the five available splits for evaluation [172]. The Friends and smokers social network (F&S), as described above, is a common test model for a social network with friendship relations, smoking habits and cancer occurrences. Evidence is generated as described by Singla and Domingos [171] for a domain size of 200 persons. Because the F&S problem is relatively simple, an additional problem in which the weights of all formulas are negated is also considered (-F&S).

In order to gain a broader insight into the performance of the inference algorithms on higher order problems, we created two additional third-order problems. The first one is based on the KB problem and the webKB dataset mentioned above (KB3). While the original KB inference problem uses words contained in the page contents as well as the link structure to infer page categories, a third-order problem on the webKB dataset is created by not only querying for the class of each page, but by also jointly inferring the links of a page, solely from the word tokens appearing on the page. Learning was performed with Alchemy, and the size of the problem was reduced such that inference is only performed over atoms that are $\top$ in the ground truth and the same number of randomly sampled atoms.

The second new third order problem is the image denoising (ID) model, which tries to restore a noisy binary image. There are rules indicating that the observed value of a pixel should correspond to the denoised value and two rules indicating that groups of three horizontally or vertically neighbouring pixels should take the same value. It can be shown that the associated MAP problem for these rules fall within the described cases of MLNs whose rules can be converted to tractable pseudo-Boolean functions described in Section 3.2.3, and can thus be solved exactly. The unary rules are given weight 1.0. To ensure that the terms of the smoothing rules do not cancel out, a rule for the ‘on’ pixels is given a weight of 0.35 and the rule for the ‘off’ pixels 0.3. A 90 × 90 pixels random binary image is used as evidence, where each pixel has a 50% chance of being on or off.

Other Engines We compare our approach with the MAP-inference solvers Alchemy, Tuffy and RockIt. Alchemy is the original solver for MLNs and, in contrast to the other engines, it does not use a relational database to ground the model, which can lead to long grounding times. Alchemy and Tuffy optimize the ground model using MaxWalkSAT, a stochastic search technique that can be made to scale well with large problems. RockIt uses an ILP solver and exploits symmetries in the model to reduce the number of constraints. Because the number of constraints may be very large, it takes an iterative approach where only the constraints that are violated for the current solution are added to the solver.

To make sure that the problem to be solved is the same for all implementations, some preprocessing is required. First, formulas with existential clauses are ignored and all formulas are converted to conjunctive normal form. Then, because Tuffy internally transforms formulas with a negative weight to an approximate formula with a positive weight, we apply the same transformation. Unfortunately, this transformation can not be applied for the higher order problems, as it reduces the order of the formula. Lastly, the ER and KB3 problems use a method to compactly specify which ground atoms to query, and assumes that all other query atoms are $\perp$. These query variables, also known as canopies [172], can be used to eliminate a large number of uninteresting variables, and can be created using a cheap distance metric [120]. Because neither Tuffy nor RockIt support this input format, they are given the extensive list of $\perp$evidence atoms instead.

	IE	**KB**	**RC**	**LP**	**ID**	**F&S**	**-F&S**	**KB3**	**ER**
Formulas	1024	106	15	24	4	6	6	66	1331
Domains	4	3	3	8	1	1	1	3	5
Query Predicates	2	1	1	1	1	3	3	2	4
Observed Predicates	16	2	3	21	2	0	0	1	6
Ground atoms	336670	9079	9650	4624	8100	40180	40180	8190	10948
Factors	351001	31283	58485	161806	55800	127982	127982	22627	910670
Higher order factors	0	0	0	0	15840	32220	32220	6736	424580

Table 3.2: Summary of the characteristics of the described datasets and the associated ground networks when grounded in their higher order form and a multi-linear representation. Trivially satisfied or dissatisfied factors are ignored.

Results on Quadratic Problems For quadratic problems from literature, we analyze the performance of QPBO and the additional persistencies computed with the probing extension described in Section 3.2.3. Table 3.3 shows that the QPBO algorithm gives a persistent solution for most variables, and even provides an exact solution for the KB problem. The *probe* procedure also solves the IE problem exactly, but still leaves some unsolved variables for the RC and LP problems. In general, the inference times for these problems are extremely short.

Comparison of Quadratization Methods For the higher order problems, the performance of each of the quadratization methods described in Section 3.2.3 is evaluated. This includes the pairwise MLN approach [50], which is equivalent to the ISH quadratization for problems with cubic potentials. The potentials of the parfactors are expressed as a multi-linear polynomial before quadratizing the model.

	IE	KB	RC	LP
Persistencies	99.87	100	90.30	85.58
Persistencies (probe)	100		90.30	86.22
Qpbo time (s)	0.030	0.002	0.006	0.069
Probe time (s)	0.150		0.021	4.800

Table 3.3: Percentage of persistencies given by the QPBO algorithm and after using the probing technique for different quadratic problems.

First we evaluate the number of persistencies that can be obtained for the different problems in Table 3.4. As expected, because of submodularity, the ID problem is completely solved by all methods. Friends and Smokers creates few non-submodular terms, and can also be solved exactly by all methods. The remaining problems can not be solved by all methods, for which in some cases only a small number of variables can be fixed. In general, it can be observed that the methods that are aware of the other terms in the potential produce better results than ISH, which applies a fixed transformation. The final probing step is computationally the most expensive, but may significantly increase the number of solved variables, and even solve some problems exactly. It should be noted that this step is important even when an approximate solution for all variables is subsequently obtained using the improve method. Otherwise, if the number of persistencies is small, the improve method needs to operate on a model with potentially many more variables, as it also needs to optimize the slack variables stemming from the remaining higher order terms.

Approximate Inference We also compared quality of the approximate solutions with those of other engines and their total running times. The problems were formulated as minimizations, and the solutions of all engines evaluated on the same ground model. In Table 3.5 it can be observed that for the quadratic problems, most engines achieve optimal costs, which are known from the optimality guarantee given by QPBO in Table 3.3 and from the small MIP gap that was used for RockIt. An exception is the LP problem, where the solver used by RockIt has problems obtaining a tight bound, and Tuffy and QPBO+I provide better solutions.

For the higher order problems, the ASM quadratization achieves the best cost and the lowest computation time for most cases. Using the GRD reduction performs slightly worse, possibly because for this quadratization the improve step needs to be executed over more variables. Tuffy does not perform very well in the higher order problems, possibly because the internal transformation it uses is an approximation of the original formula.

It should be noted that computation times are affected by multiple factors. Whereas Alchemy, Tuffy and our approach make a clear distinction between grounding and inference, RockIt uses a cutting plane algorithm that incrementally grounds factors that are not satisfied by the current solution, which leads to large speedups when many

	Step	ISH	FIX	ASM	GRD
ID	1	100.0 (0.01)	100.0 (0.01)	100.0 (0.01)	100.0 (0.01)
F&S	1	100.0 (0.02)	100.0 (0.82)	100.0 (0.86)	99.6 (1.0)
	2				100.0 (0.0)
-F&S	1	19.4 (1.79)	19.4 (0.9)	99.6 (0.42)	99.6 (1.11)
	2	19.4 (1.61)	19.4 (0.78)	99.6 (0.02)	99.6 (0.02)
	3	†	†	99.6 (2.66)	99.6 (2.54)
KB3	1	55.0 (0.06)	82.4 (0.03)	82.3 (0.02)	60.6 (0.08)
	2	56.5 (0.04)	86.8 (0.01)	86.6 (0.01)	62.7 (0.04)
	3	65.8 (19.42)	100.0 (0.1)	100.0 (0.05)	96.7 (5.59)
ER	1	92.1 (0.46)	91.9 (1.04)	95.0 (0.47)	95.4 (1.49)
	2	92.3 (0.03)	92.3 (0.04)	95.0 (0.02)	96.1 (0.03)
	3	92.8 (7.6)	93.0 (162.36)	95.3 (5.57)	96.6 (7.08)

Table 3.4: Percentage of variables solved. Step 1) Initial QPBO result 2) QPBO result after QSSR simplification 3) Probe. Inference time in seconds for each step in parentheses. (†) Did not complete

factors are easily satisfied or if the solution is largely homogeneous. On the other hand, the ID problem is an example where this approach produces considerably longer running times. Another influence on computation times is the ability to specify the evidence in the form of canopies, which allows the relational database to execute the queries for grounding more efficiently.

In Figure 3.4, the evolution of the cost of the ER problem against the running time of *improve* is shown for different quadratizations. For this problem, the methods converge to a solution with similar costs, but convergence is much faster in the cases where ISH and GRD quadratizations are used.

3.2.5 Discussion of PBMLNs

This section has presented a novel representation of a general class of probabilistic logic problems, Markov Logic Networks, in terms of pseudo-Boolean algebra. As a purely algebraic representation, this representation allows more flexible ways of analysing and manipulating the problem than the standard logical formulation. It also allows the application of powerful inference algorithms for general Markov Random Fields from the computer vision literature. In particular, the problem can be converted to an Markov Random Field (MRF) with exclusively pairwise interactions, and the QPBO algorithm can be applied on the resulting network. This procedure has been shown to produce results of state-of-the-art quality and computation demands. In addition, the effect of the quadratization method on the quality of the result has been shown.

	Alchemy	**Tuffy**	**RockIt**	**QPBO+I**	
IE	†	**-4511.6** (17)	**-4511.6** (19)	**-4511.6** (22)	
KB	-111113.5(162)	-111274.1 (115)	**-111312.4*** (27)	**-111312.4*** (6)	
RC	†	-4031.7 (17)	**-4031.8** (11)	**-4031.8** (9)	
LP	-480.8 (119)	-686.3 (424)	-507.7 (13)	**-732.6** (9)	
				ASM+QPBO+I	**GRD+QPBO+I**
ID	1772.7 (442)	1784.2 (25)	**-1003.8*** (244)	**-1003.8*** (5)	**-1003.8*** (6)
F&S	-3.8 (159)	**-4.2*** (3)	**-4.2*** (5)	**-4.2*** (6)	**-4.2*** (9)
-F&S	-182338.7 (47)	-191856.9(3230)	-185267.3 (8)	**-193715.3** (12)	**-193715.3** (14)
KB3	21.1 (543)	-1045.3 (308)	**-1492.8** (256)	-1484.4 (57)	-1476.9 (101)
ER	-10739.5 (551)	-14128.9 (433)	-15271.3 (1902)	**-15430.7** (101)	-15430.5 (113)

Table 3.5: Resulting cost for different engines on various quadratic and higher order problems. Alchemy and Tuffy were run for an increasing number of flips until no significant advances were made. RockIt was run with relative gaps 1×10^{-n}, $n = 9, 8, \ldots$ until convergence is achieved within an hour. These are compared against our method using the ASM and GRD quadratization for the higher order problems, using *improve* on the residual problem until no advances were made for 20 iterations. Total running times in seconds in parenthesis. (*) Guaranteed optimal cost by persistencies (†) Did not ground within 1 hour.

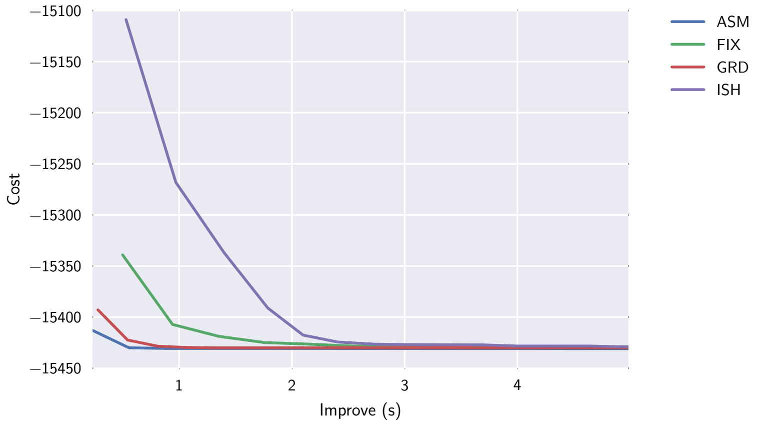


Figure 3.4: Cost in the higher order ER model as a function of the time spent on the *improve* method, using different quadratization techniques. Improve starts after solving one iteration of the original problem and removing the redundant slacks, as described in Section 3.2.3.

3.3 Estimation of Spatial Relations from Labelled Point Clouds

In this chapter, an approach to efficiently perform semantic reasoning about spatial relations between typical objects in a urban environment is presented. This information can be deployed for example in a HRI scenario, where a human identifies an object by its position with respect to the environment; the robot needs to retrieve the most likely candidates for the object being referred to from its internal environment representation. This information can then be retrieved and put to use for example for semantic navigation or in an HRI scenario. An example for a task where these abilities are required is the reasoning on route descriptions given by humans [208].

The overall pipeline of operations performed by the approach proposed in this chapter is shown in Figure 3.5. The input of the processing is a multi-attribute point cloud which defines the geometry, color, class label and object assignment for each point in the sensor field of view. The geometry and color information of each point in the point cloud is obtained from the Z+F 5010C laser scanner (see Figure 3.9) which fuses data of an RGB camera and a laser range finder. The proposed approach requires a point cloud with pre-segmented objects and class labels assigned to each point in the point cloud. The set of class labels is chosen based on categories which are commonly found in urban environments, such as sidewalks, trees, buildings etc. In addition to the class labels, an object id is assigned to be able to define relations between objects and to disambiguate between multiple objects of the same class (e.g., multiple trees or cars etc.). The multi-attribute point cloud is inserted into the SRTree which evaluates for each grid cell the occupancy probability, the most probable class label and object id attributes. This hybrid representation is then used to extract different geometric features which are employed in a Markov Logic Network framework to generate consistent spatial relations and determine higher order spatial relations between objects present in the urban environment. The pipeline becomes a closed chain once the determined spatial relations are inserted into the SRTree in form of a *spatial relation graph* between objects and used in typical HRI scenarios which require reasoning over route or scene descriptions.

3.3.1 Semantic Rtree (SRTree)

The first component of the pipeline is the SRTree, which is used to generate a metric and semantic representation of the environment. The SRTree is based on the standard Rtree [68, 128] and is an extension of the RMAP mapping approach [84], which presents a 3D occupancy grid where the cells are organized in an Rtree data structure. The Rtree is composed of a hierarchy of axis-aligned rectangular cuboids and contains a root node and a hierarchy of inner and leaf nodes. Root and inner nodes can have a maximum number M of children. The leaf nodes are grid cells in a fixed-resolution 3D grid, but they are only represented in the tree for volumes that have a nonnegligible occupancy probability, such that free space does not need to be modeled explicitly. The inner nodes define a minimum bounding rectangular cuboid over their child branches.

In addition to the occupancy probabilities, the proposed SRTree assigns a class label to each grid cell based on the labels of the points inserted in it. Each grid cell maintains a histogram of the observed counts of points with each of a fixed set of class labels. Due to the noise resulting from the discretization effects of the occupancy grid or imperfect

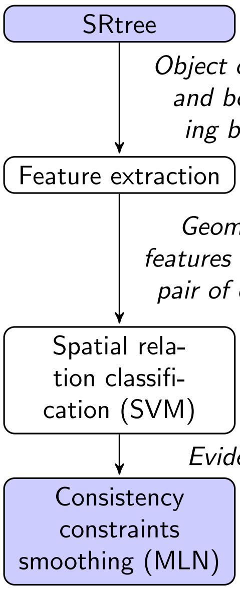

SRtree
Object cuboids and bounding boxes
Feature extraction
Geometric features for each pair of objects
Spatial relation classification (SVM)
Evidence
Consistency constraints smoothing (MLN)

Figure 3.5: The pipeline of operations performed in the proposed approach. The two components that are the focus of this chapter appear shaded.

class label assignment, it is necessary for each grid cell to take the uncertainty of the observed counts into account. To model this uncertainty, a Dirichlet distribution is used which generates a distribution over multinomial distributions for each object category.

Consider the probability density function of the Dirichlet distribution,

$$p^j(x_{i,1}, \ldots, x_{i,N-1}; \alpha_1^j, \ldots, \alpha_N^j) = \frac{1}{N(\alpha^j)} \prod_{k=1}^{N} x_{i,k}^{\alpha_k^j - 1},$$

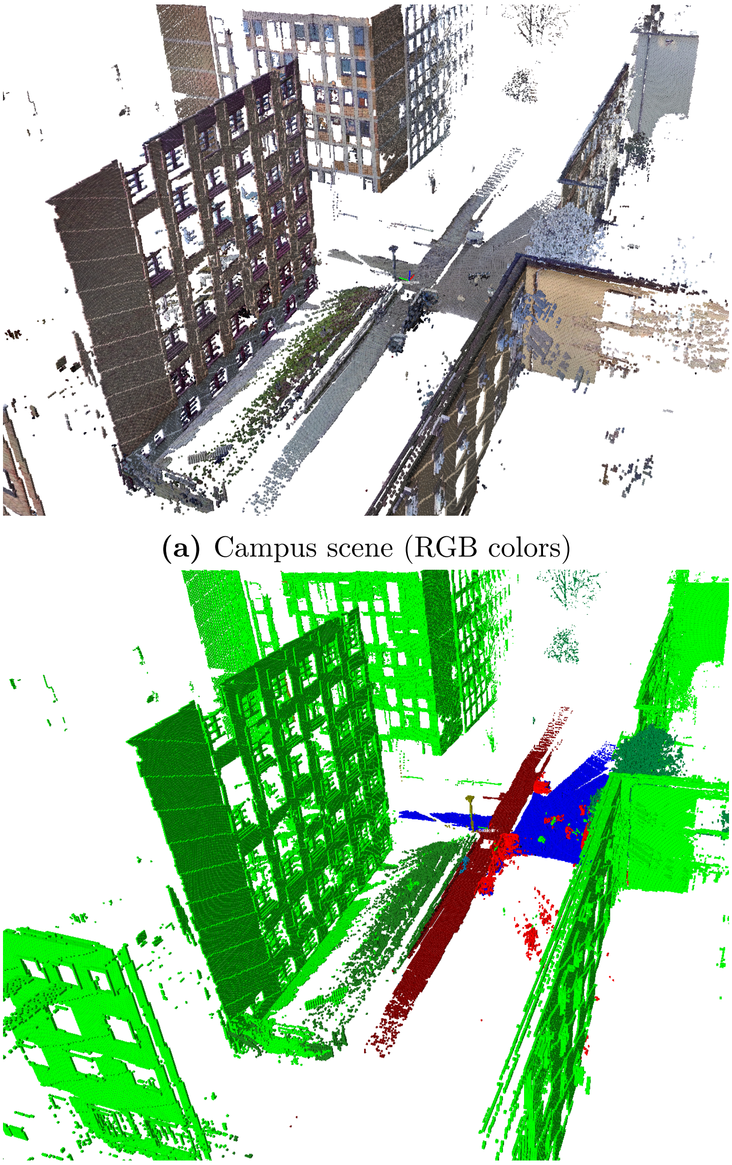
where α_k^j represents the k^{th} concentration parameter of the Dirichlet distribution that corresponds to the j^{th} object category, where $k, j \leq N$. The concentration parameters $\forall k$, α_k^j are learnt offline using the moment matching method [156]. $x_{i,1}, \ldots, x_{i,N}$ represent the class occurrence probabilities ($\sum_{k=1}^{N} x_{i,k} = 1$) of the i^{th} grid cell, which are calculated based on the normalized histogram of observed class counts. $N(\alpha)$ represents the normalization factor which can be expressed using the gamma function (γ) as follows

$$N(\alpha^j) = \frac{\prod_{k=1}^{N} \gamma(\alpha_k^j)}{\gamma(\sum_{k=1}^{N} \alpha_k^j)}.$$

The assignment of a specific label l_i to a grid cell g_i is based on

$$l_i = \underset{j}{\operatorname{argmax}}\, p^j(x_{i,1}, \ldots, x_{i,N-1}; \alpha_1^j, \ldots, \alpha_N^j), \forall j \leq N.$$

Hence the SRTree is capable of generating a probabilistic occupancy grid and additionally defines a class label for each grid cell. Figure 3.6a shows the SRTree occupancy grid



(a) Campus scene (RGB colors)

(b) Campus scene (class labels)

Figure 3.6: The SRTree with grid cells colored based on RGB values and class labels (green: building, dark red: sidewalk, blue: street, red: car)

with the grid cells colored based on the average RGB values of all points that falls within that cell. Figure 3.6b shows the occupancy grid cells in different colors based on the class label assignment using the Dirichlet distribution. The relation graph between objects is obtained after inference on the Markov Logic Network as described in the following section.

3.3.2 Reasoning over Spatial Relations

This section describes how the metric environment information from the SRTree is augmented by estimates of qualitative spatial relationships between objects. This process starts by extracting numeric features from the metric representation of the environment presented by the SRTree, which are then classified to a known set of relations. These estimates are subsequently postprocessed for a degree of global consistency with the application of rules in probabilistic logic, in the form of a Markov Logic Network as described in Section 3.2.2. The MAP assignment of spatial relations between objects is then inferred. The MLN approach allows for an accessible specification of spatial relations, in particular ones of higher order (> 2), such as the transitivity of the *On* relationship: $\mathit{On}(\mathit{base}, \mathit{middle}) \wedge \mathit{On}(\mathit{middle}, \mathit{upper}) \rightarrow \mathit{On}(\mathit{base}, \mathit{upper})$. Similarly, the qualitative location of an object to another, with respect to an observer, is a higher-order function of the relations between the two objects and their relation to the observer [125].

The graph of objects and their relations built this way can then, in turn, be stored back into the SRTree representation. This section firstly details the model for spatial relations that was chosen with the application of reasoning over route directions in mind, and introduces related approaches. Then, the individual steps of the process – feature computation, baseline classification and smoothing with a Markov Logic Network – are detailed.

We are concerned with relationships such as *left/right/behind/in front of* and relations of support between objects that can be used to locate objects in a scene. Thus, we restrict the set of qualitative spatial relations that are reasoned over in this work to the relations *On, LeftOf/RightOf* and *Behind/InFrontOf.* This representation system for the latter two pairs of mutually exclusive relations is illustrated in Figure 3.7, and examples for objects in an urban environment for which these relations apply are given in Figure 3.8.

This selection of relations is motivated by the typical terms used in route directions, the understanding of which is an important problem in collaborative robotic applications. The set of relations is inspired by more complex models used in qualitative spatial reasoning, such as the Region Connection Calculus [37], and in particular the Single Cross Calculus [55], which are described in more detail in Section 2.3.1. In the application context of processing route descriptions, the route graph [194] employs categories similar to the ones used here. The complexity of reasoning and the requirements to perception and environment understanding can make the application of these structures challenging in real-world robotics situations. For this reason, in this work we relax the constraints of exact logical consistency of the relations between objects. This enables the use of a

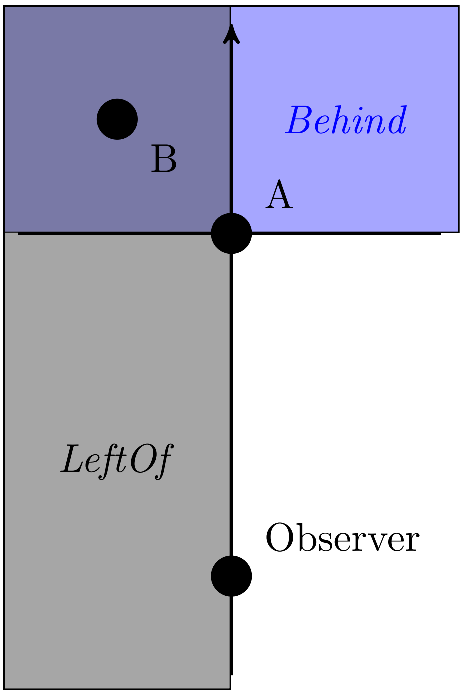


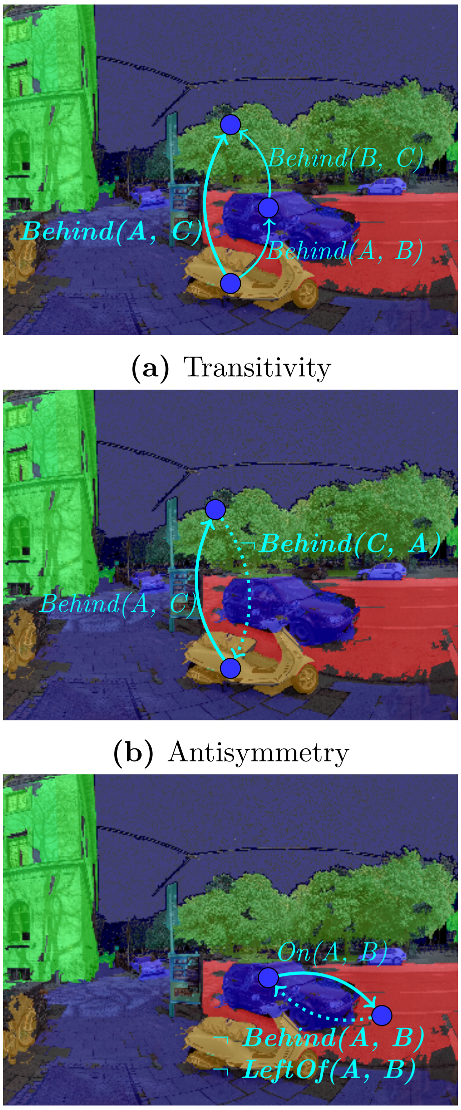
Figure 3.7: A subset of the qualitative spatial representation

probabilistic model with an approximate inference procedure.

An approach related to the one presented here, which is also based on the specification of relations between objects in a logical language, has been proposed for a different set of relations by Sjöö, Pronobis, and Jensfelt [173]. A method to estimate support relationships between objects in indoor scenes is discussed by Silberman et al. [169].

The approach presented here uses a basic set of features about the geometric relations of the objects in a scene as a first step. For every object in the scene, a bounding box is computed. The orientation of the bounding box is chosen such that it matches the mode of the histogram of the normal vectors computed for each cuboid, such that it is aligned with the principal directions of rectangular objects. The bounding boxes are shrunk along each of their axes to prune away a small percentile of points in order to obtain a tight fit. This step of using bounding boxes for feature computation assumes a scene that roughly adheres to a Manhattan model of the surroundings, which is a reasonable assumption for many objects and topologies encountered in an urban environment (e.g., houses and cars). A visualization of a scene with the point cloud as stored in the SRTree, the normal vectors and the bounding boxes, is shown in Figure 3.10.

The features are computed for each pair of objects in a coordinate system that is aligned with the simulated viewpoint of a person describing the spatial relations present in the scene. The feature set includes the distance between object centroids, as well as the maximum and minimum distances between the bounding boxes. These distances are also computed for projections of the bounding boxes along the axes of the coordinate



(a) Transitivity

(b) Antisymmetry

(c) Exclusivity of *On*

Figure 3.8: Illustration of the used spatial relations

system. For example, the minimum vertical distance between the bounding boxes of two objects is a feature. The percentage of points of an object that overlap with the convex hull of a second object in a projection of both objects to each of the coordinate axes is another feature. Further features are computed based on a cylindrical coordinate system. Namely, the angles between the central axis of the field of view and the lateral boundary points as well as the centroid of the object are computed. The differences between these coordinates for a pair of objects constitutes the set of angular features.

The features are then used for classification of the aforementioned set of spatial relations with a linear Support Vector Machine (SVM) classifier. The local estimates generated by this are combined in a MLN model, which is specified by a simple set of rules formulated in first-order logic. This way, consistency between the relations between objects can be increased. The knowledge base encoded by the MLN encompasses formulas describing the antisymmetry of the chosen spatial relations,

$$\begin{aligned} \mathit{On}(o_1, o_2) &\implies \neg \mathit{On}(o_2, o_1) \\ \mathit{LeftOf}(o_1, o_2) &\implies \neg \mathit{LeftOf}(o_2, o_1) \\ \mathit{Behind}(o_1, o_2) &\implies \neg \mathit{Behind}(o_2, o_1), \end{aligned}$$

their transitivity

$$\begin{aligned} \mathit{On}(o_1, o_2) \wedge \mathit{On}(o_2, o_3) &\implies \mathit{On}(o_1, o_3) \\ \mathit{LeftOf}(o_1, o_2) \wedge \mathit{LeftOf}(o_2, o_3) &\implies \mathit{LeftOf}(o_1, o_3) \\ \mathit{Behind}(o_1, o_2) \wedge \mathit{Behind}(o_2, o_3) &\implies \mathit{Behind}(o_1, o_3), \end{aligned}$$

and the exclusivity of the *On* relation

$$\begin{aligned} \mathit{On}(o_1, o_2) &\implies \neg \mathit{LeftOf}(o_2, o_1), & \mathit{On}(o_1, o_2) &\implies \neg \mathit{LeftOf}(o_1, o_2) \\ \mathit{On}(o_1, o_2) &\implies \neg \mathit{Behind}(o_2, o_1), & \mathit{On}(o_1, o_2) &\implies \neg \mathit{Behind}(o_1, o_2). \end{aligned}$$

The *RightOf* relation is defined with a hard rule complementary to the *LeftOf* relation according to

$$\mathit{LeftOf}(o_1, o_2) \iff \mathit{RightOf}(o_2, o_1),$$

and thus not labelled or reasoned over separately. Similar reasoning applies to the relationship between the predicates *Behind* and *InFrontOf*.

The information from the baseline classifier is entered into the model with formulas of the form $P_{SVM}(o_1, o_2) \iff P(o_1, o_2)$ for each predicate P, where the predicate P_{SVM} represents the corresponding binary decision of the classifier.

A Markov Logic Network built from this knowledge base defines a probability distribution over the application of the predicates in the knowledge base to all objects in the domain. An example for a ground atom, which is the result of this procedure, is the application of the predicate $\mathit{On}(o_1, o_2)$ in free logical variables o_1 and o_2 to the constants *car #1* and *street #3* to form the ground atom *On(car #1, street #3)*, which can take a truth value as its assignment. The probability distribution over the ground atoms $\boldsymbol{x}$ is defined by the sum of formulas satisfied by the current state of the variables, weighted by a weight w_i associated with each formula $f_i, i = 1, \ldots, N$ as

$$P(o) = \frac{1}{Z} \sum_{i=1}^{N} w_i f_i(\boldsymbol{x}).$$

In this work, we are interested in an assignment to the variables that maximises this probability, the MAP estimate. As above in Section 3.2.2, we formulate the MLN MAP inference problem as a discrete optimization problem in the binary random variables constituted by the ground atoms. This optimization problem can easily be formulated as a factor graph and solved approximately with Loopy Belief Propagation. The approximate MAP solution that is obtained in this way assigns a truth value to each of the ground predicates, and thus determines which pairs of objects are in a certain spatial relation. This information is represented in the *spatial relations graph*, which has a node for each object present in the scene, and a labelled edge for every pair of objects that has a *true* value for any relation in the MAP MLN solution. The edges are labelled with the predicate of the corresponding relations. Note that a pair of objects can have multiple relations assigned to it, so an edge of the relation graph can have multiple labels. The spatial relations graph is part of the SRTree, where pairs of objects are annotated with the corresponding quantitative spatial information. Thus, the SRTree can be used to process queries like “List all objects that are to the left of a certain point on the sidewalk!”.

3.4 Experiments

The integrated system, comprised of building the SRTree representation of dense point clouds along with its segmentation and classification, as well as the inference of spatial relations based on this data and their incorporation into the SRTree structure, is presented for the entire urban dataset. The spatial reasoning approach is evaluated based on its capability to recreate the spatial relations given by manual labeling, as they would occur in a description of the scene.

3.4.1 3D Outdoor Urban Dataset

The dataset that is used for the experiments consists of 3D point cloud data enhanced by RGB image data using a Z+F 5010C laser range finder, which is shown in Figure 3.9. The dataset consists of 62 scenes in downtown Munich, covering a total area of roughly $2\,\mathrm{km}^2$. Images and point clouds for each scene have been manually segmented into objects and background, and objects are labelled with per-object class information as well spatial relations between objects. The nine classes used for the per-object annotation are *car*, *building*, *street*, *sidewalk*, *bicycle*, *other*, *tree*, *pole*, *sky*, and *grass*. The segmentation of images and point clouds was performed using an automatic segmentation [49] of the RGB

Figure 3.9: The Zoller & Fröhlich 5010C laser scanner used to generate the dataset. Image retrieved from `http://www.zf-laser.com`

images based on a graph-based segmentation procedure which was followed by a manual correction step.

Additionally, the spatial relations *On*, *LeftOf* and *Behind* have been manually added for object pairs in a half-space of each image, corresponding to the field of view of a person describing the scene.

The labelled data is used to learn the weights of the MLN and as a ground truth to evaluate the reasoning algorithm presented in this chapter. This results in a 3D semantic dataset defining segmented objects along with their classes and additionally equipped with a relation graph between objects.

Figure 3.10 shows an example scene with spatial relations represented by arrows connecting the objects.

3.4.2 Results

The MLN approach for spatial reasoning is evaluated on the dataset annotated with spatial relations as described above. The SVM and the MLN are trained on the same set of 48 labelled scenes; testing is performed on the remaining 14 scenes. For the SVM parameter learning, 5-fold cross validation is used. The formula weights of the spatial relations MLN are learned discriminatively using the Alchemy package[3]. Table 3.6 gives

[3] `http://alchemy.cs.washington.edu/`

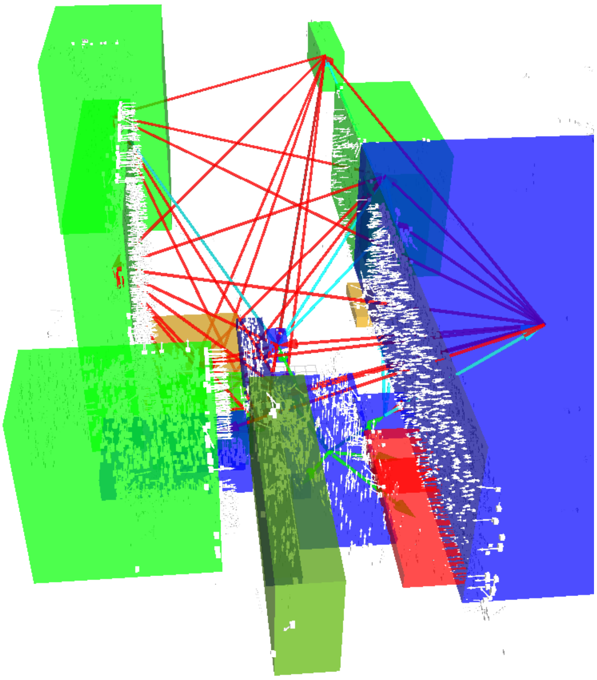


Figure 3.10: Annotated spatial relations and bounding boxes of scene objects. Red arrows stand for *LeftOf* relationships, blue ones for *Behind*, and green arrows are *On* relationships.

information metrics for the retrieval of spatial relations for all pairs of objects present in the test scenes, using the SVM formulation alone as well as the MLN model in addition to it. The main interest is in the correct identification of *true* values of relations, since these can be used for description of the environment. It can be seen that the SVM model slightly outperforms the MLN model on the *Behind* and *LeftOf* relations. The *On* relation however, which has the richest description in terms of rules in the MLN knowledge base, clearly profits from the added modelling effort in F_1 score.

3.5 Conclusion and Future Work

This chapter has described an efficient inference mechanism for probabilistic logic reasoning problems formulated as MLNs based on quadratization and subsequent pseudo-Boolean optimization, as well as the inference of spatial relations between objects in urban scenarios as an application of MLN reasoning in the domain of semantic mapping.

For inference in MLNs, we have shown a method to convert the weighted logical formulation of the problem to an equivalent pseudo-Boolean representation. This then allows the application of quadratization methods to arrive at an equivalent, larger problem with purely pairwise interactions, for which an approximate solution can be found using QPBO. The quadratized formulas can be efficiently computed on the first-order level. Ex-

Relation	Model	Value	Precision	Recall	F_1-score
On	MLN	False	0.99	0.99	0.99
		True	0.57	0.57	0.57
	SVM	False	0.99	0.98	0.98
		True	0.45	0.60	0.51
Behind	MLN	False	0.95	0.93	0.94
		True	0.37	0.45	0.41
	SVM	False	0.94	0.95	0.95
		True	0.44	0.38	0.41
LeftOf	MLN	False	0.92	0.77	0.84
		True	0.56	0.82	0.66
	SVM	False	0.91	0.86	0.89
		True	0.65	0.75	0.69

Table 3.6: Information retrieval metrics of the baseline SVM classifier and the added MLN inference for each of the three relations that were used.

periments on standard problems from the statistical relational reasoning literature show that the approach performs well in terms of both solution quality and computation time in comparison with competing methods.

There are various aspects of this research that can be expanded for fruitful future work. QPBO is only one, even though popular, inference method for pairwise Markov Models. Other inference methods, such as the family of methods based on *move-making* [102], may well provide additional benefits. Moreover, the presented algorithm only uses the logical structure of the problem for shattering and the computation of the quadratization on the first-order level. The combination of the presented work with lifting techniques or other higher-level partitioning methods, resulting in smaller networks, would be another interesting research direction. Finally, the work presented here has shown that the choice of quadratization method for a given problem matters; however, how to choose the quadratization for a given problem remains an open problem. Also in this matter, leveraging the knowledge about the logical structure of the problem may be useful, as well as a combination of the described work with an inference scheme over the chosen quadratization similar to the work of Gallagher, Batra, and Parikh [58].

Furthermore, in this chapter a novel environment representation titled SRTree is presented which generates a probabilistic 3D representation and captures the semantics of the environment. It uses probabilistic logical reasoning using the Markov Logic Network (MLN) formalism for inference of spatial relationships between objects. While the inference problems posed by this application are relatively small and can be solved using simple inference methods, its extension to larger scenes that extend to an area greater than the view of a single observer might require the application of highly efficient inference methods such as the one described in the first part of this chapter.

The inferred spatial relations can furthermore be stored in the SRTree representation, which can be useful in HRI interaction scenarios. The proposed framework is evaluated on a large-scale 3D dataset collected in downtown Munich and shows promising results.

Future work in the line of work of the semantic environment representation includes the automatic segmentation of objects, taking advantage of the hierarchical nature of the SRTree data structure. The spatial relations model in the MLN framework can be extended to be aware of object classes to incorporate a more natural usage of these terms—e.g., an object *behind* a car might well actually be on the side of it with respect to the current position of the viewer, but the direction of the road and the car will still enable the use of the quantifier *behind*. Additionally, the approach can also be extended to arbitrary viewpoints, and to larger scenes with a larger set of objects, where spatial relationships can also be inferred for objects that are not in direct view of a user.

4

Road Geometry Estimation for Urban Semantic Maps using Open Data

Complex robotic tasks require the use of knowledge that is impossible or very difficult to acquire with the sensor repertoire of a mobile, autonomous robot alone. For this reason, it is important to explore diverse sources of knowledge to integrate with a robot's sensor data in order to build semantic maps appropriate to demanding applications. For robots navigating in urban environments, geospatial open data repositories such as OpenStreetMap provide a source for such knowledge. In this chapter, the integration of a 3D metric environment representation with the semantic knowledge from such a database is proposed. The application described here is an instance of scene interpretation. It uses street network information from OpenStreetMap to improve street geometry information determined from laser data, which can then be used for high-level reasoning, for high-level navigation, or for interaction. The approach relies on a preliminary classification of the environment in street or sidewalk, and other areas, which then serves as the basis for a simple geometrical model of street layout. Semantic data is used for a coarse layout of the road network and in a final global smoothing step, where typical interactions between the geometries of adjacent road segments are taken into account. The approach presented here is evaluated on a challenging data set of point clouds from the urban environment in the Munich inner city.

The work presented in this chapter was published in [211].

4.1 Introduction

As tasks devolved to robots become ever more complex and encompass more domains, also demands towards their understanding of relationships and autonomy are growing. Different sources of knowledge that can be tapped for a higher-level understanding of concepts and tasks, which is desirable for a more intuitive and user-friendly interaction with a robot, have been explored. Human interaction partners themselves have been used as a knowledge source for example in the IURO project as described in Section 1.2, see Figure 4.1. Other approaches have considered the augmentation of robot knowledge using ontological models in databases that can be shared for learning and for usage by different robots [152, 178]. In this chapter, OpenStreetMap (OSM), a community-driven online mapping framework, is considered as a source for semantic information for robots moving autonomously in an urban environment. The chapter proposes the extension of a hybrid map, which includes a 3D occupancy grid as well as information about street and sidewalk objects in the environment, with semantic and topological information from this database.

There are multiple reasons why a tighter integration between robot mapping frameworks with data repositories like OpenStreetMap is beneficial. For once, these repositories contain manually selected and curated information, which ensures that it is specified on a level that is understandable to humans and thus usable in interaction, for example for giving or receiving route instructions. Crowdsourcing the data means that additions and modifications to the database are possible for the general public. Thus, the data is updated continuously, and errors can generally be detected and corrected quickly. Furthermore, even state-of the art scene understanding algorithms primarily rely on assigning labels on a per-pixel or per-region basis, and can have problems at determining distinctions between objects where this distinction happens primarily on a semantic level, i.e., two adjoining rooms with different functions in a space that is not clearly separated, or a building where different parts serve a different purpose. These will be hard to distinguish based on sensor data alone, but the information might be readily available as a bounding box in the OpenStreetMap annotation. On the other hand, the sensor repertoire used in robot mapping approaches will provide up-to-date metric spatial information in the near future, which can be uploaded to Open Data repositories for sharing with humans and other robots. Thus, the benefit of robots using open databases created by and for humans could be mutual.

This chapter describes applications and possibilities offered by integrating 3D metric maps with rich semantic and geospatial Open Data repositories. An overview over related approaches in literature is given in Section 4.2. The data contained in OpenStreetMap that is relevant to this chapter is described in Section 4.3, and as an application scenario, it is described how street network information from OpenStreetMap can be used to improve understanding of street geometry based on 3D laser data in Section 4.4. The approach is evaluated on a challenging data set covering an area in downtown Munich. The results are presented in Section 4.5, and the results of the chapter are summarized in Section 4.6.

4.2 Related Work

Data retrieved from OpenStreetMap and similar information sources, in particular information about the topology and layout of the street network, has been used for multiple applications in robotics. An important requirement for the use of geospatial data is knowledge about the location of the robot on a global map, i.e., a solution of the localization

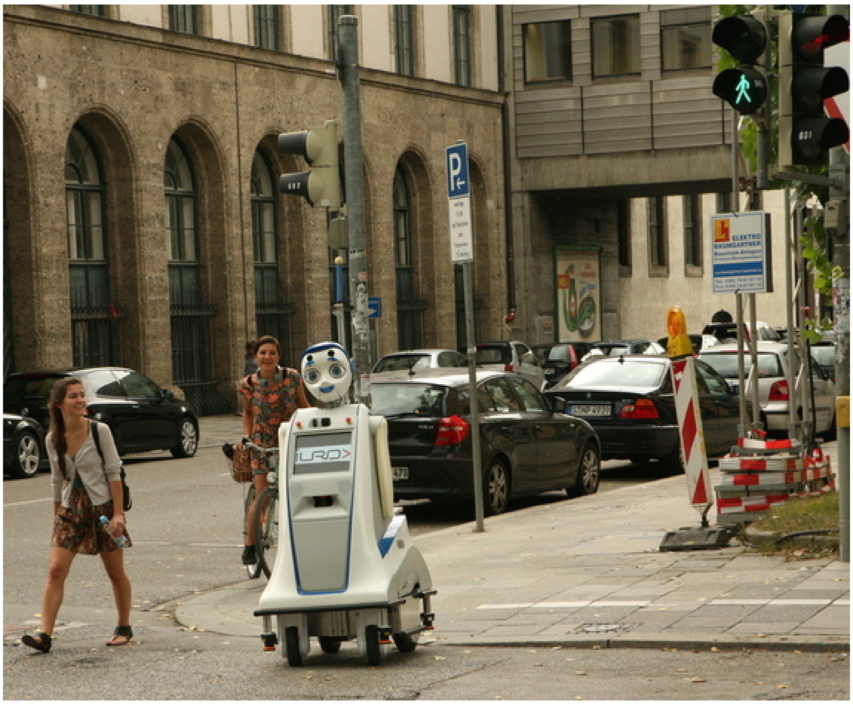


Figure 4.1: The robot IURO [204, 214] in an urban environment

problem. Hentschel and Wagner describe a localization method that uses building outlines from OpenStreetMap, which are matched to corresponding features in 3D laser scans [74]. Additionally, the work covers route planning on the OpenStreetMap route network, and robot behavior control for the robot car's lights based on semantic attributes from OpenStreetMap. An alternative to this localization approach is described later in this thesis in Chapter 5. Brubaker, Geiger, and Urtasun use the OpenStreetMap route network to localize based on visual odometry data [23]. The localization problem is modeled as a dynamic system, where the state is the vehicle position related to the current route segment, and the visual SLAM trace is the input for filtering. An approach for localisation on the OpenStreetMap route network with visual SLAM and an initial guess from GPS is presented by Floros, Zander, and Leibe [52]. The result of visual odometry is used as input to a particle filter, where the distribution is pruned based on comparison with the OpenStreetMap street network. Recently, Ruchti et al. described localization of a robot on the OpenStreetMap global map based on classification of 3D laser scan point clouds in street and non-street regions in a SLAM framework [159].

Geospatial data from open data repositories has also been used for the applications of place detection and image localization. 3D building geometries from sources similar to OpenStreetMap and vanishing point detection can be used to rectify and align training and query images for place recognition tasks [11]. Li et al. [106] use 3D point clouds for global registration of images. The 3D point clouds are generated with Structure of Motion techniques from crowdsourced, geotagged monocular images.

The application considered in this chapter, estimation of street geometry using information about the location of the street center from Open Data sources is also treated by Yuan and Cheriyadat [200], where aerial images are used as sensor data in combination with a street vector network, and by Chen, Sun, and Vodacek [30], where street geometry is inferred on the basis of high-resolution multispectral remote sensing satellite imagery. Geiger, Lauer, and Urtasun present an approach for urban scene understanding based on a generative model for street geometry and topographic information that is based on 3D data constructed from stereo vision recorded by a vehicle traveling on the street [60].

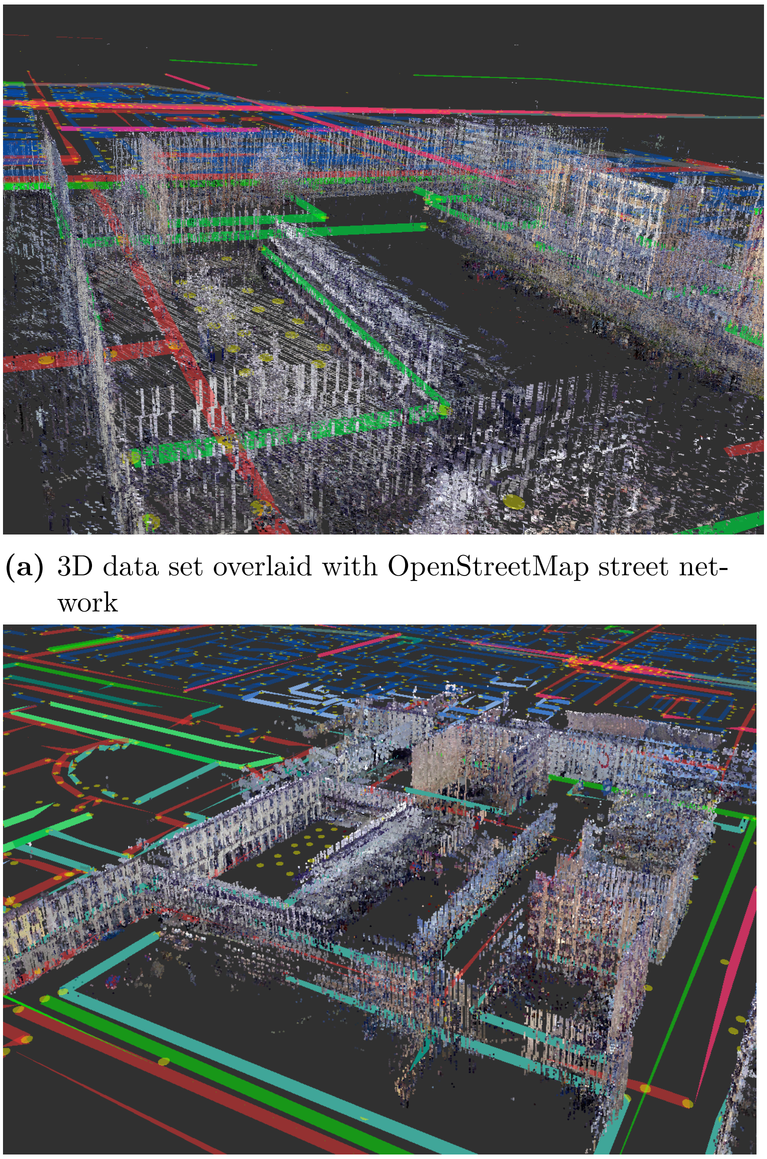
The key difference between these works and the approach presented here is in the data used for estimation. The approach presented here is designed for a static 3D point cloud, where no pose history or dynamics of other dynamic agents (e.g., cars) are available. Moreover, the data is recorded as if from a robot travelling on the sidewalk, such that large parts of the street may be occluded by parked cars or dynamic objects.

4.3 OpenStreetMap Data Model and Relevant Data

The data model of OpenStreetMap is a graph-like structure, where the basic building blocks are *nodes*, *ways* and *relations*. *Nodes* represent points on the map and are characterized by their latitude and longitude, as well as an optional elevation. *Ways* connect nodes to form open or closed paths and represent spatial entities like the path followed by railroad tracks, building outlines or the area covered by a football field. *Relations* describe higher-level characteristics of sets of nodes and ways, like all buildings belonging to an university campus, or the complete set of streets followed by a bus route. All instances of these three building blocks are identified by globally unique identifiers. Moreover, arbitrary tags can be applied to each instance of these data types, although there is an established set of tags and values that is largely adhered to, which can be used to automatically extract semantic information.

Many features from OpenStreetMap can be easily transferred to a metric map used for robot applications, provided that the transformation between the different global coordinate systems is known. Different localization approaches to address this problem have been proposed as summarized in Section 4.2, and this transformation is assumed to be known for the purposes of the work presented here. In this case, the mapping of spatial locations allows the transfer of features between the two maps, for example for route planning based on street addresses in an occupancy grid derived from sensor data, or for identifying all buildings belonging to a particular ensemble in a 3D map, as exemplified in Figure 4.2.

Since the positions of nodes in OpenStreetMap are based on manual placement, which in turn is based on processed GPS data and aerial imagery, it is difficult to give an accuracy estimate. The errors depend on the accuracy of the recorded GPS data, the number of data points, where there usually is more data in cities and places of frequent travel, and the diligence and skill of the annotators. Hentschel and Wagner [74] give a visual



(a) 3D data set overlaid with OpenStreetMap street network

(b) Buildings on the TUM campus, extracted from OpenStreetMap building outlines

Figure 4.2: Examples for combinations of 3D laser data with additional RGB information and information from OpenStreetMap. The visualizations of OpenStreetMap data in this chapter are created with software based on the `open_street_map` ROS package[1].

comparison of a ground truth cadastral map with the building outlines extracted from OpenStreetMap in the context of robotic navigation. This can be interpreted to show errors in the building edges of a few metres. Fan et al. [48] performed a quantitative analysis of positional accuracy of building outlines for the city of Munich, which, they state, is one of the most developed cities in OpenStreetMap. Their investigation showed an average positional error of 4 m with respect to administrative mapping data. Further analysis by the authors indicate a accuracy of building outlines also based on other measures, such as alignment between building outlines and neighbouring street segments, for the area of Baden-Württemberg in Germany [47]. Other research performed on road positions in other parts of the world give similar figures, e.g., the work done by Haklay [69] and Antunes et al. [6]. This relative inaccuracy is one reason for fitting the geometry parameters based on sensor data, since this could be used to improve the positional accuracy of the OpenStreetMap data.

4.4 Street Geometry Estimation using Street Topology Information

The approach for street geometry information presented here is related to the work by Ruchti et al. [159], where cells of a 3D laser-based map are classified point-by-point in order to enable localization of a robot in a street network like OpenStreetMap. In the work on semantic mapping presented here, additionally, the modelling imposes a strong geometric consistency constraint—street cells have to be adjacent and located in a strip around the street center. Depending on the intended robotic application, the term 'street' can be understood as either only the area of the street that is driven on, including parking spaces on the side of the road, or the combination of this drivable area with the sidewalk directly beside it. The simple model for street geometry used here incorporates both cases, but is not applicable if there is a larger spatial separation between drivable area and sidewalk.

For the work presented here, topological information about the street network is extracted from OpenStreetMap. The goal is to augment this graph with additional metric information in the form of street geometry, which is largely not existing as annotation in the OpenStreetMap database. This relies on the street network data being available and sufficiently accurate. This is the case for the regions considered in the evaluation of this chapter, and has also been found to suffice for the different purposes of the other works that use street network data and perform evaluation on data from other parts of the world. However, street *width*, even though the infrastructure (an attribute tag defined for the purpose of annotating it) exists, is not annotated often. In the data set used for evaluation in this chapter, only one street segment is annotated with a width tag in the OpenStreetMap database.

4.4.1 Modelling Street Geometry Information

Basis for the estimation is the street network from OpenStreetMap, which provides approximate street center lines subdivided into segments of varying length, within which

[1] authored by Jack O'Quinn, `https://github.com/ros-geographic-info/open_street_map`.

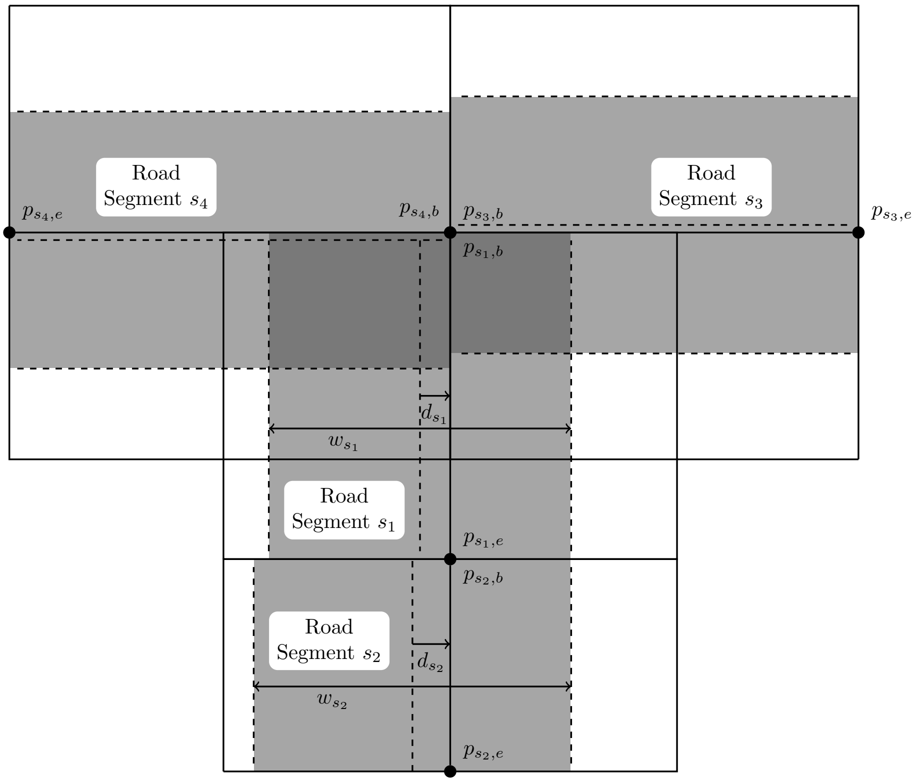


Figure 4.3: Illustration of street position and width model

the street is assumed to be straight. In order to reconcile this information with a metric 3D representation, two parameters need to be estimated for each street segment s: The vertical offset d_s of the actual street center from the vector connecting the OpenStreetMap waypoints $p_{s,b}$ and $p_{s,e}$ defining the street segment in the street network, and the width w_s of the street around this actual center line. This model for the layout is displayed in Figure 4.3. Let the joint geometrical parameters for segment s be denoted by $\theta_s = (w_s, d_s)$, and the full set of parameters for all segments by $\mathbf{\Theta}$. The directions of the street segments from OpenStreetMap are assumed to be in keeping with the actual topology of the environment.

While this simple model of street geometry fits well with the interior of street segments, it does not cover intersection areas, as can be observed in Figure 4.3. In such areas, the vertical strip of the intersecting street does not conform well with this model. Since intersection points are known from the street network information from OpenStreetMap, this information can be used to mask intersection areas for the purpose of street geometry estimation. The approach described below considers only areas that are at least 10 m away from the middle of an intersection, so that the considered environment can be assumed to have the strip-like geometry expected under this model.

4.4.2 Inferring Street Geometry from 3D Laser Data and Street Network Information

The approach for estimating street width from 3D laser data followed here is a two-step process. Firstly, based on the topology information from OpenStreetMap, areas of interest which contain the street segments are extracted from the point cloud of the covered area. In each of these areas, the information from the point cloud is condensed to a 2D grid, where features are computed for each bin. A binary classifier provides an estimate about the assignment of each bin to the street or non-street class. Based on these estimates, the geometrical parameters for each street segment are determined by maximizing their probability determined through a Graphical Model.

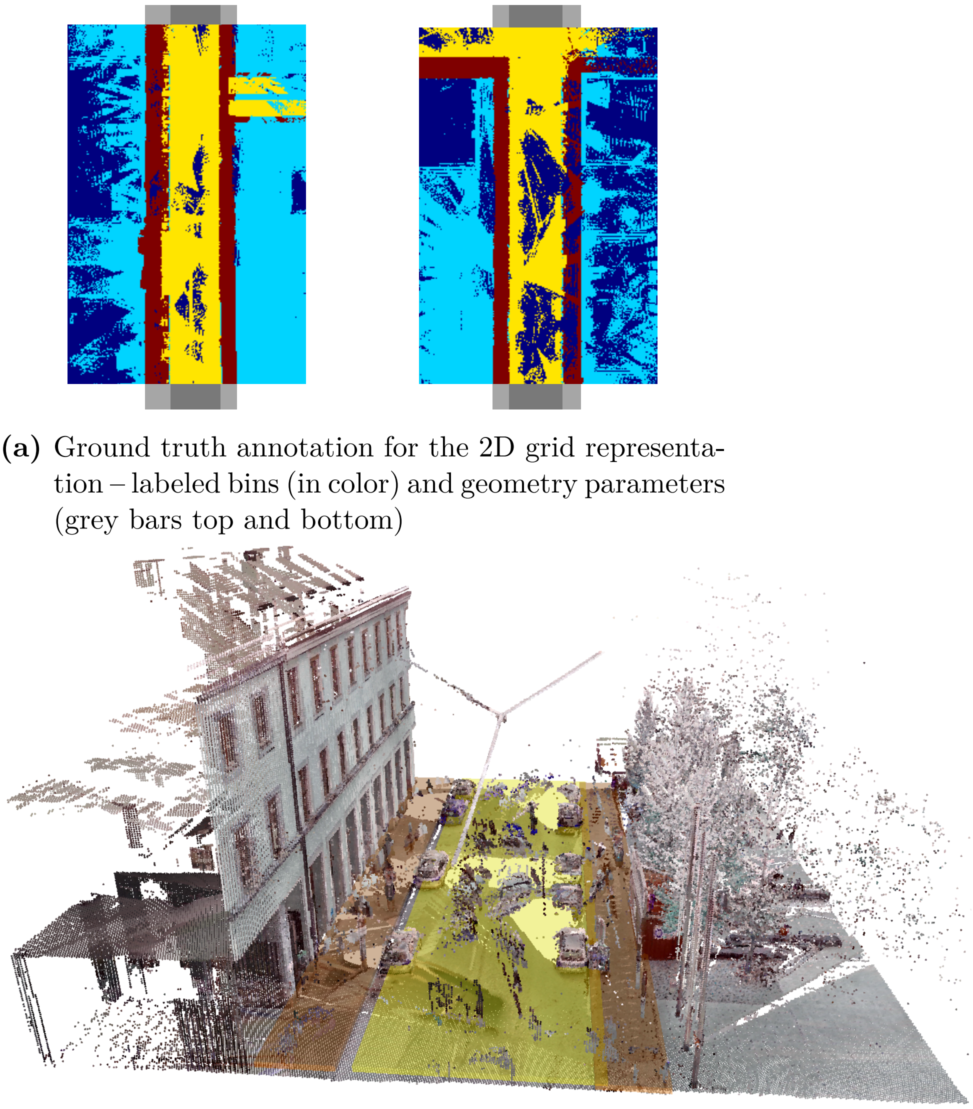
The connectivity of the street network is encoded in a graph $\mathcal{G} = (P, S)$. Its edges $s \in S$ are the street segments in the relevant area extracted from OpenStreetMap, and the nodes $p \in P$ are the corresponding waypoints. Each segment connects two waypoints $p_{s,b}$ and $p_{s,e}$. This graph reflects the street topology and is used to model dependencies between the parameters of neighbouring street segments. From this graph, the set of pairs of neighboring segments N can be derived as $\{\{s_1, s_2\} : s_1, s_2 \in S \wedge p_{s_1,b} = p_{s_2,e}\}$.

The positions of the start and end nodes of each segment also determine the area that is considered for estimating the street width. For this task, a candidate environment of a predefined width around each segment center line from the street network is retrieved from the 3D map. For the experiments reported in this chapter, a total width of 40 m was chosen. This section of the map is then discretized in the ground plane, such that each segment is divided into a rectangular grid of bins of size $L \times N$, where L is the number of bins in the direction parallel to the street, and N the number of bins in the considered area vertical to the direction of the street. The length of the sides of the square bins are chosen as 0.2 m.

An illustration of this representation of the environment is given in Figure 4.4. Figure 4.4a shows two projected 2D segments with bins labeled according to their class membership. It also shows the street segment geometries for the drivable area and the sidewalk annotated for these segments. Figure 4.4b shows the full point cloud of one road segment.

For each resulting bin a set of local features is computed. The feature set contains standard geometric and appearance-based features. The geometric features are comprised of the mean, median, standard deviation and absolute range of the z-coordinates of all points projected to each bin as well as the polar angle of a normal vector computed for a small neighborhood around each point. Appearance-based features consist of the same statistics for the intensity values as well as histograms of the recorded color values in each bin in the RGB and HSV color spaces.

Using these features, a baseline classifier is trained to separate between street and non-street bins. For this binary classification problem, the labels are chosen as 0 for non-street bins and 1 for bins classified as belonging to a street. For the experiments described in this chapter, a Support Vector Machine (SVM) with radial basis function kernel is used for this



(a) Ground truth annotation for the 2D grid representation – labeled bins (in color) and geometry parameters (grey bars top and bottom)

(b) Section of 3D map with overlaid ground truth annotation

Figure 4.4: Example annotated street segments. For a, yellow areas belong to the drivable area, brown areas are on the sidewalk, light blue areas are non-street and for dark blue areas, no features are available because of occlusions. The grey blocks in the background show the annotation of the geometry parameters—light grey for the sidewalk; dark grey for the drivable area of the street. For b, drivable area and sidewalk points are overlaid over the point cloud in yellow and brown, respectively.

purpose. The result of the classification for segment s is a matrix of labels $Z_s \in \mathbb{B}^{L\times N}$, containing the classification result for each bin $z_s\left[i,j\right], 0 \leq i < L, 0 \leq L < N$. The entirety of estimates for all segments is denoted by Z. Additionally, the confusion matrix

$\boldsymbol{C}$ of the classifier can be determined from the labeled data used for training the classifier.

The classification results then provide candidate information for the second step, which introduces the strong global geometric constraint described above on the inferred street geometry, i.e., that each street segment has straight parallel side lines.

These constraints are formalized in a probabilistic graphical model which encodes both the dependency of the geometrical parameters of a single segment on the estimate provided by the classifier, as well as dependencies of the geometrical parameters of neighbouring segments. The probability of a set of geometrical parameters is modeled by

$$P(\boldsymbol{\Theta}|Z) = \prod_{s_1,s_2 \in N} P(\theta_{s_1}, \theta_{s_2}) \prod_{s \in S} P(\theta_s|Z_s). \tag{4.1}$$

The factors in the rightmost product of (4.1), the *segment geometry potentials*, describe the dependency of a segment geometry on the raw classification result. With the help of the definition of the matrix $\boldsymbol{X}_s(\theta_s) \in \mathbb{B}^{L \times N}$, which describes the labels assigned to each bin of the segment $x_s\left(\theta_s\right)[i,j]$ under a specific geometry θ_s, this can be further developed as

$$\begin{aligned} P(\theta_s|Z_s) \propto P(\theta_s)P(Z_s|\boldsymbol{X}_s(\theta_s)) \approx P(\theta_s) \prod_{i,j} P(z_s[i,j]|x_s\left[i,j\right]\left(\theta_s\right)) \qquad (4.2)\\ = P(\theta_s) \prod_{i,j} P(x_s\left[i,j\right]\left(\theta_s\right), z_s[i,j]) / P(x_s\left[i,j\right]\left(\theta_s\right)) \\ = P(\theta_s) \prod_{i,j} \boldsymbol{C}\left[x_s\left[i,j\right]\left(\theta_s\right), z_s[i,j]\right] / P(x_s\left[i,j\right]\left(\theta_s\right)). \end{aligned}$$

The confusion matrix $\boldsymbol{C}$ is used as an estimate of the classification error probability, and $P(x_s\left[i,j\right]\left(\theta_s\right))$ denotes the class marginals. This distribution penalizes street segment geometries where many bins receive a label that is different from their initial classification result.

The prior distribution for the street geometry parameters $P(\theta_s)$ is chosen as a product of independent normal distributions for the segment offset and the logarithm of the street width as $P(\theta_s) = \mathcal{N}(d_s|\mu_d, \sigma_d)\mathcal{N}(\log(w_s)|\mu_w, \sigma_w)$, the mean and variance of which are estimated from a training set of segments.

The factors in the first product in (4.1), the *intersegment potentials*, serve the purpose of relating the geometrical parameters of neighbouring street segments. By transforming the geometry specified in offset from the segment center and the street width to the left and right boundary of the street according to

$$\begin{aligned} a_s &= d_s - w_s/2 \\ b_s &= d_s + w_s/2, \end{aligned}$$

it is possible to define a measure for the mismatch between the parameters of the two segments as

$$P(\theta_{s_1}, \theta_{s_2}) = \mathcal{N}(a_{s_1} - a_{s_2}\,|0, \sigma)\mathcal{N}(b_{s_1} - b_{s_2}\,|0, \sigma).$$

The value of σ is set to 0.5 for the experimental evaluation.

Given the model (4.1), street offset and width for all segments are determined as the parameters which maximise the probability density function as

$$\boldsymbol{\Theta}^* = \operatorname*{argmax}_{\boldsymbol{\Theta}} P(\boldsymbol{\Theta}|Z). \tag{4.3}$$

4.4.3 Inference in the Graphical Model

Exact inference of (4.3) is intractable for street networks of general structure, since the street network may contain large loops. In order to obtain an approximate solution for the street geometry optimization problem, a Markov Chain Monte Carlo (MCMC) approach is used. For this computation, the values of the potential functions must be determined either analytically or in tabular form for all possible arguments.

While the intersegment potentials are normally distributed and thus have analytical expressions, the segment geometry potentials (4.2) depend on the classification results in a more complex fashion. This table of values can be computed efficiently using a dynamic programming approach. This is best seen by transforming the probability into its logarithmic form so it can be expressed as a summation over the individual bins (i, j) of the segment grid. Since the computation can be done independently for each segment, the corresponding subscript s is dropped in the following.

$$\begin{aligned}
\log(P\left(Z|\boldsymbol{X}(\theta)\right)) &= \log\left\{\prod_{i,j} \boldsymbol{C}\left[x\left[i,j\right](\theta), z[i,j]\right]/P(x_s\left[i,j\right](\theta_s))\right\} \\
&= \sum_{i,j} \log(\boldsymbol{C}\left[x\left[i,j\right](\theta), z[i,j]\right]/P(x_s\left[i,j\right](\theta_s)))
\end{aligned}$$

Replacing the summands in this expression with the expression $\tilde{c}(x\left[i,j\right](\theta), z[i,j])$, one can use the fact that the labels assigned to each bin are identical in the regions assigned as sidewalk and street, respectively, to simplify the expression:

$$\begin{aligned}
\log(P\left(Z|\boldsymbol{X}(\theta)\right)) &= \sum_{i,j} \tilde{c}(x\left[i,j\right](\theta), z[i,j]) \\
&= \sum_{j=0}^{a-1}\sum_{i=0}^{L} \tilde{c}(0, z[i,j]) + \sum_{j=a}^{b}\sum_{i=0}^{L} \tilde{c}(1, z[i,j]) + \sum_{j=b+1}^{N}\sum_{i=0}^{L} \tilde{c}(0, z[i,j]) \\
&= \sum_{j=0}^{a-1}\sum_{v\in\mathbb{B}} \tilde{c}(0,v)\sum_{i=0}^{L}\delta(z[i,j]=v) + \sum_{j=a}^{b}\sum_{v\in\mathbb{B}} \tilde{c}(1,v)\sum_{i=0}^{L}\delta(z[i,j]=v) \\
&\qquad + \sum_{j=b+1}^{N}\sum_{v\in\mathbb{B}} \tilde{c}(0,v)\sum_{i=0}^{L}\delta(z[i,j]=v)
\end{aligned} \tag{4.4}$$

with the indicator function δ. Then, each of the inner summations does not depend on the geometry any more, but only on the number of bins that have been classified as sidewalk or street in each row of Z. Thus, the value table for these factors can be efficiently computed for all possible segment geometries θ.

4.5 Experimental Evaluation

4.5.1 Munich Urban 3D Data Set

The data set that was used for experiments is in part overlapping with the one described in our earlier work on spatial relations in semantic maps [214]. It consists of 80 high-resolution laser range finder scans in 3D, acquired with an Z+F 5010C laser range finder, of an area in downtown Munich around the university campus. Additionally, laser intensity and RGB channels are recorded; GPS and odometry data are however not available. In this data set, object instances are manually segmented and annotated for object classes such as *building*, *street*, *sidewalk* or *car*, as well as for qualitative spatial relations, such as *left of* or *behind*, between objects. All OpenStreetMap street segments covered by the point cloud are annotated with class labels for street, sidewalk or neither of the two on a per-point level, and ground truth street geometry parameters are determined, as shown in Figure 4.4. This street network comprises a set of 60 route segments with a total length of about 2 km.

The data set provides a challenging environment for scene understanding tasks, since it incorporates a considerable range of different environments, such as residential streets with parked and artefacts of moving cars, tunnels, and cobbled or gravelled streets closed for motor vehicles. Additionally, the laser scans are taken from positions on the sidewalk, such that in many cases the ground plane is not visible because of occlusions or dynamic objects blocking visibility at the time of registering the laser scan.

4.5.2 Registration of Point Cloud Data with OpenStreetMap

Since the data set is recorded sequentially with no ground truth information about the absolute robot position at the time of recording a scan, nor about the relative movement of the sensor between scans, a registration step is necessary to obtain a complete 3D representation of the area covered by the union of the different laser scans. To this end of estimating the relative transformations between the sensor positions for each recorded 3D scan, registration was carried out with multiple iterations of the 3D Iterative Closest Point algorithm [19], with the maximum allowed correspondence distance decreasing with each iteration, starting from a rough manually defined initial guess. Boundedness of the registration error was ensured by manually labeling key points for pairs of scenes and monitoring the registration error, and by visual inspection of the registration result. Also for the lack of a global ground truth position data of the laser data, manual alignment of the 3D data with an export of OpenStreetMap data for the region covered by the laser data was carried out. This alignment was based on positioning building outlines in OpenStreetMap in accordance with vertical surfaces in the point cloud, and the correctness of the alignment was determined by projecting the OpenStreetMap data into the point cloud data. Note also that the accuracy of this alignment is not critical for the experimental evaluation as long as the road segments of interest in the point cloud are inside the regions of interest defined by the road network. This was verified visually.

Since handling the complete point cloud data for the combination of all laser scans is intractable, the data was filtered and downsampled using the RMAP algorithm [85]. This procedure produces a denoised occupancy grid at a variable resolution, where the grid size was chosen as 0.03 m for the experiments in this chapter.

4.5.3 Experiments

The method for street geometry estimation described above was evaluated on this augmented Munich 3D Urban Data Set. In order to evaluate the benefits and limitations of the method as well as to gauge the influences of the different components of the model, a set of computational experiments with different settings was run.

As goals of the inference, two different applications were investigated: First, the target was to estimate the geometrical parameters of the drivable area of the street alone, counting all surrounding area as non-road. Secondly, the target area was defined to include both sidewalk and the drivable section of the street. The geometric model described in Section 4.4.1 can be used in both cases; the experiments are only distinguished by the choice of target class in the training of the baseline classifier.

The requirement of labeled training data for the training of the classifier, the calculation of the confidence matrix for the computation of the segment geometry potentials (4.2), and the parameters of the segment geometry prior requires splitting the data set into a training and a test set for evaluation. In order to be able to evaluate the full model, including the intersegment dependencies, on the complete available graph of street segments, a round robin scheme was adopted for the supervised training. For this, the data set was split into 5 folds, for each of which a classifier was trained on data from the 4 remaining ones. These were used to compute the potential value tables using (4.4). Then, inference was carried out with the full model including the intersegment dependencies on the full street network. For inference in the full model, 50 chains of Markov Chain Monte Carlo (MCMC) inference were run for 1,000,000 iterations each.

A summary of the results in terms of per-bin retrieval of the correct labels, measured against the ground truth per-bin labelling. This metric, expressed as precision, recall and F_1 score, is given in Table 4.1. Additionally, a measure for the error of the estimated street widths against the manually labelled ground truth geometry parameters is given by the root mean squared error between the estimated segment width w_s^* and the true

Target Area	Method	P	R	F_1	$RMSE_w[m]$
driving area & sidewalk	raw	0.854	0.854	0.854	N. A.
	single segment	0.877	0.872	0.873	5.67
	full model	0.872	0.869	0.87	2.62
	opt. on ground truth	0.946	0.943	0.943	1.58
	ground truth geometry	0.943	0.937	0.938	0
driving area only	raw	0.815	0.808	0.81	N. A.
	single segment	0.856	0.851	0.852	5.61
	full model	0.876	0.876	0.876	2.65
	opt. on ground truth	0.955	0.953	0.953	1.54
	ground truth geometry	0.951	0.947	0.947	0

Table 4.1: Database recall metrics and root mean square error of the estimated street widths for the baseline classifier and for the solution including geometric constraints. The upper part of the table contains result for the case where the sidewalk is included in the street; the lower doesn't include the sidewalk in the street area.

annotated width $\hat{w}_s$, weighted by the length of each segment l_s

$$RMSE_w = \sqrt{\frac{\sum_{s \in S} l_s (w_s^* - \hat{w}_s)^2}{\sum_{s \in S} l_s}}.$$

Different configurations of the method are evaluated. First, the baseline classifier by itself is evaluated (*raw*). Since its result do not include geometry information, no width error is given. Then, the geometries resulting from the probabilistic model are evaluated with (*full model*) and without (*single segment*) including the intersegment potential functions. As an upper bound to the achievable results, the geometric model is fitted to the per-bin ground truth class labels (*opt. on ground truth*). The final evaluation is the evaluation of bin-wise labels implied by the ground truth geometry (*ground truth geometry*). An illustration of the resulting geometries for a part of the data set in comparison with ground truth data is shown in Figure 4.5.

4.5.4 Analysis of the computational properties

For a brief analysis of the computational properties, the processing pipeline can be separated into the following individual processing steps:

1. creation of the RMAP 3D occupancy grid

2. feature extraction

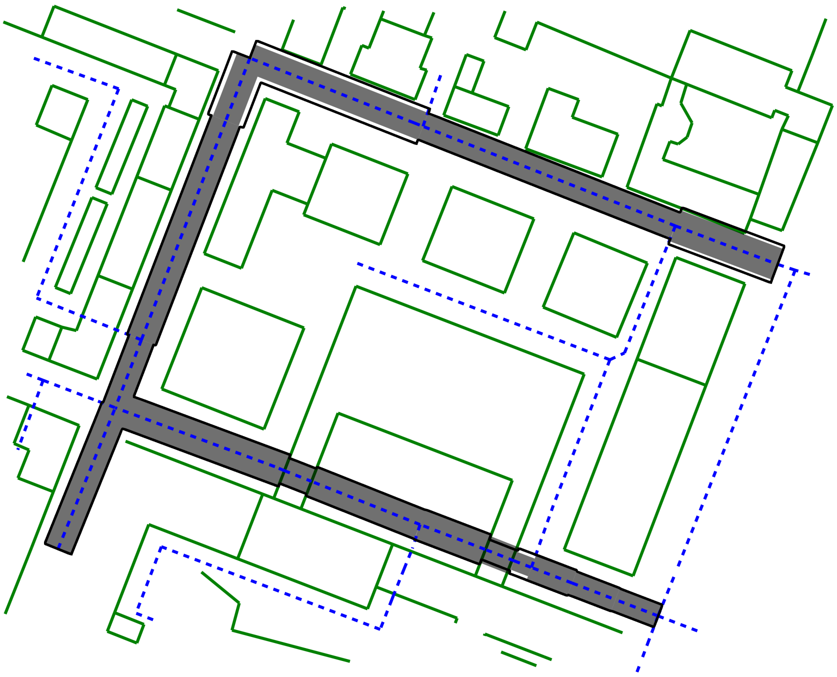


Figure 4.5: A part of the map with the estimated road geometries. Street center lines and building outlines are drawn as dashed blue and solid green lines, respectively. The estimated road geometries are shown as black outlines and ground truth geometries are shown in grey.

3. SVM prediction and segment geometry cost function estimation

4. global optimization with geometric model

5. SVM training

The current implementation does all these steps in an offline fashion on the full data set, in a proof-of-concept, largely non-optimized python implementation. For an online application of the method with a pre-trained model, only steps 1 – 4 would be relevant. The complexity of step 1 is linear in the number of points contained in the point cloud, and thus depends on the rate of discovery of the environment (speed of a vehicle) and the structure of the environment. The computational complexity of steps 2 and 3 is linear in the number of bins that are added at each step, thus also proportional to the speed of the vehicle. The global optimization step is NP-hard if the road network contains loops; however, the quality of the approximate solution that is practically used can be traded off against computation time by adapting the parameters of the MCMC inference algorithm. Furthermore, the influence of new observations will generally be local and not affect the global solution, such that efficiency gains could possibly be made by adapting the sampling scheme accordingly.

	Computation time [min]
RMAP total	47.2
features total	104.2
SVM training and prediction (parallel)	12.0
cost computation	4.8
MCMC inference (parallel)	2.8

Table 4.2: Computation times for processing the whole dataset

	Computation time [ms]
features per bin	5.39
SVM prediction per bin	2.22

Table 4.3: Overview over computation times per bin

The runtime of the individual steps for the whole dataset of the current implementation is distributed as given in the Table 4.2.

The SVM training and prediction step as well as the MCMC inference use a parallelized approach on a maximum of 64 cores; all other steps run on a single core on state-of-the-art hardware.

This distribution shows that much computation time is spent on preprocessing of the data, which was not a focus for the evaluation done. These steps could probably be sped up considerably by optimizing the implementation, and would also benefit from parallelization.

The SVM training is one bottleneck of the approach. Adding new training data for more diverse environments would affect computation times greatly, since the complexity of nonlinear SVM training is approximately cubic in the number of samples in the worst case. However, a tradeoff between complexity and quality could be reached by subsampling the training data, and trying to distribute the training samples evenly across different types of environments (which could be distinguished by OpenStreetMap metadata, for example).

Regarding the real-time usability of the approach, a brief assessment can be made using the per-bin processing times of the current implementation given in Table 4.3.

This shows that the processing time for the robot advancing 1 m is approximately 7.6 s (the product of the width of the observed area, 40 m, 25 bins per square meter of area covered, and the sum of the times given above). Recomputation of the global solution is not necessary at the same rate, and should be based on the occurrence of specific events, such as whenever a landmark is reached. In the current experimental setting, where the recording of a scan takes about 15 minutes and the displacement between scans is about 20 m, this is sufficient for “real-time” implementation; however, for a moving robot, it would not be fast enough. Nevertheless, the author is confident that an actual real-time operation would be possible with moderate optimization and parallelization of the execution.

4.5.5 Discussion of the results

It can be seen in Table 4.1 that introducing the geometric constraints improves retrieval metrics of labels for individual regions, as well as it also decreases the error in the street width estimation. An analysis of the failure modes on segments where street geometry estimates exhibited larger errors showed that environments were the street area is directly adjacent to an open space with a surface very similar to the street were difficult to handle for the estimation procedure. Additionally, the data set also contains streets of different categories (i.e., residential urban streets as well as cobbled streets closed for general traffic and without sidewalks as well as tunnels), which again are quite different in nature from a generic scene. In order to further improve the geometry estimation results, more qualitative information from OpenStreetMap could be used, for example by building and employing different models for streets of different categories, or street segments that are annotated as tunnels.

4.6 Conclusion

In this chapter, the benefits of including information from open geospatial repositories in hybrid maps have been demonstrated. The application of street classification and street geometry estimation, parameters which are often missing in OpenStreetMap and could be added automatically from 3D maps, has shown that including a geometric constraint based on OpenStreetMap data provides an improvement in the geometry error over a baseline solution based on classification alone. Experiments have been carried out on a challenging data set, where laser scans have been recorded from the sidewalk, so that the full width of the street is often occluded, and which contains a widely varying array of street types, including tunnels. With the increase in mobile robot platforms navigating in urban scenarios that are equipped with a 3D laser scanners, it is to be expected that different avenues for use of additional information will be explored.

There are several directions in which the work presented here can be extended. Especially in the vein of improving urban scene interpretation by using mapping data from OpenStreetMap would be the use of information about additional properties of streets such as traversability and the existence of bike paths and sidewalks. Furthermore, it can be expected that knowledge about the type of street from the annotation as *residential*, *primary*, *secondary* etc. will be useful if separate models are built and conditioned on the different types of environment.

5

Global Localization of 3D Point Clouds in Building Outline Maps of Urban Outdoor Environments

This chapter presents a method to leverage semantic maps for the task of localization based on sparse semantic data. The presented method is able to localize a robot with high accuracy in a global coordinate frame based on a sparse 2D map containing outlines of building and road network information and no location prior information. Its input is a single 3D laser scan of the surroundings of the robot. The approach extends the generic Chamfer Matching (CM) template matching technique from image processing by including visibility analysis in the cost function. Thus, the observed building planes are matched to the expected view of the corresponding map section instead of to the entire map, which makes a more accurate matching possible. Since this formulation operates on generic edge maps from visual sensors, the matching formulation can be expected to generalize to other input data, e.g., from monocular or stereo cameras. The method is evaluated on two large datasets collected in different real-world urban settings and compared to a baseline method from literature and to the standard chamfer matching approach. This evaluation shows considerable performance benefits of the novel localization method, as well as the feasibility of global localization based on sparse building outline data.

The presented approach was published in [210].

5.1 Introduction

Accurate localization in urban environments is a crucial dependency of many developing robotic applications, such as autonomous vehicles, delivery and service robots, or augmented reality applications. While systems like the Global Navigation Satellite System (GNSS) or localization based on wireless signals are sufficient for many applications, there is a benefit to a robot being able to localize based purely on its own sensors in cases these external services are unavailable or lacking in accuracy. In urban and highly structured environments, large, usually artificial, planar structures provide robust features for localization and registration of 3D sensor data [136]. Many vertical planes in urban environments are represented in human-readable maps as building outlines, such that a mapping between the two allows to localize a robot in the global map coordinate frame. This chapter describes a method to perform this localization based on data from a 3D laser range finder, for example for a robot travelling in an urban environment, in a 2D map containing building outlines. Such map information is freely available from common online map sources like OpenStreetMap [70], Google Street Maps or official municipal cadastral maps. As discussed in Section 4.3, the information contained in such maps is reasonably accurate for the use in robotic localization and navigation as well as semantic mapping, and its accuracy is steadily improving. The proposed localization method uses only information about building outlines and the street network, which keeps its demands for storage capacity or bandwidth low. It is based on the geometry of the environment alone, without the requirement of visual features such as appearance or texture data. Thus, it is largely independent from seasonal variation or variation based on the time of day. The matching procedure needs a single 3D laser scan as input. Therefore, no odometry or time series of measurements is necessary. As a global localization method using an external map, it is not necessary for the robot to have visited the location before or to build a feature database for the purpose of localization, since all necessary map information is freely available online.

The localization problem as posed here is an instance of the template matching problem: Finding a relation between the query features, consisting of the planar segments in the robot observation, and the building outlines in the map. Theoretically, this problem could be solved by knowing the correspondence between a single observed plane and one building edge in the map; however, this correspondence problem is highly nontrivial, especially when no appearance information is used.

As for all localization methods, the environment needs to contain a sufficient amount of salient information to uniquely distinguish it; the lack of this uniqueness is known as *perceptual aliasing*. For a localization method that builds on geometry alone, this means it will not perform well in very highly structured or highly artificial environments, but our experiments show that there is sufficient information contained for the method to work for a large part of two different real-life urban environments containing scenes with varying urban characteristics such as streets with high building density, tunnels, courtyards and open spaces.

This chapter is structured as follows: In Section 5.2, the proposed method is categorized with respect to the different localization tasks important in robotics, and an overview over related work particular to the semantic localization problem is given. Section 5.3 describes in detail the steps performed to estimate the robot pose in the building outline map. The approach is experimentally evaluated on two datasets and compared to a baseline method in Section 5.4. Section 5.5 concludes the chapter.

5.2 Related Work

Localization is a field of research that, due to its crucial importance for the successful operation of autonomous robots, has received extensive attention from the scientific community. For a categorization of the different methods and approaches discussed in this overview over related work, it is helpful to distinguish a number of related robotics problems associated with localization.

- *Place recognition* is the problem of matching sensor data collected in a place to a database of features collected in a number of distinct places, and retrieving the correct one. To build this database, the robot has to have visited all eligible places before.
- *Simultaneous Localization and Mapping (SLAM)* describes the process of building a consistent metric map of an environment, which then can be used for localization. The input usually consists of a sequence of distance measurements from a laser scanner or similar sensor and odometry information, while other sensor measurements, for example about appearance, can be incorporated as well. An initial pose estimate, e.g., the result of a global localization method, is needed for starting the SLAM process.
- *Semantic Localization* is sometimes used for the process of labeling the surroundings of the robot based on sensor data (image or otherwise) with semantic categories [44, 158, 187]. Even though this is not a problem of localizing a robot on a map, its result can be used as part of such a localization method as an additional feature.
- *Global localization* or the *kidnapped robot problem*, which is the topic of this work, describes the task of localizing a robot on a map in a global frame without any prior information. For general applicability, it is desirable that the map comes from an external source, such as a topographical or cadastral map, and does not have to be built based on sensor measurements specifically for the purpose of localization. Usually, global localization should work from a single sensor measurement or a short sequence of measurements, such that it can be used as initialization procedure, for example for SLAM as described above.

Further distinctions between localization methods for robots in urban environments can be made based on the sensors that are used to provide observations about the environment. Many robots are equipped with a GPS sensor, which often provides information about the global location of the robot, which however may be noisy or temporarily unavailable due to obstructions in the environment. Other methods are based on camera images, either from monocular cameras or images with attached depth information from stereo cameras. Laser distance measurements and odometry measurements are often used as inputs in SLAM localization methods, while *visual SLAM* relies on monocular or stereo camera images.

The following gives an overview over different recent attempts at localization in urban areas, moving from appearance-based methods to ones that use semantic features of the environment. Finally, the approaches that localize on maps of building outlines, such as the one presented in this work, are surveyed.

For global localization approaches, different kinds of maps have been considered as a reference against which to determine the location of the robot. Many approaches have focused on using appearance data for localization. Common to these is a databases of sensory images annotated with location information, against which a query image is matched to retrieve the camera location. The following paragraph gives an overview over these approaches.

Aerial images have been used as prior information for localization in approaches such as the one by Leung, Clark, and Huissoon [104], which extracts line segments from street-level images and matches the geometric relationships derived from them to aerial orthoimagery using a particle filter. Another example for this group of methods is the work by Kümmerle et al. [97], which presents a SLAM system that uses aerial images as a global prior. It matches structures found in aerial images to laser data, and uses the relationships as constraints in graph-based SLAM. Agarwal, Burgard, and Spinello [1] show how to improve an approximate location estimate by matching short series of camera views with Google Street View panoramic images. This method enables global localization in an area of about 1 km radius. Majdik, Albers-Schoenberg, and Scaramuzza [118] present a similar approach for the localization of flying vehicles in the Google Street View image database, where the difference in viewpoint between the images taken from the street level and the images from flying height presents a challenge. Localization in indoor environments modeled by a database of 2.5D images is shown by Liang et al. [107], where the localization problem is divided into a place recognition step, where a template image is retrieved from the database, and a subsequent pose matching between the query and the template image. Cappelle et al. [27] compare robot observations with images sampled from a highly accurate dataset of 3D geometry and RGB appearance data to determine the robot position in cases where for example GPS is not available. A database of street-level image data augmented with 3D building models is used in the work of Baatz et al. [11] to localize a device just from monocular images, where the geometry of the query image is approximated with vanishing point detection.

A second group of approaches does not rely on appearance data, but uses sparser maps containing different sets of semantic features of urban environments for localization. For moving robots with the capability of estimating their trajectory, this knowledge can be used to localize the robot by comparing the travelled path with the paths that are feasible in the road network. Lee, Wijesoma, and Guzmán [100] integrate approximate digital maps of the road network as additional constraints with a SLAM framework based on traditional on-board sensors. The OpenStreetSLAM system [52] uses chamfer matching to compare the trajectory of the robot, which is determined with visual odometry, to street map information. Localization is achieved by tracking pose hypotheses in a particle filter and selecting those which fit best with the paths traversible on the road network. Gupta and Yilmaz [67] and Brubaker, Geiger, and Urtasun [23] follow similar approaches, but use different representations for the travelled trajectories, which allow for different matching formalisms. Irie, Sugiyama, and Tomono [80] present a localization mechanism on high-level street maps, which contain street as well as sidewalk outlines, that relies on labeling streets in images and retrieving a matching map position using a dependence maximisation approach. The method put forward by Ruchti et al. [159] also depends on the labeling of areas as street or non-street in laser scans. The semantic labeling results are then used as sensor measurement in Monte Carlo localization on a map containing the street network of an urban environment. In a different approach presented by Hentschel and Wagner [74], buildings extracted from OpenStreetMap are used as the reference map in a Monte Carlo localization framework. Vysotska and Stachniss [190] use building outlines retrieved from laser scan data to improve the localization in a SLAM framework. The matching of local surrounding buildings with a 2D map is performed using the ICP algorithm [19], which is used to provide additional constraints for a graph-based SLAM formulation. In contrast to these approaches, the localization method presented here aims at global localization, where no sequence of observations and no odometry data are available.

Building outlines in urban environments provide a salient source of geometric information, which has also been used for pose estimation with a single frame of sensor data. Many of these approaches are based on estimating the geometry of the surroundings of the robot from camera images, and then estimating the camera pose in the map by finding matches with elements from the map data. For example, Antigny, Servieres, and Renaudin [5] use distinctive objects with the same appearance and constant, known dimensions (billboards etc.) which are contained in semantically annotated maps, and localize with respect to them. This allows users to refine a rough position estimate, which is used to select the road furniture object, to an accurate pose. Cham et al. [29] perform localization in a 2D map based on a single omnidirectional ground level image, where the geometry of buildings is estimated using line and vanishing point detection, and geometric hashing is used to look up the transformation of the camera pose with respect to the map frame. The work presented by Chu, Gallagher, and Chen [36] builds on this approach, but uses a similar method to refine the position retrieved from a GPS device, i.e. localize in a smaller area around a given position. The method also relies on extracting building edges from a monocular camera image and matching the resulting geometry of a single building to buildings contained in a 2D map. Arth et al. [8] use monocular images and an initial GPS fix to localize in a 2.5D map. The ground plane of the map is given by the building outlines contained in OpenStreetMap, whereas building heights are manually annotated. Matching is done by extracting lines from the camera images and matching them to the 2.5D map; additional filtering is executed by performing a semantic segmentation of the image and matching this against OpenStreetMap information.

The work presented here is most closely related to the approach of Cham et al. [29], but it works on data from a laser scanner instead of on omnidirectional images, and uses the fact that building outlines are made up from line segments, for which the chosen matching method of chamfer matching is suited well. Furthermore, the proposed method includes visibility analysis for a more accurate matching between expected and actual observations. It also relaxes the assumption that multiple corners of a building need to be visible at the same time, which can be difficult in urban scenarios with large buildings, and particularly with occlusions. While the results presented in that work show that it is possible to reduce the number of candidate poses with the method presented there, a reliable global localisation without additional information cannot be based on it alone. Similar differences exist between the present work and the approach of Chu, Gallagher, and Chen [36], which furthermore has the different goal of refining the position estimate received from a GPS device, and not global localization. This is also a relevant difference between the work presented here and the approach of Arth et al. [8], where again the localization problem is solved for the case where monocular images of a location are available along with a location estimate from GPS or a similar sensor. Arth et al. also perform a step of rescoring pose hypotheses by comparing the input data with the content of the map that is visible from the candidate location, which is related to the formulation of the cost function taking into account visibility information put forward in this chapter. However, their method relies on performing a semantic classification of an input camera image and comparing it with a backprojection of map data including building height, which is different from the information available in the scenario envisioned here. In this work, the input data is given by a 3D laser scans, and the matching is done against building outline data alone. Evaluation shows that the method performs well in a region significantly bigger than the typical error of a GPS device, such that the method can be said to perform global localization on an urban scale, rather than GPS pose refinement using additional sensor data.

5.3 Description of the Localization Method

5.3.1 Method Overview

The global localization method described in this chapter uses 3D laser scans as sensor input data. It is matched against a 2D map of an urban environment, which contains information about building outlines as well as the street network. Data of this type can be retrieved from various sources. For the evaluation done in this chapter, semantic building outline data from OpenStreetMap is used.

The sensor data used for localization in the experiments presented here comes from a 3D laser scanner. Only distance data is used, although appearance data in the form of laser intensities is often also available. Since the localization problem as discussed here is a 2D template matching problem, the initially three-dimensional sensor data is reduced to a 2D representation by extracting vertical planar segments from the data, and reducing it further to a set of line segments representing these presumed building

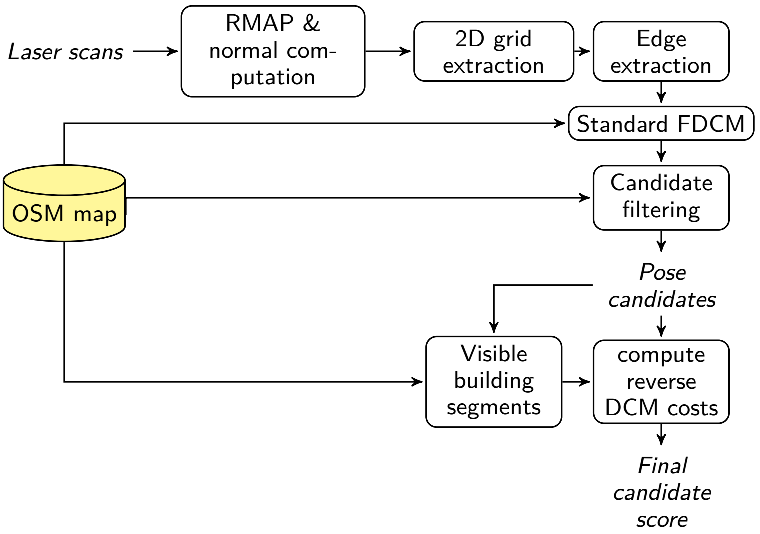
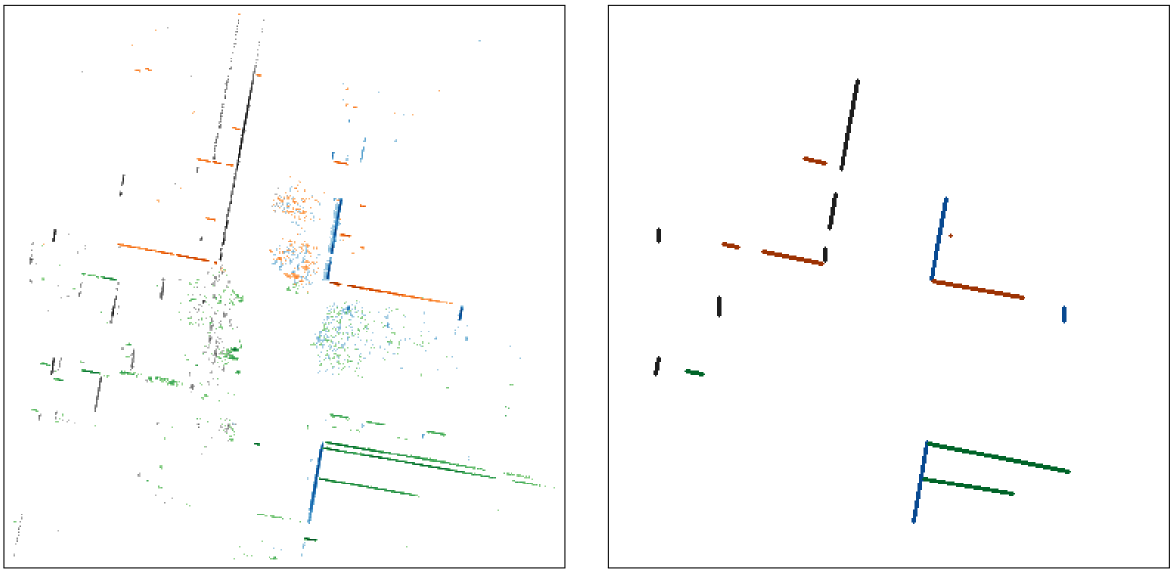


Figure 5.1: Sequence of operations performed for localization on the building outline map

outlines. Matches between the building edges from the sensor data and the 2D building outline map are computed using a fast and simple template matching procedure known from image processing. Since the template matching problem for mapping has special properties which are not taken into account by standard procedures, the results of this approach can be improved upon. Information from the building map and street network are also used to further reduce the number of candidates valid for subsequent processing. The remaining candidate poses are then further refined by a variation of the chamfer matching procedure, which takes into account visibility considerations particular to the laser data matching problem, and penalizes matches where buildings that are absent in the sensor data appear in the corresponding map section. The result of this computation is used to rank the candidates and either extract the top candidate as the estimated pose, or use a ranked set of candidates for further processing, e.g., for the initialization of a SLAM system. The sequence of processing steps is also illustrated in Figure 5.1.

5.3.2 Point Cloud Processing and Building Outline Segment Detection

Before the template matching problem of localizing the robot on the building outline map can be addressed, the input data must be reduced to a set of lines representing the presumed building outlines in the sensor's field of view. To this end, the very dense point clouds are reduced in size as a first step. For this, the rectangular cuboid approximation framework (RMAP) [85] is used to convert the point cloud into an occupancy grid consisting of cuboid cells at a lower resolution, and reduce the number of noisy observations. In this data structure, normal vectors can be efficiently computed for each occupied cuboid



(a) Histogram of cuboid cells

(b) Extracted lines

Figure 5.2: Illustration of the line segment extraction based on normal direction. The elements in the plots are colored according to the orientation of their normal vector.

cell. Since we are interested in building outlines, and the roll and pitch angles of the robot travelling on the street can be assumed to be known, vertical surfaces can be extracted from the occupancy grid by selecting cuboid cells that have a normal vector parallel to the ground plane.

These vertically oriented cuboid cells are then projected to the ground plane by setting their z coordinate to zero, and the number of cells per area unit is counted. The result is a histogram of the vertically oriented cuboid cells in the sensor range of the robot. For the goal of extracting building outlines from this representation, the normal information from the point cloud should be preserved, since only points that have a similar normal direction can belong to a common planar surface. We use this information by binning the yaw angles of the cuboid cells and creating separate histograms for each angle range. In each of these histograms, line segments are extracted using the Probabilistic Hough Transform [119]. Parallel line segments with small distances between them and collinear lines with small gaps are merged to reduce noise in the resulting set of edges. The building outline extraction process is illustrated in Figure 5.2, which shows both the histograms of oriented cuboids, and the line segments computed based on them.

5.3.3 Adapting Directional Chamfer Matching to the Localization Problem

After the building outlines have been retrieved from the laser data, retrieving the robot pose in the building map becomes a template matching problem. Chamfer Matching [14] is a well-established method for template matching, which is especially suitable to find correspondences between sets of line segments. This section describes the idea of chamfer matching and extensions of its original cost function to adapt it to the problem of matching templates for localization.

Chamfer matching is designed to find a transformation of a template edge map in the robot coordinate frame $U = \{u_i\}, i = 1, \ldots, n$ such that it optimally matches a section of a query edge map $V = \{v_i\}, i = 1, \ldots, m$ in the map coordinate frame. This transformation is a 2D Euclidean transformation $s \in SE(2)$, where $s = (\theta, t_x, t_y)$. It can be interpreted to define a pose of the robot in the coordinate frame of the map, where its location is given by (t_x, t_y), and its heading by θ. The effect of this transformation on the robot measurements can be calculated by a rotation and a subsequent translation as

$$W(\boldsymbol{x}; s) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

The optimal alignment of the query edge map with the template map is the result of the transformation which minimizes a distance function d between the two maps

$$\hat{s} = \operatorname*{argmin}_{s \in SE(2)} d\left(W(U, s), V\right). \tag{5.1}$$

In the following, let the transformed query edge set $W(U, s)$ be denoted by $\hat{U}$.

Different distance functions can be used. For standard Chamfer matching, the distance function is given by the minimal distances to a template edge point for each point in the query edge map

$$d_{CM}\left(\hat{U}, V\right) = \frac{1}{n} \sum_{\hat{u}_i \in \hat{U}} \min_{v_j \in V} |\hat{u}_i - v_j| . \tag{5.2}$$

For edge maps consisting of linear segments, it is more robust and efficient to consider the orientation for the edge, and penalize matches between edge points with different directions. This reasoning leads to the distance function of Directional Chamfer Matching (DCM) [112]

$$d_{DCM}\left(\hat{U}, V\right) = \frac{1}{n} \sum_{\hat{u}_i \in \hat{U}} \min_{v_j \in V} |\hat{u}_i - v_j| + \lambda \left|\phi(\hat{u}_i) - \phi(v_j)\right|, \tag{5.3}$$

where an edge orientation ϕ is determined for each edge point, and the distance of the orientations is determined as the minimal rotation necessary between them. In applications where it is acceptable to discretize the space of edge orientations, the optimization (5.1)

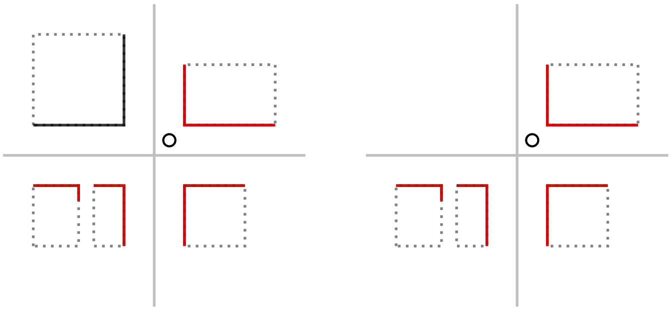


Figure 5.3: Illustration of examples for street corners with identical CM score, even though there are no sensor percepts of the building on the top left in the left-hand example. The candidate robot position is marked with a circle. Building outlines contained in the map are drawn dotted in grey, and their visible part in black. Lines extracted from a laser scan are drawn dashed in red. The street center lines are drawn in grey.

can be efficiently computed by computing a distance transform tensor, which contains the cost contributions for each query edge point. This approximation is formulated in the Fast Directional Chamfer Matching (FDCM) method [112]. In cases where the template and query edge maps can be represented as sets of linear segments, the summation of individual contributions per point can be replaced by computations only involving the end points of the line segments by computing an integral distance transform.

These cost functions are designed for the task of finding simple query edge maps in template edge maps derived from cluttered images. It is expected that, for a good match between template and query edge map transformation, each edge in the query edge map is close to a matching edge in the template edge map. All edges at larger distances are not considered for cost computation. For the application of localizing a set of building edges in a building outline map, where, due to the structured nature of typical building maps, there can be many areas that are similar to parts of what the robot sensors observe, it is desirable to also penalize matches where some part of the template that should exist in the query is not there. This is illustrated in Figure 5.3, which shows two possible transformations of a template edge map, both of which result in the same (D)CM cost values, but one of them is clearly a worse match than the other, since the building edges derived from the scan do not contain a building that would be expected to be observed.

While this information about which edges of the template map should be matched to edges in the query map is not available in a general template matching task, an estimate of the expected observation for the localization task can be generated by extracting all the lines visible in the map from a given robot pose. We denote this set of edges visible from a position (t_x, t_y) by $V_e(t_x, t_y)$. With this definition, a *forward* cost function that takes only the expected observations for a given robot position into account can be defined as

$$d_f\left(\hat{U}, V, s\right) = \frac{1}{n} \sum_{\hat{u}_i} \min_{v_j \in V_e(t_x, t_y)} |\hat{u}_i - v_j| \,. \tag{5.4}$$

Furthermore, knowledge about the expected observation also allows to define a *reverse* cost function that describes the extent to which the expected observation V_e is represented in the actual observation $\hat{U}$

$$d_r\left(\hat{U}, V, s\right) = \frac{1}{n} \sum_{v_i \in V_e(t_x, t_y)} \min_{u_j \in \hat{U}} |\hat{u}_i - v_j| \,. \tag{5.5}$$

Finally, the forward cost (5.4) and reverse cost (5.5) can be combined to form a cost function that is *symmetric* in the expected template edge map and the query map

$$d_s\left(\hat{U}, V, s\right) = \frac{1}{2}\left(d_f\left(\hat{U}, V, s\right) + d_r\left(\hat{U}, V, s\right)\right).$$

A directional extension of these latter three cost functions similar to (5.3) is possible analogously.

Computing the optimization (5.1) for these latter cost functions is significantly more complex than the cost functions (5.2) and (5.3), as the set of visible edges, which constitute the template edge map used in the computation of the cost function, depends on the translation of the considered coordinate frame transformation. This means that a computation of a distance transform tensor, which is independent of the coordinate transformation and allows the efficient computation in the FDCM approach, is not possible when the area covered by the template map is large. Even though visibility analysis can be implemented efficiently using a Binary Space Partition (BSP) [56], a brute force optimization of (5.1) with either cost function d_f, d_r, or d_s can be prohibitively computationally expensive. For this reason, in this work we adopt a heuristic approach by assuming that minimizers of these cost functions also result in low values of the simpler cost function d_{DCM}, if not the globally optimal ones. Under this assumption, the FDCM method can be used in a first pass to generate a set of pose candidates $C = \{c_i\} = \{(\theta_i, t_{x,i}, t_{y,i})\}, i = 1, \ldots, n_C$ that result in values of d_{DCM} within a given factor of its global minimum. Only for these transformations, the visible lines are computed, and the more complex cost functions are evaluated.

5.3.4 Filtering Position Candidates using OpenStreetMap information

The number of poses to consider for valid localization candidates can be restricted further with additional knowledge available from the building map. For instance, poses that lie inside buildings can be discarded. Furthermore, if, like in our case, the robot travels alongside the road, poses that are more than a given distance removed from any edge of the road network can be discarded as well. For the experiments carried out in this chapter, we consider only pose candidates that are less than 12 m removed from street elements in the OpenStreetMap network. Of the many candidate poses generated by the first FDCM optimization, many are invalid according to either their position inside a building or their

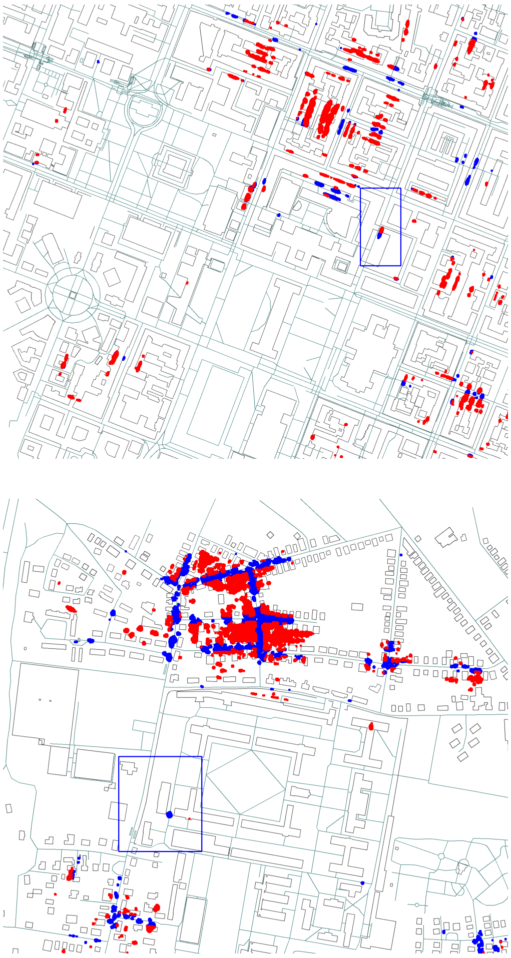


Figure 5.4: Filtering results for one scan with a large number of pose candidates from the Munich and Bremen datasets, respectively. Possible robot positions are marked with dots in red for invalid and blue for valid locations. The actual area covered by the corresponding scan is marked with a blue frame.

distance from a marked road, and thus do not have to be considered for further evaluation. This is illustrated in Figure 5.4, which shows the positions of candidate poses for an input scene that produces many matches within the area considered for localization. The figure visualizes which points are considered as valid candidates and which ones are discarded based on the criteria laid out above.

5.4 Experiments

The global localization method described above was evaluated extensively for localization accuracy. Data from two different datasets of urban environments with different characteristics were used for the evaluation. A baseline approach from literature was implemented for comparison, and the benefit of using the extended cost functions described in Section 5.3.3 over the standard DCM approach is shown.

5.4.1 Dataset

The global localization method described in Section 5.3 was evaluated on 3D laser scans from two datasets. The Munich Urban Dataset [214] contains 80 scans covering an area around the inner city campus of the Technical University of Munich. It was recorded with a Zoller & Fröhlich 5010C 3D laser range finder and also contains RGB data. Scans were manually registered by annotating salient points in overlapping scans and finding the transformation that minimizes the error between the transformed positions of these scans. The Jacobs University Bremen dataset[1] covers the campus of that university with 132 scans and was recorded with a Riegl VZ-400 laser scanner. The scans in this dataset were registered using reflective markers. Both datasets were manually aligned with the data retrieved from OpenStreetMap in a global coordinate frame.

5.4.2 Experimental Setup

As template data for the localization experiments, map data from OpenStreetMap was downloaded for a rectangular area of about 2 km width around the area covered by each dataset. This was used to generate the template building outline edge maps. The implementation of FDCM from [112] was used to obtain the candidate poses with quantization of line orientations to 12 different direction channels. The grid size for the discretization of the positions that are searched by FDCM was set to 0.5 m. All poses that yielded a cost within a factor of 1.6 of the globally optimal FDCM cost were considered as candidate poses for further processing. The three cost functions newly proposed in Section 5.3.3 as well as the original FDCM cost were used to compute a final ranking of the pose candidates.

To the best of the author's knowledge, the only method from literature that has the same goal of global localization on building outline data alone and can thus serve as a baseline is the template matching method based on geometric hashing from Cham et al. [29]. Later methods that are based on this [8, 36] use a similar matching method, but with added information in form of a GPS estimate, which is not available for the purpose of global localization. For a comparison with these prior methods, we implemented a hashing-based method similar to the one used in Cham et al. [29] to be used with scale-invariant laser data, and measurements from the urban environments represented in the experimental datasets. It relies on extracting building corners from the building outline map, which are indexed with a hash function encoding building side length and the angle between the two sides belonging to the corner.

To localize a scan using the baseline method, first, the same edge extraction process as described for the proposed method is applied. Then, corners are found in the extracted

[1] by Prashant K.C., Dorit Borrmann, Jan Elseberg, and Andreas Nüchter, retrieved from the *Robotic 3D Scan Repository* `http://kos.informatik.uni-osnabrueck.de/3Dscans/`

line segments, and all matching corners from the map are retrieved using the hash index. The transformation between the corner and the laser scanner position is computed and applied to all matching corners from the map. The resulting poses are recorded in an accumulator, such that poses where multiple corners in the map are observed from the same scanner pose receive a higher score. From this accumulator, the cells with the highest scores of matching positions are retrieved as final scanner pose estimates. The parameters for quantizing the accumulator were optimized in a coarse grid search on the experimental data to a cell size of 5 m and a pose quantization that distinguishes 6 different orientations. Localization candidates were determined for each scan in both datasets using this baseline method.

5.4.3 Experimental Results

For the evaluation of the proposed localization methods, we focus on the error in the pose of the lowest-cost pose candidates. For this analysis, a pose candidate for which both the displacement as well as the rotational error with respect to the ground truth pose are below a threshold is denoted as *accurate*. For the proposed methods, these thresholds were chosen as 4 m and 0.2 radians as maximum displacement and rotational error, respectively. For the hashing-based method, the thresholds for determining whether a candidate is accurate were chosen to reflect the size of the grid used for the accumulator, which results in a maximal distance of 5 m and an allowed rotation of $\pi/6$. Note that this pair of thresholds is less strict than the one used for the proposed method. These numbers were chosen to allow for some error in the ground truth registration with respect to the OpenStreetMap map data, and to be significantly smaller than the typical error of GPS localization in urban areas [201].

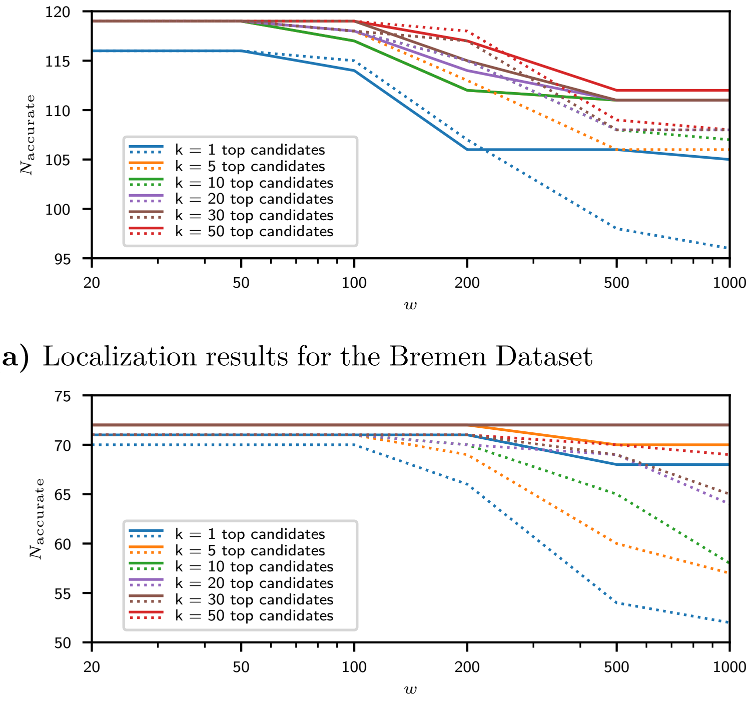
For each dataset, the evaluation is performed in the number of scans N_{accurate} for which the set of k candidates with the lowest cost within a circular area of radius w around the ground truth position contains a candidate with an accurate pose. Thus, for the strictest evaluation criterion $k = 1$, a match means that the candidate pose with the lowest cost is accurate with respect to the given thresholds; for $k = 5$ it means that there is at least one accurate candidate among the 5 candidates with least costs.

The proposed method with the chosen parametrization produced a set of pose candidates containing an accurate pose candidate for 73 of the 80 scans in the Munich dataset, and for 119 of the 132 scans in the Bremen dataset. The average number of candidates per scan for the full map used for the experiments before filtering was 1845, and 472 after filtering based on street and building data for the Munich dataset; for the Bremen dataset these numbers were 13254 and 5581, respectively. Forward and symmetric costs were computed for a maximum of 500 pose candidates with the lowest FDCM costs per scan because of their high computational demands with the current implementation; all other pose candidates were not evaluated for these costs.

This evaluation of the results of the proposed localization methods is visualized in Figure 5.5. It compares the number of accurate best-ranked scans, depending on the size of the search area, for the standard cost function d_{DCM}, and the newly proposed reverse cost function d_r. The results for the other two newly defined cost functions d_f and d_s were slightly worse than with the reverse cost function, but still outperformed the standard DCM cost d_{DCM}, so the individual results are omitted for brevity. Figure 5.6 presents a comparison of the hashing-based baseline method and the DCM method using the reverse cost function using the same analysis method.

Figure 5.7 gives an overview over the variability of scans contained in the dataset and an illustration of the nature of the results of the localization procedure. The first two rows show scans from the Munich data set where the localization provides accurate candidates, while the scans in the second two rows cover wide open areas that do not provide a sufficient number of salient features to allow the retrieval of accurate pose candidates, so the localization fails in these two cases. The fifth, sixth and seventh row show examples of successful localization from the Bremen dataset. The bottom row shows an example of a discrepancy between the observed reality and the map, since a temporary building site fence has been set up at a distance from the corresponding structure in the map. Nevertheless, candidate poses are also generated in the vicinity of the correct localization result.

The pipeline of operations is run on a largely non-optimized python implementation wrapping a modified version of the FDCM implementation of [112] for matching and cost computations. A cursory analysis of the computational properties of the processing was carried out in single-threaded computation on a Intel Quadcore i5 CPU at 3.3 GHz with 16 GB RAM. In this analysis, it can be expected that each operation can be sped up considerably with careful optimization. With the current implementation, the computation of the histogram of oriented occupied cells and the line extraction take on average 54 ms and 220 ms, respectively. Building the distance transform tensor for the first FDCM step, which needs to be done once per dataset, takes 35 s and 155 s for the maps covering the Munich and Bremen datasets, respectively. This computation time depends largely on the size of the map and the number of elements it contains, as well as the grid size chosen for the candidate extraction. Matching the observed lines to the full building outline map takes 60 s on average. For the candidate selection process, filtering all candidates takes 34 ms per scan. Further computation times are given per candidate that is evaluated; hence, computation times for the candidate selection process can be adapted by limiting the number of candidates that are being evaluated with further cost computation. Computation times for retrieving the visible lines for a candidate pose is 210 ms on average, and computing the reverse costs takes 620 ms per scan to compute the distance transform tensor, and less than 1 ms to compute the cost per candidate. Computing the forward cost takes on average 460 ms per candidate. The reason for this is that the distance transform tensor needs to be computed for each evaluation since the template for the matching changes with each candidate. In addition to the room for computational optimization, it can also be noted that the candidate evaluation can very easily be computed in parallel.



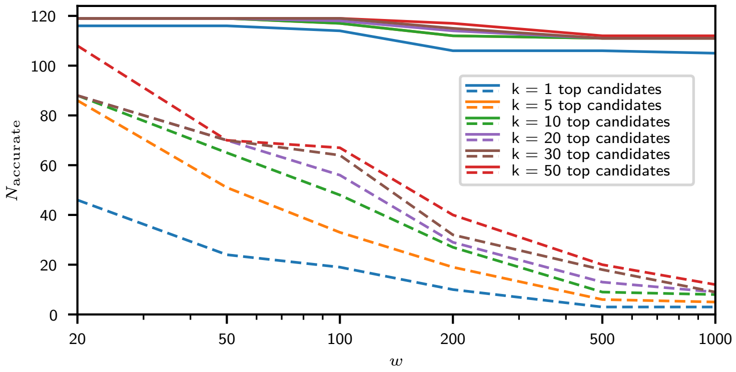
(a) Localization results for the Bremen Dataset

(b) Localization results for the Munich Dataset

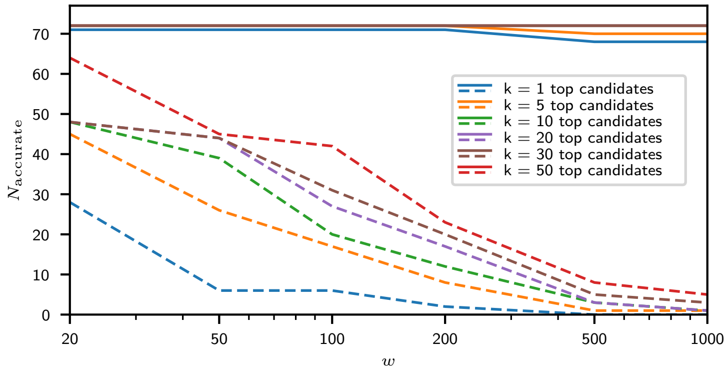
Figure 5.5: Numbers of accurately localized scenes using two different DCM cost functions for both datasets. Results obtained using the reverse cost function d_r are drawn using solid lines, and those from the standard directional chamfer matching cost function d_{DCM} are drawn dotted. The plot indicates the number of scans $N_{accurate}$ where one among the k best-rated candidates within a radius of a given size w around the ground truth position is accurate.

5.4.4 Discussion of Results

As it can be seen by inspecting the results, the proposed method generates an accurate highest-ranking pose candidate even for large search areas in a majority of cases. Taking into account a larger number of high-ranked candidates improves upon this result, which can be useful in applications such as generating an initial distribution of pose estimates for the use in a Monte Carlo localization system. The cost functions taking into account the expected observations of the robot consistently improve the result with respect to the DCM template matching method, although at the cost of increased computational cost. In particular, as it can be seen in Figure 5.5, the reverse cost function produces more stable localization results as the search area increases in comparison to the standard DCM cost function. As displayed in Figure 5.6, the proposed method also outperforms the simpler approach based on geometric hashing, which is nevertheless able to localize laser scans accurately within smaller areas. This is useful if a position estimate for example from GPS is available, but does not provide satisfactory results for larger areas.



(a) Comparison of localization results for the Bremen Dataset



(b) Comparison of localization results for the Munich Dataset

Figure 5.6: Numbers of correctly localized scenes using the hashing method for both datasets, compared to results from the proposed reverse cost function d_r. Results obtained using the reverse cost function d_r are drawn using solid lines, and those obtained using the baseline method are drawn dotted. The plot indicates the number of scans N_{correct} where one among k best-rated candidates within a radius of a given size w around the ground truth position is within 5 m of the correct position, and the pose of the corresponding accumulator bin is correct.

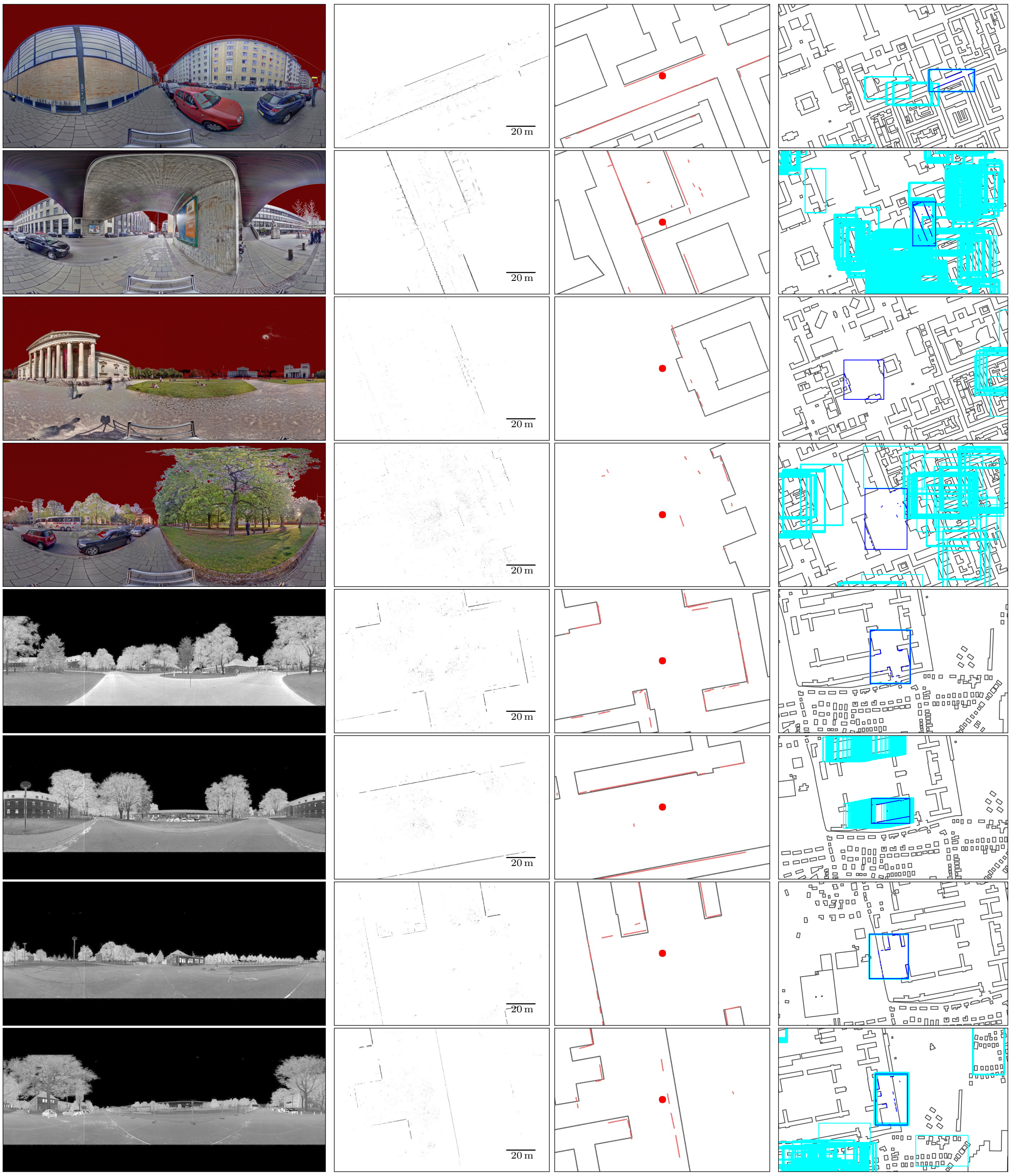


Figure 5.7: Example results of the line extraction and localization. Images from left to right: Color/intensity image; projection of building points to the ground plane; extracted building segments and OpenStreetMap building map at ground truth pose (indicated by the red dot); section of the localization result: area covered by the observations of the ground truth pose framed in dark blue, candidates for alternate poses in light blue. The top four rows of images show scans from the Munich dataset; the lower four from the Bremen dataset. For a detailed description of the example cases, please refer to the explanation in Section 5.4.3.

As illustrated in Figure 5.7, two causes for failure of the method are the lack of reliable features in areas that contain few buildings, and mismatches between the map and the environment caused by errors in the map or temporary changes in the environment. Results also show that the proposed method works successfully even in instances where no building corners are visible, for example in cases where only sides of a large buildings, the corners of which are outside the sensor range, are visible, or in situations where building corners are occluded. This is not possible with methods relying on geometric hashing or similar techniques, which rely on accurate building corner locations.

5.5 Conclusion

This chapter has shown an approach to estimate the pose of a robot in a global coordinate frame based on only a laser scan and a map containing building outlines and street network data. The evaluation has shown that this approach performs well on a large part of the data used for experimentation, which includes urban scenes with varying characteristics. It has been demonstrated that explicitly comparing the expected observation with the actual sensor data by including visibility analysis in the cost function benefits the localization accuracy.

The presented approach could be improved in a number of ways. Freely accessible databases offer much more semantically annotated data than what is used in this approach. For example, detecting the area covered by roads and paths could be used as an additional feature. Building heights, which are annotated in some maps, could be used to make the candidate selection process more concise and to generate a more accurate representation of the expected observation.

The presented system can also serve as the basis for other robotic applications, and be used in connection with other sensors. For example, the generated pose candidates with their associated costs can be used to provide the initial pose distribution for a SLAM pipeline. While the matching method has been described and tested on scale-invariant laser data, it can also be applied to building outlines extracted from camera images using computer vision methods, when scale is added as an additional degree of freedom to the first DCM search step.

6

Conclusion and Future Research Directions

This chapter provides a summary and discussion of the presented approaches and methods of each chapter from this thesis. In addition, possible directions for future research in semantic mapping for autonomous robots in urban areas are outlined.

6.1 Summary and Conclusions

Urban environments have been identified as the domain for several promising robotic applications in industry and research. Since these applications require robots to be co-located and to interact with humans who are inexperienced with using robotic technologies, high-level interaction is an important ability they need to be equipped with. In addition to sensor-level environment data, which is a basic requirement for robotic navigation, these interaction scenarios and the high-level reasoning required from the robots affirm the need for symbol-level, semantic environment knowledge that lends itself to being related to humans and to task-related reasoning. The field of semantic mapping formalises representation and reasoning techniques to gain, store and reason about such knowledge.

This thesis has explored ways of generating, using and augmenting semantic knowledge for autonomous, interactive robots in urban environments. The following paragraphs summarize the presented work and present conclusions from the findings.

Augmenting Semantic Maps with Qualitative Spatial Relations using Probabilistic Logic Chapter 3 of this thesis concerned the augmentation of a metric occupancy grid including object information with knowledge about spatial relations between objects. The presented approach builds on the SRTree environment structure, which stores pointcloud and object information in a hierarchical occupancy grid. For the envisioned application of giving and interpreting route directions, a qualitative spatial representations with the three relations *On*/*Under*, *LeftOf*/*RightOf* and *Behind*/*InFrontOf* is selected. Spatial relations between objects are determined first using a supervised learning approach. Then, higher-level knowledge about the relations between the spatial relations is entered into the model in a logical formulation, which can then be reasoned over using the Markov Logic Network (MLN) approach to Probabilistic Logic. The evaluation shows that the approach is generally able to retrieve spatial relations as they were given by users describing an urban scene, even though applying the consistency reasoning doesn't always improve the result, suggesting the conclusion that users don't always give fully consistent sets of spatial relations.

The chapter also includes the discussion of a novel inference method for the most probable variable assignment in an MLN. It is based on the conversion of the initial logical formulation of the problem to a purely algebraic pseudo-Boolean formulation. In this framework, the inference problem can be converted to an equivalent one with only pairwise interactions between variables in a quadratization step. For this problem, efficient inference algorithms such as Quadratic Pseudo-Boolean Optimization (QPBO) are applicable. Evaluation of the solutions shows that the solution quality and efficiency are comparable or superior to state-of-the-art alternatives. It is also shown that the choice of the method which is chosen to perform the quadratization step influences the solution quality. Further research is necessary to investigate how to choose the appropriate method for a given problem.

Combining Point Clouds with Semantic Data from Open-Source Maps for Scene Interpretation In Chapter 4, it is investigated how metric point clouds can be integrated with semantically annotated information from a crowdsourced map, OpenStreetMap, for the task of scene interpretation. Concretely, the task of estimating street geometry from a set of pointclouds recorded with the point of view of a robot travelling on the sidewalk is addressed. For this task, street network information from OpenStreetMap is used to identify areas of interest and to be able to reason about geometries of connected street segments with the knowledge of the full street network. The evaluation of the algorithm on 3D pointclouds from the Munich 3D urban dataset shows that including the geometric constraints based on semantic data from OpenStreetMap improves the geometry estimation with respect to the baseline, thus illustrating the benefit of using semantic data to aid the processing of sensor data. The geometric information arrived at using this method can be input back into the semantic attributes of the OpenStreetMap database.

Global Localization on a Sparse Semantic Map of Building Outlines Chapter 5 continues in the vein of exploring how semantically annotated data from OpenStreetMap can be leveraged for increased autonomy of robots in urban environments. The work presented in this chapter looks at the task of localization in environments where only semantic map data and no prior sensor data is available. In particular, global localization on a city level is described in a map containing only building outlines and road network information. The sensor-level input data that is used to localize the robot is a single 3D point cloud. The matching between sensor and map data is done by extending the chamfer matching technique for template matching and extending it with functionality to take occlusions occurring in the input data into account. Evaluation of the method on two data sets with different characteristics shows that it performs very well in comparison with competing approaches. Thus, this research describes another instance where the inclusion of semantic data from external data sources into the reasoning process benefits the performance of sensor data processing. In this case, it is remarkable that the global localization problem on a city scale can be solved based on this very sparse set of data made up of only building and road geometries.

6.2 Future Research Directions

The topics presented in this thesis have shown multiple possible avenues for further work. The following paragraphs highlight particular directions which appear promising for future research.

Using Semantic Data to Improve Sensor Data Processing This thesis has explored the combination of pointcloud data and semantic annotations from other sources for the task of scene interpretation and global localization. However, there are other robotic tasks that will likely benefit as well from taking into account semantic knowledge. In particular, one could use semantic information to improve point cloud registration, especially over longer periods of time, for example when revisiting a location after a certain amount of time. As it was demonstrated in Chapter 5 for the task of global localization, semantic knowledge about parts of the environment can provide information about the dynamic nature of an object, and thus about its reliability for purposes of scan matching or registration.

Long-Term Semantic Mapping Most existing semantic approaches model an environment for a specific instant or over a very short period of time. When this mapping information is reused, the environment is implicitly assumed to be static. However, especially urban environments are inherently dynamic, and many applications such as navigation, perception and object recognition, simulation of urban scenarios or location-based services would benefit from knowledge about the dynamics present in the environment. Multiple methods would lend themselves to storing and reasoning about information about environment dynamics in a semantic map, among them Dynamic Bayesian Networks (DBNs), spectral methods [91], (probabilistic) temporal logics, or flow analysis, e.g., for large numbers of vehicles or humans.

Increasing Information Diversity in Semantic Maps by Using Diverse Information Sources This thesis has argued that diverse information sources, such as sensor data, online databases, and interaction increase the autonomy and capability of mobile robots. In particular, this has been shown to be the case for the combination of sensor data and semantic data from OpenStreetMap in Chapter 4 and Chapter 5, and for the combination of natural language interaction and sensor data e.g. in the IURO project.

Increasing the diversity of information sources further would allow novel approaches to getting feedback or verification for the semantic data. For example, a semantically salient localization result could be verified in interaction, or information arrived at through sensor data processing can be entered into an online database for human review. This would also allow a supervised or semi-supervised creation of semantic maps, or teaching a robot what is relevant and what is not. On the other hand, the increase in available information comes also at an increased effort for planning how to use information most efficiently, which are topics which have been addressed in a different context in the fields of Partially Observable Markov Decision Processes (POMDPs) and active perception [12].

General Topics in the Realm of Semantic Mapping As it has been mentioned in Section 2.5, the term *semantic mapping* may mean different things to different people. This thesis takes a broad approach to defining it, which is derived from the heritage of combining qualitative and quantitative geometric information to semantic information to form a rich hybrid environment representation, as it was laid out in Chapter 2. In contrast, the definition used in some parts of the literature which understands semantic mapping as limited to single scene understanding and object recognition is much more narrow. As in other fields of robotics research, commonly defined challenges, datasets and evaluation criteria could help to build a clearer understanding of the topic in the robotics community, as well as focus the efforts in this area of research.

Bibliography

References

[1] P. Agarwal, W. Burgard, and L. Spinello. “Metric localization using Google Street View.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. Sept. 2015, pp. 3111–3118.

[2] James F Allen. “Maintaining knowledge about temporal intervals.” In: *Communications of the ACM* 26.11 (1983), pp. 832–843.

[3] Abhishek Anand, Hema Swetha Koppula, Thorsten Joachims, and Ashutosh Saxena. “Contextually guided semantic labeling and search for three-dimensional point clouds.” In: *The Int. Journal of Robotics Research* 32.1 (2013), pp. 19–34.

[4] Roy Anati, Davide Scaramuzza, Konstantinos G Derpanis, and Kostas Daniilidis. “Robot localization using soft object detection.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE. 2012, pp. 4992–4999.

[5] Nicolas Antigny, Myriam Servieres, and Valérie Renaudin. “Hybrid visual and inertial position and orientation estimation based on known urban 3D models.” In: *Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*. Alcala de Henares, Spain: IEEE, Oct. 2016, pp. 1–8.

[6] Francisco Antunes, Cidália C. Fonte, Maria Antonia Brovelli, Marco Minghini, Monia Molinari, and Peter Mooney. “Assessing OSM Road Positional Quality With Authoritative Data.” In: *Confèrencia Nacional de Cartografia e Geodesia*. 2015.

[7] Udi Apsel and Ronen I. Brafman. “Exploiting Uniform Assignments in First-Order MPE.” In: *Proc. Conf. Uncertainty in Artificial Intelligence*. 2012, pp. 74–83.

[8] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit. “Instant Outdoor Localization and SLAM Initialization from 2.5D Maps.” In: *IEEE Transactions on Visualization and Computer Graphics* 21.11 (Nov. 2015), pp. 1309–1318.

[9] F. Aurenhammer. “Voronoi diagrams – a survey of a fundamental geometric data structure.” In: *ACM Computing Surveys (CSUR)* 23.3 (1991), pp. 345–405.

[10] Alper Aydemir, Patric Jensfelt, and John Folkesson. “What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 4675–4682.

[11] Georges Baatz, Kevin Köser, David Chen, Radek Grzeszczuk, and Marc Pollefeys. "Leveraging 3D city models for rotation invariant place-of-interest recognition." In: *Int. J. Computer Vision* 96.3 (2012), pp. 315–334.

[12] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. "Revisiting active perception." In: *Advanced Robotics* (Feb. 2017).

[13] Philippe Balbiani, Jean-François Condotta, and Luis Farinas del Cerro. "A new tractable subclass of the rectangle algebra." In: *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*. Vol. 99. Citeseer. 1999, pp. 442–447.

[14] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Bolles, and Helen C. Wolf. "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching." In: *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI)*. Ed. by R. Reddy. Cambridge, MA, USA: William Kaufmann, Aug. 1977, pp. 659–663.

[15] Kaustubh Beedkar, Luciano Del Corro, and Rainer Gemulla. "Fully Parallel Inference in Markov Logic Networks." In: *15th GI-Symposium Database Systems for Business, Technology and Web*. Magdeburg, Germany: Bonner Köllen, 2013.

[16] Patrick Beeson, Matt MacMahon, Joseph Modayil, Aniket Murarka, Benjamin Kuipers, and Brian Stankiewicz. "Integrating Multiple Representations of Spatial Knowledge for Mapping, Navigation, and Communication." In: *Interaction Challenges for Intelligent Assistants*. 2007, pp. 1–9.

[17] Patrick Beeson, Joseph Modayil, and Benjamin Kuipers. "Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarchy." In: *Int. J. of Robotics Research* 29.4 (2010), pp. 428–459.

[18] Fernando Bernuy and Javier Ruiz del Solar. "Semantic Mapping of Large-Scale Outdoor Scenes for Autonomous Off-Road Driving." In: *Proc. IEEE Int. Conf. Computer Vision Workshop*. 2015, pp. 35–41.

[19] P.J. Besl and Neil D. McKay. "A method for registration of 3-D shapes." In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992), pp. 239–256.

[20] Jaime Boal, Álvaro Sánchez-Miralles, and Álvaro Arranz. "Topological simultaneous localization and mapping: A survey." In: *Robotica* 32 (05 Aug. 2014), pp. 803–821.

[21] E. Boros, P. L. Hammer, and G. Tavares. *Preprocessing of unconstrained quadratic binary optimization*. Tech. rep. Rutgers Center for Operations Research, 2006.

[22] E. Boros and P.L. Hammer. "Pseudo-boolean optimization." In: *Discrete Applied Mathematics* 123.1 (2002), pp. 155–225.

[23] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. "Lost! Leveraging the crowd for probabilistic visual self-localization." In: *IEEE Conf. Computer Vision and Pattern Recognition*. IEEE. 2013, pp. 3057–3064.

[24] Emma Brunskill, Thomas Kollar, and Nicholas Roy. "Topological mapping using spectral clustering and classification." In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2007, pp. 3491–3496.

[25] Pär Buschka and Alessandro Saffiotti. "Some notes on the use of hybrid maps for mobile robots." In: *Proc. Int. Conf. on Intelligent Autonomous Systems*. 2004, pp. 547–556.

[26] Ryan Calo, ed. *A Roadmap for US Robotics – From Internet to Robotics (2016 Edition)*. Robotics Virtual Organization. 2016.

[27] C. Cappelle, M. El Badaoui El Najjar, D. Pomorski, and F. Charpillet. “Localisation in urban environment using GPS and INS aided by monocular vision system and 3D geographical model.” In: *Proc. IEEE Intelligent Vehicles Symposium (IV)*. June 2007, pp. 811–816.

[28] Carl Case, Bipin Suresh, Adam Coates, and Andrew Y Ng. “Autonomous sign reading for semantic mapping.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE. 2011, pp. 3297–3303.

[29] Tat-Jen Cham, Arridhana Ciptadi, Wei-Chian Tan, Minh-Tri Pham, and Liang-Tien Chia. “Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map.” In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. San Francisco, CA, USA: IEEE Computer Society, June 2010, pp. 366–373.

[30] Bin Chen, Weihua Sun, and A. Vodacek. “Improving image-based characterization of road junctions, widths, and connectivity by leveraging OpenStreetMap vector map.” In: *Proc. IEEE Int. Geoscience and Remote Sensing Symposium*. July 2014, pp. 4958–4961.

[31] Juan Chen, Anthony G. Cohn, Dayou Liu, Shengsheng Wang, Jihong Ouyang, and Qiangyuan Yu. “A survey of qualitative spatial representations.” In: *The Knowledge Engineering Review* 30 (01 Jan. 2015), pp. 106–136.

[32] Eric Chown. “Making predictions in an uncertain world: Environmental structure and cognitive maps.” In: *Adaptive Behavior* 7.1 (1999), pp. 17–33.

[33] Elizabeth R Chrastil and William H Warren. “From cognitive maps to cognitive graphs.” In: *PloS one* 9.11 (2014), e112544.

[34] Hendrik I. Christensen, ed. *Robotics 2020 Multi-Annual Roadmap for Robotics in Europe*. SPARC – Thepartnership for Robotics in Europe. 2017.

[35] G. Christodoulou, E. G. M. Petrakis, and S. Batsakis. “Qualitative Spatial Reasoning Using Topological and Directional Information in OWL.” In: *Proc. IEEE 24th Int. Conf. Tools with Artificial Intelligence*. Vol. 1. Nov. 2012, pp. 596–602.

[36] Hang Chu, Andrew C. Gallagher, and Tsuhan Chen. “GPS Refinement and Camera Orientation Estimation from a Single Image and a 2D Map.” In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE Computer Society, 2014, pp. 171–178.

[37] Anthony G Cohn, Brandon Bennett, John Gooday, and Nicholas Mark Gotts. “Qualitative spatial representation and reasoning with the region connection calculus.” In: *GeoInformatica* 1.3 (1997), pp. 275–316.

[38] Anthony G Cohn, Sanjiang Li, Weiming Liu, and Jochen Renz. “Reasoning about topological and cardinal direction relations between 2-dimensional spatial objects.” In: *J. Artificial Intelligence Research* (2014), pp. 493–532.

[39] Anthony G. Cohn and Jochen Renz. “Qualitative Spatial Representation and Reasoning.” In: *Handbook of Knowledge Representation*. Foundations of Artificial Intelligence 3 (2008). Ed. by Vladimir Lifschitz Frank van Harmelen and Bruce Porter, pp. 551–596.

[40] J-F Condotta, Mahmoud Saade, and Gerard Ligozat. “A generic toolkit for n-ary qualitative temporal and spatial calculi.” In: *Int. Symposium Temporal Representation and Reasoning*. IEEE. June 2006, pp. 78–86.

[41] Luc De Raedt. *Logical and relational learning*. Springer, 2008.

[42] Bertrand Douillard, Dieter Fox, F Ramos, and H Durrant-Whyte. “Classification and semantic mapping of urban environments.” In: *Int. J. of Robotics Research* 30.1 (2011), pp. 5–32.

[43] R. Drouilly, P. Rives, and B. Morisset. “Semantic Representation For Navigation In Large-Scale Environments.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. Seattle, WA, May 2015.

[44] Romain Drouilly, Patrick Rives, and Benoit Morisset. “Fast hybrid relocation in large scale metric-topologic-semantic map.” In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*. 2014, pp. 1839–1845.

[45] Romain Drouilly, Patrick Rives, and Benoit Morisset. “Hybrid metric-topological-semantic mapping in dynamic environments.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, 2015, pp. 5109–5114.

[46] Ivan Dryanovski, William Morris, and Jizhong Xiao. “Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. 2010, pp. 1553–1559.

[47] Hongchao Fan, Anran Yang, and Alexander Zipf. “The intrinsic quality assessment of building footprints data on OpenStreetMap in Baden-Württemberg.” In: *Flächennutzungsmonitoring VIII Flächensparen-Ökosystemleistungen-Handlungsstrategien* (), pp. 253–260.

[48] Hongchao Fan, Alexander Zipf, Qing Fu, and Pascal Neis. “Quality assessment for building footprints data on OpenStreetMap.” In: *Int. J. Geographical Information Science* 28.4 (2014), pp. 700–719.

[49] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient graph-based image segmentation.” In: *Int. Journal of Computer Vision* 59.2 (2004), pp. 167–181.

[50] Daan Fierens, Kristian Kersting, Jesse Davis, Jian Chen, and Martin Mladenov. “Pairwise Markov Logic.” In: *Inductive Logic Programming*. Springer, 2013, pp. 58–73.

[51] Alexander Fix, Aritanan Gruber, Endre Boros, and Ramin Zabih. “A graph cut algorithm for higher-order Markov random fields.” In: *Int. Conf. Computer Vision*. IEEE. 2011, pp. 1020–1027.

[52] Georgios Floros, Benito van der Zander, and Bastian Leibe. “OpenStreetSLAM: Global vehicle localization using OpenStreetMaps.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE. 2013, pp. 1054–1059.

[53] Andrew U. Frank. “Qualitative Spatial Reasoning with Cardinal Directions.” In: *Proc. Austrian Conf. Artificial Intelligence*. Ed. by Hermann Kaindl. Berlin, Heidelberg: Springer, 1991, pp. 157–167.

[54] Christian Freksa. *Qualitative spatial reasoning*. Springer, 1991.

[55] Christian Freksa. “Using orientation information for qualitative spatial reasoning.” In: *Int. Conf. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Ed. by A. U. Frank, I. Campari, and U. Formentini. Berlin, Heidelberg: Springer, 1992, pp. 162–178.

[56] Henry Fuchs, Gregory D. Abram, and Eric D. Grant. “Near real-time shaded display of rigid objects.” In: *Proc. Conf. Computer Graphics and Interactive Technologies (SIGGRAPH)*. Ed. by Peter P. Tanner. Detroit, Michigan, USA: ACM, July 1983, pp. 65–72.

[57] Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernandez-Madrigal, and Javier Gonzalez. “Multi-hierarchical semantic maps for mobile robotics.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2005, pp. 2278–2283.

[58] Andrew C Gallagher, Dhruv Batra, and Devi Parikh. “Inference for order reduction in Markov random fields.” In: *Proc. Int. Conf. Computer Vision and Pattern Recognition*. IEEE. 2011, pp. 1857–1864.

[59] Zeno Gantner, Matthias Westphal, and Stefan Wölfl. “GQR—A fast reasoner for binary qualitative constraint calculi.” In: *AAAI Workshop on Spatial and Temporal Reasoning*. 2008.

[60] Andreas Geiger, Martin Lauer, and Raquel Urtasun. “A generative model for 3D urban scene understanding from movable platforms.” In: *IEEE Conf. Computer Vision and Pattern Recognition*. IEEE. 2011, pp. 1945–1952.

[61] Reginald G. Golledge, R Daniel Jacobson, Robert Kitchin, and Mark Blades. “Cognitive maps, spatial abilities, and human wayfinding.” In: *Geographical Review of Japan, Series B.* 73.2 (2000), pp. 93–104.

[62] Roop Goyal and Max J Egenhofer. “Cardinal directions between extended spatial objects.” In: *IEEE Trans. Knowledge and Data Engineering* (2000), pp. 291–301.

[63] Hugo Grimmett, Mathias Buerki, Lina Paz, Pedro Pinies, Paul Furgale, Ingmar Posner, and Paul Newman. “Integrating metric and semantic maps for vision-only automated parking.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE. 2015, pp. 2159–2166.

[64] G. Grisetti, C. Stachniss, and W. Burgard. “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters.” In: *Trans. Rob.* 23.1 (Feb. 2007), pp. 34–46.

[65] Giorgio Grisetti, Rainer Kummerle, Cyrill Stachniss, and Wolfram Burgard. “A tutorial on graph-based SLAM.” In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43.

[66] Hans Werner Guesgen. *Spatial reasoning based on Allen’s temporal logic*. International Computer Science Institute Berkeley, 1989.

[67] Ashish Gupta and Alper Yilmaz. “Ubiquitous real-time geo-spatial localization.” In: *Proceedings of the Eighth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, ISA@SIGSPATIAL*. (Burlingame, California, USA). Oct. 2016, pp. 1–10.

[68] Antonin Guttman. “R-trees: A dynamic index structure for spatial searching.” In: *Association for Computing Machinery* 14.2 (1984).

[69] Mordechai Haklay. “How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets.” In: *Environment and Planning B: Planning and Design* 37.4 (2010), pp. 682–703.

[70] Mordechai Haklay and Patrick Weber. “OpenStreetMap: User-Generated Street Maps.” In: *IEEE Pervasive Computing* 7.4 (2008), pp. 12–18.

[71] Stevan Harnad. “The symbol grounding problem.” In: *Physica D: Nonlinear Phenomena* 42.1-3 (1990), pp. 335–346.

[72] Nick Hawes, Marc Hanheide, Jack Hargreaves, Ben Page, Hendrik Zender, and Patric Jensfelt. “Home alone: Autonomous extension and correction of spatial representations.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* IEEE. 2011, pp. 3907–3914.

[73] Hu He and Ben Upcroft. “Nonparametric semantic segmentation for 3D street scenes.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS).* Tokyo, Japan, 2013, pp. 3697–3703.

[74] Matthias Hentschel and Bernardo Wagner. “Autonomous robot navigation based on OpenStreetMap geodata.” In: *Proc. IEEE Int. Conf. on Intelligent Transportation Systems.* IEEE. 2010, pp. 1645–1650.

[75] Stephen C Hirtle and John Jonides. “Evidence of hierarchies in cognitive maps.” In: *Memory & Cognition* 13.3 (1985), pp. 208–217.

[76] Jerry R Hobbs and Srini Narayanan. “Spatial representation and reasoning.” In: *Encyclopedia of Cognitive Science* (2002).

[77] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. “Octomap: An efficient probabilistic 3D mapping framework based on octrees.” In: *Autonomous Robots* (2013), pp. 1–18.

[78] Tuyen N. Huynh and Raymond J. Mooney. “Online Max-Margin Weight Learning for Markov Logic Networks.” In: *Proc. Int. Conf. Data Mining*, pp. 642–651.

[79] Tuyen Huynh and Raymond Mooney. “Max-margin weight learning for Markov logic networks.” In: *Machine Learning and Knowledge Discovery in Databases* (2009), pp. 564–579.

[80] Kiyoshi Irie, Masashi Sugiyama, and Masahiro Tomono. “Dependence maximization localization: a novel approach to 2D street-map-based robot localization.” In: *Advanced Robotics* 30.22 (2016), pp. 1431–1445.

[81] H. Ishikawa. “Transformation of general binary MRF minimization to the first-order case.” In: *Trans. Pattern Analysis and Machine Intelligence* 33.6 (2011), pp. 1234–1249.

[82] Fredrik Kahl and Petter Strandmark. “Generalized roof duality.” In: *Discrete Applied Mathematics* 160.16-17 (2012), pp. 2419–2434.

[83] Henry Kautz, Bart Selman, and Yueyen Jiang. “A general stochastic approach to solving problems with hard and soft constraints.” In: *The Satisfiability Problem: Theory and Applications* 17 (1997), pp. 573–586.

[84] Sheraz Khan, Athanasios Dometios, Chris Verginis, Costas Tzafestas, Dirk Wollherr, and Martin Buss. “RMAP: a rectangular cuboid approximation framework for 3D environment mapping.” English. In: *Autonomous Robots* (2014), pp. 1–17.

[85] Sheraz Khan, Athanasios Dometios, Chris Verginis, Costas Tzafestas, Dirk Wollherr, and Martin Buss. “RMAP: a rectangular cuboid approximation framework for 3D environment mapping.” In: *Autonomous Robots* (2014), pp. 1–17.

[86] Angelika Kimmig, Lilyana Mihalkova, and Lise Getoor. “Lifted graphical models: A survey.” In: *Machine Learning* 99.1 (2015), pp. 1–45.

[87] Thorsten Kluss, William E Marsh, Christoph Zetzsche, and Kerstin Schill. “Representation of impossible worlds in the cognitive map.” In: *Cognitive Processing* 16.1 (2015), pp. 271–276.

[88] Markus Knauff. “The cognitive adequacy of Allen’s interval calculus for qualitative spatial representation and reasoning.” In: *Spatial Cognition and Computation* 1.3 (1999), pp. 261–290.

[89] V. Kolmogorov and C. Rother. “Minimizing nonsubmodular functions with graph cuts-a review.” In: *Trans. Pattern Analysis and Machine Intelligence* 29.7 (2007), pp. 1274–1279.

[90] Ioannis Kostavelis and Antonios Gasteratos. “Semantic mapping for mobile robotics tasks: A survey.” In: *Robotics and Autonomous Systems* 66 (2015), pp. 86–103.

[91] Tomas Krajnik, Jaime P Fentanes, Grzegorz Cielniak, Christian Dondrup, and Tom Duckett. “Spectral analysis for long-term robotic mapping.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE. 2014, pp. 3706–3711.

[92] Bernd Krieg-Brückner, Thomas Röfer, Hans-Otto Carmesin, and Rolf Müller. “A Taxonomy of Spatial Knowledge for Navigation and its Application to the Bremen Autonomous Wheelchair.” In: *Proc. Spatial Cognition*. Ed. by Christian Freksa, Christopher Habel, and Karl F. Wender. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 373–397.

[93] Benjamin Kuipers. “An intellectual history of the spatial semantic hierarchy.” In: *Robotics and cognitive approaches to spatial mapping*. Springer, 2007, pp. 243–264.

[94] Benjamin Kuipers and Yung-Tai Byun. “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations.” In: *Robotics and Autonomous Systems* 8.1 (1991), pp. 47–63.

[95] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. “Local metrical and global topological maps in the hybrid spatial semantic hierarchy.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. Vol. 5. IEEE. 2004, pp. 4845–4851.

[96] Rainer Kümmerle, Michael Ruhnke, Bastian Steder, Cyrill Stachniss, and Wolfram Burgard. “A navigation system for robots operating in crowded urban environments.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)* (2013), pp. 3225–3232.

[97] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Alexander Kleiner, Giorgio Grisetti, and Wolfram Burgard. “Large scale graph-based SLAM using aerial images as prior information.” In: *Autonomous Robots* 30.1 (2011), pp. 25–39.

[98] Stefan Laible and Andreas Zell. “Building local terrain maps using spatio-temporal classification for semantic robot localization.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS).* Chicago, IL, USA: IEEE, 2014, pp. 4591–4597.

[99] Dagmar Lang, Susanne Friedmann, Marcel Häselich, and Dietrich Paulus. “Definition of Semantic Maps for Outdoor Robotic Tasks.” In: *Proc. IEEE Int. Conf. on Robotics and Biomimetics.* IEEE, 2014, pp. 2547–2552.

[100] Kwang Wee Lee, Sardha Wijesoma, and Javier Ibañez Guzmán. “A constrained SLAM approach to robust and accurate localisation of autonomous ground vehicles.” In: *Robotics and Autonomous Systems* 55.7 (2007), pp. 527–540.

[101] Séverin Lemaignan, Raquel Ros, E Akin Sisbot, Rachid Alami, and Michael Beetz. “Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction.” In: *International Journal of Social Robotics* 4.2 (2012), pp. 181–199.

[102] Victor Lempitsky, Carsten Rother, Stefan Roth, and Andrew Blake. “Fusion moves for markov random field optimization.” In: *Trans. Pattern Analysis and Machine Intelligence* 32.8 (2010), pp. 1392–1405.

[103] J. J. Leonard and H. F. Durrant-Whyte. “Simultaneous map building and localization for an autonomous mobile robot.” In: *Proc. IEEE/RSJ Int. Workshop Intelligence for Mechanical Systems, Intelligent Robots and Systems.* Nov. 1991, pp. 1442–1447.

[104] Keith Yu Kit Leung, Christopher Michael Clark, and Jan Paul Huissoon. “Localization in urban environments by matching ground level video images with an aerial image.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* Pasadena, California, USA: IEEE, May 2008, pp. 551–556.

[105] Kun Li and Max Q.-H. Meng. “Incorporating extrinsic object properties in robotic semantic mapping.” In: *Int. Conf. Robotics and Biomimetics, ROBIO.* Bali, Indonesia: IEEE, 2014, pp. 1392–1397.

[106] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. “Worldwide pose estimation using 3D point clouds.” In: *Proc. European Conf. Computer Vision (ECCV).* Springer, 2012, pp. 15–29.

[107] Jason Zhi Liang, Nicholas Corso, Eric Turner, and Avideh Zakhor. “Image-Based Positioning of Mobile Devices in Indoor Environments.” In: *Multimodal Location Estimation of Videos and Images.* Ed. by Jaeyoung Choi and Gerald Friedland. Cham: Springer International Publishing, 2015, pp. 85–99.

[108] Gérard Ligozat and Jochen Renz. “What is a qualitative calculus? A general framework.” In: *Proc. Pacific Rim Int. Conf. Artificial Intelligence (PRICAI).* Springer, 2004, pp. 53–64.

[109] Benson Limketkai, Lin Liao, and Dieter Fox. “Relational object maps for mobile robots.” In: *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI).* 2005, pp. 1471–1476.

[110] Ming Liu, Francis Colas, François Pomerleau, and Roland Siegwart. “A Markov semi-supervised clustering approach and its application in topological map extraction.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS).* IEEE. 2012, pp. 4743–4748.

[111] Ming Liu, Francis Colas, and Roland Siegwart. “Regional topological segmentation based on mutual information graphs.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* IEEE. 2011, pp. 3269–3274.

[112] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, and Rama Chellappa. “Fast directional chamfer matching.” In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR).* IEEE. 2010, pp. 1696–1703.

[113] Ziyuan Liu, Dong Chen, and Georg von Wichert. “Online semantic exploration of indoor maps.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* 2012.

[114] Daniel Lowd and Pedro Domingos. “Efficient weight learning for Markov logic networks.” In: *Knowledge Discovery in Databases* (2007), pp. 200–211.

[115] Dominik Lücke, Till Mossakowski, and Diedrich Wolter. “Qualitative Reasoning about Convex Relations.” In: *Proc. Spatial Cognition.* Ed. by Christian Freksa, Nora S. Newcombe, Peter Gärdenfors, and Stefan Wölfl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 426–440.

[116] Matteo Luperto, Leone D’Emilio, and Francesco Amigoni. “A generative spectral model for semantic mapping of buildings.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS).* IEEE. 2015, pp. 4451–4458.

[117] Matteo Luperto, Alberto Quattrini Li, and Francesco Amigoni. “A System for Building Semantic Maps of Indoor Environments Exploiting the Concept of Building Typology.” In: *Proc. RoboCup.* Ed. by Sven Behnke, Manuela Veloso, Arnoud Visser, and Rong Xiong. Berlin, Heidelberg: Springer, 2014, pp. 504–515.

[118] Andras Majdik, Yves Albers-Schoenberg, and Davide Scaramuzza. “MAV urban localization from Google street view data.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS).* Tokyo, Japan: IEEE, Nov. 2013, pp. 3979–3986.

[119] Jiri Matas, Charles Galambos, and Josef Kittler. “Robust Detection of Lines Using the Progressive Probabilistic Hough Transform.” In: *Computer Vision and Image Understanding* 78.1 (2000), pp. 119–137.

[120] Andrew McCallum, Kamal Nigam, and Lyle H Ungar. “Efficient clustering of high-dimensional data sets with application to reference matching.” In: *Proc. Int. Conf. Knowledge Discovery and Data Mining.* ACM. 2000, pp. 169–178.

[121] Timothy P McNamara. “Mental representations of spatial relations.” In: *Cognitive psychology* 18.1 (1986), pp. 87–121.

[122] David Meger, Per-Erik Forssén, Kevin Lai, Scott Helmer, Sancho McCann, Tristram Southey, Matthew Baumann, James J Little, and David G Lowe. “Curious george: An attentive semantic robot.” In: *Robotics and Autonomous Systems* 56.6 (2008), pp. 503–511.

[123] Nikos Mitsou, Roderick de Nijs, David Lenz, Johannes Frimberger, Dirk Wollherr, Kolja Kühnlenz, and Costas S. Tzafestas. “Online Semantic Mapping of Urban Environments.” In: *Proc. Spatial Cognition.* Ed. by Cyrill Stachniss, Kerstin Schill, and David H. Uttal. Vol. 7463. Lecture Notes in Computer Science. Springer, 2012, pp. 54–73.

[124] Happy Mittal, Prasoon Goyal, Vibhav G Gogate, and Parag Singla. “New Rules for Domain Independent Lifted MAP Inference.” In: *Proc. Advances in Neural Information Processing Systems.* 2014, pp. 649–657.

[125] Reinhard Moratz and Marco Ragni. “Qualitative spatial reasoning about relative point position.” In: *J. Visual Languages & Computing* 19.1 (2008), pp. 75–98.

[126] Oscar Martinez Mozos, Rudolph Triebel, Patric Jensfelt, Axel Rottmann, and Wolfram Burgard. “Supervised semantic labeling of places using information extracted from sensor data.” In: *Robotics and Autonomous Systems* 55.5 (2007), pp. 391–402.

[127] L. Murphy and G. Sibley. “Incremental unsupervised topological place discovery.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* May 2014, pp. 1312–1318.

[128] Alexandros Nanopoulos, Apostolos N Papadopoulos, and Yannis Theodoridis. “R-trees: Theory and Applications.” In: *Springer* (2006).

[129] Mathias Niepert and Guy Van den Broeck. “Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference.” In: *Proc. National Conf. Artificial Intelligence (AAAI).* Québec City, Québec, Canada., 2014, pp. 2467–2475.

[130] Carlos Nieto-Granda, John G Rogers, Alexander JB Trevor, and Henrik I Christensen. “Semantic map partitioning in indoor environments using regional analysis.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS).* IEEE. 2010, pp. 1451–1456.

[131] Roderick de Nijs, Sebastian Ramos, Gemma Roig, Xavier Boix, LV Gool, and Kolja Kuhnlenz. “On-line semantic perception using uncertainty.” In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conference on.* IEEE. 2012, pp. 4185–4191.

[132] Feng Niu, Christopher Ré, AnHai Doan, and Jude Shavlik. “Tuffy: Scaling up statistical inference in markov logic networks using an RDBMS.” In: *Proc. VLDB Endowment* 4.6 (2011), pp. 373–384.

[133] Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. “RockIt: Exploiting Parallelism and Symmetry for MAP Inference in Statistical Relational Models.” In: *AAAI Workshop: Statistical Relational Artificial Intelligence.* 2013.

[134] Andreas Nüchter and Joachim Hertzberg. “Towards semantic maps for mobile robots.” In: *Robotics and Autonomous Systems* 56.11 (2008), pp. 915–926.

[135] Dejan Pangercic, Benjamin Pitzer, Moritz Tenorth, and Michael Beetz. “Semantic object maps for robotic housework-representation, acquisition and use.” In: *iros.* IEEE. 2012, pp. 4644–4651.

[136] Kaustubh Pathak, Andreas Birk, Narunas Vaskevicius, and Jann Poppinga. “Fast Registration Based on Noisy Planes With Unknown Correspondences for 3-D Mapping.” In: *Trans. Robotics* 26.3 (2010), pp. 424–441.

[137] Rodrigo Polastro, Fabiano Corrêa, Fabio Cozman, and Jun Okamoto Jr. “Semantic mapping with a probabilistic description logic.” In: *Advances in Artificial Intelligence–SBIA 2010.* Springer, 2011, pp. 62–71.

[138] David Poole. “First-order Probabilistic Inference.” In: *Int. Joint Conf. Artificial Intelligence.* Ed. by Georg Gottlob and Toby Walsh. Morgan Kaufmann, 2003, pp. 985–991.

[139] Ingmar Posner, Mark Cummins, and Paul M. Newman. “A generative framework for fast urban labeling using spatial and temporal context.” In: *Autonomous Robots* 26.2-3 (2009), pp. 153–170.

[140] Ingmar Posner, Derik Schröter, and Paul M. Newman. “Online generation of scene descriptions in urban environments.” In: *Robotics and Autonomous Systems* 56.11 (2008), pp. 901–914.

[141] A Pronobis and P Jensfelt. “Large-scale Mapping and Reasoning with Heterogeneous Modalities.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* 2012, p. 28.

[142] Andrzej Pronobis and Patric Jensfelt. “Large-scale semantic mapping and reasoning with heterogeneous modalities.” In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on.* IEEE. 2012, pp. 3515–3522.

[143] Andrzej Pronobis and Patric Jensfelt. “Multi-modal semantic mapping.” In: *RSS Workshop on Grounding Human-Robot Dialog for Spatial Tasks.* Los Angeles, CA, USA, 2011.

[144] Baoxing Qin, Zhuang Jie Chong, Tirthankar Bandyopadhyay, Marcelo H. Ang, Emilio Frazzoli, and Daniela Rus. “Learning pedestrian activities for semantic mapping.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)* (2014), pp. 6062–6069.

[145] David A Randell, Zhan Cui, and Anthony G Cohn. “A spatial logic based on regions and connection.” In: *Proc. Int. Conf. Knowledge Representation and Reasoning.* 1992, pp. 165–176.

[146] A. Ranganathan and F. Dellaert. “Bayesian surprise and landmark detection.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* May 2009, pp. 2017–2023.

[147] Ananth Ranganathan and Frank Dellaert. “Online probabilistic topological mapping.” In: *Int. J. of Robotics Research* 30.6 (2011), pp. 755–771.

[148] Ananth Ranganathan and Frank Dellaert. “Semantic Modeling of Places using Objects.” In: *Proc. Robotics: Science and Systems (RSS).* Ed. by Wolfram Burgard, Oliver Brock, and Cyrill Stachniss. Atlanta, Georgia, USA: The MIT Press, 2007.

[149] Emilio Remolina and Benjamin Kuipers. “Towards a general theory of topological maps.” In: *Artificial Intelligence* 152.1 (2004), pp. 47–104.

[150] Jochen Renz and Debasis Mitra. “Qualitative direction calculi with arbitrary granularity.” In: *Proc. Pacific Rim Int. Conf. Artificial Intelligence (PRICAI).* Vol. 3157. 2004, pp. 65–74.

[151] Jochen Renz and Bernhard Nebel. “Spatial reasoning with topological information.” In: *Proc. Spatial Cognition.* Springer. 1998, pp. 351–371.

[152] Luis Riazuelo, Moritz Tenorth, Daniel Di Marco, Marta Salas, Dorian Gálvez-López, Lorenz Mösenlechner, Lars Kunze, Michael Beetz, Juan D Tardos, Luis Montano, et al. “RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach.” In: *IEEE Trans. Automation Science and Engineering* 12.2 (2015), pp. 432–443.

[153] M. Richardson and P. Domingos. “Markov logic networks.” In: *Machine learning* 62.1 (2006), pp. 107–136.

[154] Sebastian Riedel. “Cutting Plane MAP Inference for Markov Logic.” In: *Int. Workshop Statistical Relational Learning*. 2009.

[155] A. Rituerto, A.C. Murillo, and J.J. Guerrero. “Semantic labeling for indoor topological mapping using a wearable catadioptric system.” In: *Robotics and Autonomous Systems* 62.5 (2014). Special Issue Semantic Perception, Mapping and Exploration, pp. 685–695.

[156] Gerd Ronning. “Maximum likelihood estimation of Dirichlet distributions.” In: *Journal of statistical computation and simulation* 32.4 (1989), pp. 215–221.

[157] IG Rosenberg. “Reduction of bivalent maximization to the quadratic case.” In: *Cahiers du Centre d’etudes de Recherche Operationnelle* 17 (1975), pp. 71–74.

[158] Fernando Rubio, Jesus Martínez-Gómez, M. Julia Flores, and José Miguel Puerta. “Comparison between Bayesian network classifiers and SVMs for semantic localization.” In: *Expert Syst. Appl.* 64 (2016), pp. 434–443.

[159] Philipp Ruchti, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. “Localization on OpenStreetMap Data using a 3D Laser Scanner.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. Seattle, Washington, USA, May 2015.

[160] Julian Ryde and Jason J Corso. “Fast Voxel Maps with Counting Bloom Filters.” In: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2012.

[161] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, Juha Ala-Luhtala, and Achim J Lilienthal. “Normal Distributions Transform Occupancy Maps: Application to Large-Scale Online 3D Mapping.” In: *Proc. IEEE Int. Conf. on Robotics and Automation*. 2013.

[162] Somdeb Sarkhel, Deepak Venugopal, Tuan Anh Pham, Parag Singla, and Vibhav Gogate. “Scalable Training of Markov Logic Networks Using Approximate Counting.” In: *Proc. Conf. Artificial Intelligence*. Phoenix, Arizona, USA: AIII Press, 2016, pp. 1067–1073.

[163] Somdeb Sarkhel, Deepak Venugopal, Parag Singla, and Vibhav Gogate. “Lifted MAP Inference for Markov Logic Networks.” In: *Proc. Int. Conf. Artificial Intelligence and Statistics*. 2014, pp. 859–867.

[164] Bertrand Le Saux and Martial Sanfourche. “Rapid semantic mapping: Learn environment classifiers on the fly.” In: *Int. Conf. Intelligent Robots and Systems*. Tokyo, Japan: IEEE, 2013, pp. 3725–3730.

[165] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. “Urban 3D semantic modelling using stereo vision.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE. 2013, pp. 580–585.

[166] Sunando Sengupta, Paul Sturgess, Philip HS Torr, et al. “Automatic dense visual semantic mapping from street-level imagery.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 857–862.

[167] Jude W. Shavlik and Sriraam Natarajan. “Speeding Up Inference in Markov Logic Networks by Preprocessing to Reduce the Size of the Resulting Grounded Network.” In: *Proc. Int. Joint Conf. Artificial Intelligence*. 2009, pp. 1951–1956.

[168] Alexander W. Siegel and Sheldon H. White. “The Development of Spatial Representations of Large-Scale Environments.” In: *Advances in Child Development and Behavior* 10 (1975). Ed. by Hayne W. Reese, pp. 9–55.

[169] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. “Indoor Segmentation and Support Inference from RGBD Images.” In: *Proc. European Conf. Computer Vision*. 2012.

[170] G. Singh and J. Košecká. “Acquiring semantics induced topology in urban environments.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. May 2012, pp. 3509–3514.

[171] P. Singla and P. Domingos. “Lifted first-order belief propagation.” In: *Proc. National Conf. Artificial Intelligence*. Vol. 2. 2008, pp. 1094–1099.

[172] Parag Singla and Pedro Domingos. “Entity resolution with markov logic.” In: *Proc. Int. Conf. Data Mining*. IEEE. 2006, pp. 572–582.

[173] K. Sjöö, A. Pronobis, and P. Jensfelt. “Functional topological relations for qualitative spatial representation.” In: *Proc. IEEE Int. Conf. on Advanced Robotics*. IEEE. 2011, pp. 130–136.

[174] Randall C. Smith and Peter Cheeseman. “On the Representation and Estimation of Spatial Uncertainty.” In: *Int. J. of Robotics Research* 5.4 (1986), pp. 56–68.

[175] Markus Stocker and Evren Sirin. “PelletSpatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine.” In: *Proc. 6th Int. Conf. OWL: Experiences and Directions (OWLED)*. Chantilly, VA, 2009, pp. 39–48.

[176] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford. “Place categorization and semantic mapping on a mobile robot.” In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. May 2016, pp. 5729–5736.

[177] Adriana Tapus and Roland Siegwart. “Incremental robot mapping with fingerprints of places.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2005, pp. 2429–2434.

[178] Moritz Tenorth and Michael Beetz. “KnowRob: A Knowledge Processing Infrastructure for Cognition-enabled Robots.” In: *Int. J. of Robotics Research* 32.5 (Apr. 2013), pp. 566–590.

[179] Moritz Tenorth, Lars Kunze, Dominik Jain, and Michael Beetz. “KNOWROB-MAP – Knowledge-Linked Semantic Object Maps.” In: *Proc. IEEE/RAS Int. Conf. Humanoid Robots*. IEEE. 2010, pp. 430–435.

[180] Sebastian Thrun, Jens-Steffen Gutmann, Dieter Fox, Wolfram Burgard, Benjamin Kuipers, et al. “Integrating topological and metric maps for mobile robot navigation: A statistical approach.” In: *Proc. National Conf. Artificial Intelligence (AAAI)*. 1998, pp. 989–995.

[181] Edward C Tolman. “Cognitive maps in rats and men.” In: *Psychological review* 55.4 (1948), p. 189.

[182] Alexander JB Trevor, John G Rogers, Carlos Nieto-Granda, and Henrik I Christensen. “Feature-based mapping with grounded landmark and place labels.” In: *RSS Workshop on Grounding Human-Robot Dialog for Spatial Tasks* (2011).

[183] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. “Multi-level surface maps for outdoor terrain mapping and loop closing.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE. 2006, pp. 2276–2282.

[184] Barbara Tversky. “Cognitive maps, cognitive collages, and spatial mental models.” In: *Spatial Information Theory: A Theoretical Basis for GIS*. Ed. by A. U. Frank and I. Campari. Springer, 1993, pp. 14–24.

[185] Barbara Tversky. “Levels and structure of spatial knowledge.” In: *Cognitive mapping: Past, present and future* (2000). Ed. by R. Kitchin and S. M. Freundschuh.

[186] Shrihari Vasudevan, Stefan Gächter, Viet Nguyen, and Roland Siegwart. “Cognitive maps for mobile robots—an object-based approach.” In: *Robotics and Autonomous Systems* 55.5 (2007), pp. 359–371.

[187] Shrihari Vasudevan and Roland Siegwart. “Bayesian space conceptualization and place classification for semantic maps in mobile robotics.” In: *Robotics and Autonomous Systems* 56.6 (2008), pp. 522–537.

[188] Laure Vieu. “Spatial Representation and Reasoning in Artificial Intelligence.” In: *Spatial and Temporal Reasoning*. Ed. by Oliviero Stock. Dordrecht: Springer Netherlands, 1997, pp. 5–41.

[189] Pooja Viswanathan, David Meger, Tristram Southey, James J Little, and Alan K Mackworth. “Automated spatial-semantic modeling with applications to place labeling and informed search.” In: *Proc. Canadian Conf. Computer and Robot Vision*. IEEE. 2009, pp. 284–291.

[190] Olga Vysotska and Cyrill Stachniss. “Exploiting building information from publicly available maps in graph-based SLAM.” In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*. (Daejeon, South Korea). 2016, pp. 4511–4516.

[191] Jan Oliver Wallgrün. “Qualitative spatial reasoning for topological map learning.” In: *Spatial Cognition & Computation* 10.4 (2010), pp. 207–246.

[192] Jan Oliver Wallgrün, Lutz Frommberger, Diedrich Wolter, Frank Dylla, and Christian Freksa. “Qualitative spatial representation and reasoning in the SparQ-toolbox.” In: *Proc. Spatial Cognition*. Springer, 2006, pp. 39–58.

[193] Matthew R Walter, Sachithra Hemachandra, Bianca Homberg, Stefanie Tellex, and Seth Teller. “A framework for learning semantic maps from grounded natural language descriptions.” In: *Int. J. of Robotics Research* 33.9 (2014), pp. 1167–1190.

[194] S. Werner, B. Krieg-Brückner, and T. Herrmann. “Modelling navigational knowledge by route graphs.” In: *Spatial Cognition II* (2000), pp. 295–316.

[195] Steffen Werner, Bernd Krieg-Brückner, Hanspeter A Mallot, Karin Schweizer, and Christian Freksa. “Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation.” In: *Informatik ’97 – Informatik als Innovationsmotor*. Springer, 1997, pp. 41–50.

[196] Matthias Westphal and Stefan Wölfl. “Confirming the QSR Promise.” In: *AAAI Spring Symposium: Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*. 2009.

[197] Denis F Wolf and Gaurav S Sukhatme. “Semantic mapping using mobile robots.” In: *Trans. Robotics* 24.2 (2008), pp. 245–258.

[198] D. Wolter and J.H. Lee. “Qualitative reasoning with directional relations.” In: *Artificial Intelligence* 174.18 (2010), pp. 1498–1507.

[199] Diedrich Wolter and Jan Oliver Wallgrün. “Qualitative spatial reasoning for applications: New challenges and the SparQ toolbox.” In: *Qualitative Spatio-Temporal Representation and Reasoning: Trends and Future Directions* (2012), pp. 336–362.

[200] Jiangye Yuan and Anil M Cheriyadat. “Road segmentation in aerial images by exploiting road vector data.” In: *Proc. IEEE Int. Conf. on Computing for Geospatial Research and Application.* IEEE. 2013, pp. 16–23.

[201] Paul A Zandbergen and Sean J Barbeau. “Positional accuracy of assisted GPS data from high-sensitivity GPS-enabled mobile phones.” In: *Journal of Navigation* 64.03 (2011), pp. 381–399.

[202] Hendrik Zender, O Martínez Mozos, Patric Jensfelt, G-JM Kruijff, and Wolfram Burgard. “Conceptual spatial representations for indoor mobile robots.” In: *Robotics and Autonomous Systems* 56.6 (2008), pp. 493–502.

Own Publications

[203] Martin Buss, Daniel Carton, Barbara Gonsior, Kolja Kühnlenz, Christian Landsiedel, Nikos Mitsou, Roderick de Nijs, Jakub Zlotowski, Stefan Sosnowski, Ewald Strasser, Manfred Tscheligi, Astrid Weiss, and Dirk Wollherr. “Towards Proactive Human-Robot Interaction in Human Environments.” In: *Proc. Int. Conf. Cognitive Infocommunications.* 2011.

[204] Martin Buss, Daniel Carton, Sheraz Khan, Barbara Kühnlenz, Kolja Kühnlenz, Roderick de Nijs, Annemarie Turnwald, and Dirk Wollherr. “IURO – Soziale Mensch-Roboter-Interaktion in den Straßen von München [IURO – Social Human-Robot Interaction in the streets of Munich].” German. In: *at – Automatisierungstechnik* (2015). German.

[205] Roderick de Nijs, Christian Landsiedel, Dirk Wollherr, and Martin Buss. “Quadratization and Roof Duality of Markov Logic Networks.” In: *J. Artificial Intelligence Research* 55 (2016), pp. 685–714.

[206] Barbara Gonsior, Christian Landsiedel, Antonia Glaser, Dirk Wollherr, and Martin Buss. “Dialog strategies for handling miscommunication in task-related HRI.” In: *Proc. IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN).* Ed. by Henrik I. Christensen. IEEE, 2011, pp. 369–375.

[207] Barbara Gonsior, Christian Landsiedel, Nicole Mirnig, Stefan Sosnowski, Ewald Strasser, Jakub Zlotowski, Martin Buss, Kolja Kühnlenz, Manfred Tscheligi, Astrid Weiss, and Dirk Wollherr. “Impacts of Multimodal Feedback on Efficiency of Proactive Information Retrieval from Task-Related HRI.” In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 16.2 (2012), pp. 313–326.

[208] Christian Landsiedel, Roderick de Nijs, Kolja Kühnlenz, and Dirk Wollherr. “Route description interpretation on automatically labeled robot maps.” In: *IEEE Int. Conf. on Robotics and Automation.* 2013.

[209] Christian Landsiedel, Verena Rieser, Matt Walter, and Dirk Wollherr. “A Review of Spatial Reasoning and Interaction for Real-World Robotics.” In: *Advanced Robotics* 31.5 (2017), pp. 222–242.

[210] Christian Landsiedel and Dirk Wollherr. “Global localization of 3D point clouds in building outline maps of urban outdoor environments.” In: *International Journal of Intelligent Robotics and Applications* 1.4 (Dec. 2017), pp. 429–441.

[211] Christian Landsiedel and Dirk Wollherr. “Road Geometry Estimation for Urban Semantic Maps using Open Data.” In: *Advanced Robotics* 31.5 (2017), pp. 282–290.

[212] Nicole Mirnig, Barbara Gonsior, Stefan Sosnowski, Christian Landsiedel, Dirk Wollherr, Astrid Weiss, and Manfred Tscheligi. “Feedback guidelines for multimodal human-robot interaction: How should a robot give feedback when asking for directions?” In: *Proc. IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN).* IEEE, 2012, pp. 533–538.

[213] Michael Van den Bergh, Daniel Carton, Roderick de Nijs, Nikos Mitsou, Christian Landsiedel, Kolja Kühnlenz, Dirk Wollherr, Luc Van Gool, and Martin Buss. “Real-time 3D hand gesture interaction with a robot for understanding directions from humans.” In: *Proc. IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN).* 2011, pp. 357–362.

[214] Dirk Wollherr, Sheraz Khan, Christian Landsiedel, and Martin Buss. “The Interactive Urban Robot IURO: Towards Robot Action in Human Environments.” In: *Experimental Robotics: The 14th International Symposium on Experimental Robotics.* Ed. by Ani M. Hsieh, Oussama Khatib, and Vijay Kumar. Cham, Switzerland: Springer International Publishing, 2016, pp. 277–291.