Learning Probabilistic Models for Mobile Robot Navigation

Henrik Kretzschmar

Technische Fakultät
Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Dissertation zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften

Betreuer: Prof. Dr. Wolfram Burgard

Learning Probabilistic Models for Mobile Robot Navigation

Henrik Kretzschmar

Dissertation zur Erlangung des akademischen Grades Doktor der Naturwissenschaften
Technische Fakultät, Albert-Ludwigs-Universität Freiburg

Dekan	Prof. Dr. Yiannos Manoli
Erstgutachter	Prof. Dr. Wolfram Burgard
Zweitgutachter	Prof. Dr. Martin Riedmiller
Tag der Disputation	6. Oktober 2014

Zusammenfassung

Eine neue Generation autonomer Roboter ist im Begriff, das Leben vieler Menschen maßgeblich zu verändern. So zeichnet es sich mittlerweile ab, dass mobile Roboter in naher Zukunft viele alltägliche Aufgaben übernehmen werden wie etwa die Reinigung von Wohnungen und Büroumgebungen. Des Weiteren werden autonom fahrende Autos Fahrdienste anbieten und Transportroboter dringende Pakete, beispielsweise mit wichtigen Medikamenten, zu jeder Zeit schnell, zuverlässig und dennoch kostengünstig selbst in abgelegene Gegenden bringen. Um diese und andere Aufgaben erfolgreich ausführen zu können, müssen die Roboter jedoch in der Lage sein, über längere Zeiträume hinweg gänzlich autonom zu agieren. Dabei spielt die Fähigkeit, autonom zu navigieren, eine entscheidende Rolle.

Roboter sind im Allgemeinen auf detaillierte Modelle angewiesen. So ist ein Modell der Umgebung in aller Regel eine Grundvoraussetzung für die erfolgreiche Navigation eines jeden autonom agierenden mobilen Roboters. In Umgebungen, in denen sich zudem Menschen aufhalten, ist sein Verhalten diesen gegenüber ein kritischer Faktor für seine Akzeptanz. Idealerweise passt der Roboter sein Verhalten an die Menschen in seiner Umgebung an, um eine angenehme Interaktion mit ihnen zu ermöglichen und sie nicht unnötig zu stören. Das gewünschte Verhalten hängt jedoch stark vom konkreten Anwendungsbereich des Roboters ab. In den meisten Fällen ist es nicht praktikabel, die Modelle, die für den Einsatz eines Roboters in einer bestimmten Umgebung nötig sind, manuell zu erstellen. Flexibel einsetzbare Roboter müssen daher in der Lage sein, die Modelle an Ort und Stelle mittels ihrer eigenen Sensoren zu generieren.

Mit dieser Arbeit stellen wir neue Techniken vor, mit deren Hilfe ein Roboter selbständig Modelle erstellen kann, die es ihm dann ermöglichen, in von Menschen bevölkerten Gebieten unter Beachtung sozialer Normen autonom zu navigieren. Dies stellt ein überaus komplexes Problem dar. So sind die Sensormessungen des Roboters in aller Regel mit einer nicht zu vernachlässigenden Unsicherheit behaftet. Beobachtet der Roboter mit seinen Sensoren Menschen, um sich an ihr Verhalten anzupassen, so muss er darüber hinaus berücksichtigen, dass dieses natürlichen Schwankungen unterworfen ist.

Zuerst beschäftigen wir uns mit der Kartierung der Umgebung des Roboters. Hier stellen wir ein Verfahren vor, welches schätzt, wieviel eine Sensormessung zum Umgebungsmodell beiträgt. Dadurch ist der Roboter in der Lage, überwiegend redundante Messungen aus seinem Speicher zu löschen, um so Platz für neue Messungen zu schaffen, was in vielen Situationen deutlich längere Einsätze des Roboters ermöglicht. Um ein konsistentes Umgebungsmodell zu erstellen, muss der Roboter in der Lage sein, während des Kartiervorgangs jene Orte, die er bereits besucht hat, wiederzuerkennen. Besonders in Umgebungen, in denen sich bestimmte Bereiche ähneln, kann es dabei jedoch zu

Verwechslungen kommen, welche in der Folge meistens zu einer für die Navigation unbrauchbaren Karte führen. Ist der Roboter während des Kartierens in der Lage, künstliche Landmarken in seiner Umgebung auszubringen, so kann er diese später nutzen, um die Orte, die mit einer solchen Landmarke versehen sind, zweifelsfrei wiederzuerkennen. Wir untersuchen, wo der Roboter während des Kartierens die Landmarken ausbringen sollte. Hierzu stellen wir ein Verfahren vor, das eine Strategie zur effizienten Ausbringung der künstlichen Landmarken unter Zuhilfenahme von verstärkendem Lernen berechnet. Die resultierende Strategie ist bestrebt, die Landmarken so in der Umgebung zu verteilen, dass sie sich im weiteren Verlauf des Kartierens als möglichst nützlich erweisen.

Roboter, die in von Menschen belebten Umgebungen autonom navigieren, sollten in der Lage sein, mit den Menschen auf natürliche Weise zu kommunizieren, beispielsweise um ihnen zu beschreiben, wo sie beabsichtigen, als Nächstes hinzufahren. In diesem Zusammenhang untersuchen wir, wie sich Menschen gegenseitig Wegbeschreibungen geben. Mit den gewonnenen Erkenntnissen entwickeln wir ein neuartiges Verfahren, das automatisch Wegbeschreibungen generiert. Das Besondere an unserem Ansatz ist hierbei, dass er aus vorgegebenen Wegbeschreibungen, die von einer Gruppe von Menschen stammen, ein Modell des Stils dieser Beschreibungen ableitet. Dieses Modell ermöglicht es nun einem Roboter, auf die gleiche natürliche Art und Weise Wegbeschreibungen zu beliebigen Orten, die in seiner Umgebungskarte verzeichnet sind, zu geben.

Damit autonom navigierende Roboter von Menschen tatsächlich als hilfreich und nicht als störend wahrgenommen werden, müssen sie soziale Normen einhalten. So dürfen die Roboter auf der einen Seite die Menschen nicht unnötig behindern. Auf der anderen Seite aber sollten sie auf dem Weg zu ihrem Ziel verzichtbare Umwege infolge überzogener Vorsicht vermeiden. Eine Möglichkeit, ein Modell des gewünschten Navigationsverhaltens des Roboters zu erhalten, ist es, die Menschen in seiner Umgebung zu beobachten und zu lernen, wie sie sich verhalten. Hierbei ist es besonders wichtig, zu erfassen, wie sich Menschen bewegen, wenn sie in die Nähe anderer Menschen oder eines fahrenden Roboters kommen. In diesem Zusammenhang stellen wir daher zuerst ein Verfahren vor, mit dem die Bewegungen von Menschen aufgezeichnet werden können. Unser Ansatz verwendet dazu Messungen von Inertialsensoren, die die Menschen am Körper tragen. Da die resultierenden Positionsschätzungen jedoch zu ungenau sind, lässt unser Verfahren Messungen der Positionen der Menschen, die von einem mobilen Roboter gemacht werden, in die Schätzung einfließen. Auf diese Weise lassen sich überaus genaue Aufnahmen der Bewegungen von Menschen sogar in großen Gebieten machen. Der Roboter kann diese Daten nun verwenden, um ein Modell des den Bewegungen zugrundeliegenden Navigationsverhaltens der Menschen zu ermitteln. Diese Aufgabe ist besonders anspruchsvoll, wenn nur wenige Aufnahmen vorliegen. Wir erarbeiten einen Ansatz, der selbst unter solch schwierigen Umständen in der Lage ist, zuverlässig ein Modell des beobachteten menschlichen Verhaltens zu berechnen. Mit Hilfe eines solchen Modells kann ein mobiler Roboter dann während der Navigation vorhersagen, wie sich ein Mensch in der aktuellen Situation verhalten würde. Dies ermöglicht ihm, soziale Normen, die er vorher beobachtet hat, einzuhalten, zum Beispiel indem er das Verhalten der Menschen imitiert.

Alle in dieser Arbeit vorgestellten Verfahren wurden in umfangreichen Experimenten sowohl in Simulation als auch mit echten Robotern getestet. Die Ergebnisse zeigen, dass unsere Verfahren den Stand der Technik voranbringen und somit einen nützlichen Beitrag zur Robotik darstellen.

Abstract

Mobile robots are envisioned to revolutionize how people live and work by dealing with everyday tasks such as cleaning and by providing services such as transportation. Such robotic systems need to function autonomously over extended periods of time in a socially compliant way for unobtrusive integration with humans. The robots thereby depend on accurate models, such as a map of the environment for autonomous navigation and, at the same time, models of human behavior for socially compliant human-robot interaction. In most instances, however, these models cannot be provided by human experts. The robots rather need to autonomously learn the models using their on-board sensors.

The contribution of this thesis is a set of novel techniques that enable a robot to learn probabilistic models for socially compliant mobile robot navigation. Learning accurate models from sensor data during long-term operation, however, is challenging owing to computational constraints and the inherent uncertainty in the measurements. In the context of learning maps for robot navigation, we present a technique that reasons about the information gained from the measurements that the robot obtains. This enables the robot to discard highly redundant measurements, which facilitates mapping during long-term operation. In order to build a consistent map that is suitable for navigation, the robot needs to recognize previously observed places, which, however, is challenging, especially in ambiguous environments. We mitigate this problem by having the robot deploy uniquely identifiable artificial landmarks in the environment. Our approach learns an efficient landmark deployment policy that facilitates place recognition when the robot returns. In the context of socially compliant human-robot interaction, we first explore how humans give route directions to others who are unfamiliar with the environment. Our goal is to enable robots to engage in such conversations by imitating humans. Our approach learns a model of the style of a set of descriptions given by a group of humans, which then allows the robot to give natural and intuitive directions just as well, even to goal locations in new environments. To seamlessly integrate mobile robots into everyday life, they require a model of acceptable navigation behavior. In this respect, we first accurately capture the movements of humans by combining readings of inertial measurement units worn by the humans with observations from a mobile robot. Our approach compensates for the drift in the inertial measurements, thereby obtaining accurate estimates of the human movements, even in large areas. The robot can then use these estimates to learn a model of the underlying human navigation behavior. Learning a behavior model, however, is especially challenging when only a limited set of imperfect training examples is available. We propose a framework for learning such a model under these circumstances, which then enables the robot to predict the movements of nearby pedestrians in new situations and to imitate their behavior in order to seamlessly blend in with the humans.

We evaluate the presented methods for learning probabilistic models on real mobile robots and demonstrate that they outperform the state of the art in robotics. Our approaches enable mobile robots to build highly accurate maps of the environment, even in ambiguous environments and during long-term operation. We furthermore demonstrate that our techniques enable socially compliant mobile robot navigation in populated environments. The approaches presented in this thesis are therefore useful for developing flexible mobile robots that autonomously collaborate with humans in a socially compliant way over extended periods of time even in previously unknown environments.

Acknowledgment

It is my pleasure to thank all the wonderful people who contributed to my thesis. First of all, I would like to thank Wolfram Burgard for being a great advisor. I am grateful for the exceptional work environment he has created and for all the amazing opportunities that have arisen owing to his efforts and high expectations. For instance, he sent me to a two-week machine learning summer school in the beginning of my studies. In addition to that, he has frequently sent me to great robotics conferences and workshops all over the world. Even when I was still rather doubtful of new findings, he has encouraged me to submit my work to the most prestigious conferences, and, even though this did not always work out as planned, I have learned a ton in the process. He furthermore suggested that I do an internship in the US, which turned out to be a great experience. Being a member of his Autonomous Intelligent Systems group has been an exciting adventure.

I thank Martin Riedmiller, Thomas Brox, and Bernhard Nebel for agreeing to also serve on my thesis committee. I highly appreciate their time.

I would like to express my deepest gratitude to Cyrill Stachniss for his enduring guidance and support over the years. I enjoyed countless fruitful discussions and collaborations with him and am thankful for the invaluable ideas he contributed to my thesis.

I have been fortunate enough to have collaborated with many other wonderful people over the years. First of all, I would like to single out Markus Kuderer. I remember countless late nights working side by side with him, trying to make our research happen on recalcitrant robots. I no doubt have greatly benefited from our collaboration. What is more, it was a pleasure to work with him at all times, and I am grateful for that.

Furthermore, I thank my collaborators and co-authors for insightful discussions, their hard work, and the resulting fruitful collaborations. In particular, I would like to extend my gratitude to Wolfram Burgard, Cyrill Stachniss, Christian Plagemann, Giorgio Grisetti, Jakob Ziegler, Nichola Abdo, Markus Kuderer, Christoph Sprunk, Maximilian Beinhofer, Luciano Spinello, Kai Wurm, Rainer Kümmerle, Stefan Oßwald, and Jiajun Zhu. In addition to that, I would like to thank the MSc students whom I co-supervised. In particular, I would like to thank Jakob Ziegler, Nichola Abdo, Martin Gloderer, and Philipp Ruchti. It was a great experience to work with all of them.

Many people gave me feedback on early draft manuscripts of this thesis. In particular, I would like to thank Wolfram Burgard, Cyrill Stachniss, Markus Kuderer, Christoph Sprunk, Stefan Oßwald, Maximilian Beinhofer, Bastian Steder, Nichola Abdo, Felix Endres, Barbara Frank, Andreas Wachaja, and Pratik Agarwal. I want to give another shout-out to Christoph Sprunk for always be willing to go the extra mile.

I would like to thank all the members of our lab for the cooperative and intellectually

stimulating environment. I thank my long-term officemate Maximilian Beinhofer for the fun and inspiring atmosphere and for giving me invaluable feedback on my life and research over the years. I also thank my subsequent officemates Andreas Wachaja and Michael Partheil for tolerating that I have mostly been busy writing up this thesis. Still, I greatly enjoyed playing ping-pong during breaks at all times. I wonder whether I would enjoy breaks writing such a thesis if I were a professional table tennis player. Anyway, I especially have fond memories of numerous intense games of ping-pong with Jörg Müller, Markus Kuderer, Jeff Trinkle, Pratik Agarwal, Dan Muñoz, and Andreas Wachaja. I thank Christian Plagemann, Jürgen Sturm, and Dominik Joho for enlightening discussions about machine learning and Rainer Kümmerle for help with graph optimization. I thank Susanne Bourjaillat, Kristine Haberer, and Michael Keser for administrative and technical support.

I thank the numerous volunteers of our "psychological research study". Each one of them spent a stressful hour trying to memorize weird newspaper articles in preparation for a memory test while at the same time walking around our test environment. We could not tell them that there was not going to be a memory test in the end as advertised. Memorizing the articles was merely a distractor task to prevent them from focusing on their walking behavior, which we captured in the course of the study. Our goal was to record natural training data for our algorithms.

I thank Dirk Hähnel, Cyrill Stachniss, and Giorgio Grisetti for providing some of the mobile robot datasets that we used for the experimental evaluation of this thesis. These datasets allowed me to quickly assess the effects of changes in my code when mapping a variety of distinct buildings.

I would like to thank Sven Behnke for inviting me to join his humanoid robot soccer team when I was still an undergraduate student. Being part of his team reinforced my interest in autonomous robots and culminated in competing at the RoboCup world championship in China.

I would like to thank Google for an amazing internship in the company's headquarters in Mountain View, California. The opportunity to contribute to the self-driving car project was a stunning experience. Many thanks to Jiajun Zhu, Dmitri Dolgov, and Chris Urmson for supporting me and letting me see behind the curtain.

Finally and most importantly, I thank my family and friends for supporting me at all times. This thesis would not exist without your ceaseless moral support. Thank you!

This work has partly been supported by the German Research Foundation (DFG) under contract numbers SFB/TR-8 and EXC 1086, and by the European Community (EC) under contract numbers FP7-231888-EUROPA, FP7-610603-EUROPA2, FP7-260026-TAPAS, and FP7-ICT-248873-RADHAR.

Contents

Chapter 1

Introduction

A new generation of autonomous robots is about to change how people live and work. Self-driving cars will safely chauffeur commuters to their workplaces. Domestic robots will reliably perform household chores in private homes, such as cleaning, tidying up, or mowing the lawn. Affordable yet flexible robotic coworkers will help businesses of all sizes in manufacturing, working hand in hand with humans. Delivery robots will quickly bring urgent parcels containing medicine to patients in hospitals or even remote locations. Assistive robots in health care will support elderly and disabled people, restoring their autonomy and capacity to live independently. Such breakthroughs in technology are widely expected to have a broad impact on society and the economy.

Today's robots, however, are mostly stationary and limited to performing menial and repetitive tasks in manufacturing and assembly. For instance, the automotive industry has long relied on the speed, precision, and force of these machines to address tasks such as welding, assembly, and packaging. Most of today's robots need to be manually customized to specific applications in highly structured environments, such as an assembly line. Changes in the applications then typically entail time-consuming and costly modifications to the robots that need to be carried out by skilled programmers. Deploying such robots therefore usually only pays off when using them for mass production. Furthermore, owing to their inability to properly adapt to humans, most of these robots need to operate behind fences or in safety cages in order to prevent nearby workers from getting injured or even killed. These constraints further limit the applicability of the robots.

Developing a flexible robot that overcomes these limitations by autonomously collaborating with humans in a socially compliant way potentially moving in previously unknown environments is highly challenging for numerous reasons. For instance, a robot that needs to autonomously move in the environment to carry out tasks such as cleaning the house or running an errand needs an accurate map of the environment. Such a map allows the robot to pinpoint its location in the environment and plan a route to its destination. The robot can then use the map during navigation to keep track of its position. Endowing the robot with a detailed map of its target environment prior to its deployment, however, is infeasible in many applications. The target environment may be unknown. The robot manufacturer may not have the necessary information and is therefore unable to deliver a map. Even once the robot has been equipped with a map, changes in the environment may immediately invalidate that map. For these reasons, the robot itself needs to be able to infer a map from its own sensor measurements during operation.

Learning a map of the environment is a fundamental problem in mobile robotics, which has been investigated intensively in the past. As a result, the robotics community has developed effective techniques that build highly accurate maps. Learning a map of an initially unknown environment requires the robot to incorporate its on-board sensor measurements into a map while at the same time keeping track of its location within that very map. The task is therefore known as simultaneous localization and mapping. State-of-the-art techniques tend to store all the sensor measurements that the robot obtains throughout its mission in the memory of the robot. During long-term operation, however, the memory may fill up. As a result, the robot is unable to store new measurements and may cease to operate correctly. In Chapter 3, we present a novel approach to simultaneous localization and mapping that enables the robot to discard measurements in order to free memory for new measurements. The measurements that the robot obtains are typically highly redundant, especially when the robot moves in previously observed parts of the environment. The reason is that many sensor readings typically capture the same parts of the environment. Our approach therefore discards uninformative measurements that mostly convey information about the environment that has already been captured by other measurements. Consequently, as long as the robot stays in the same area, our approach prevents the memory usage from growing, which enables longer missions.

Despite the remarkable progress in the field of simultaneous localization and mapping in recent years, building a map of the environment still proves to be a difficult problem. The bottleneck turns out to be the robots' ability to recognize previously observed places, which is required to build a consistent map. Especially in large, ambiguous environments, this data association problem can become intractable, even for state-of-the-art techniques. A robot that is able to deploy a limited number of uniquely identifiable artificial landmarks while mapping the environment can use these landmarks later on to resolve ambiguities. The question arises as to when the robot should deploy these landmarks. State-of-the-art approaches typically rely on handcrafted heuristics to decide when to deploy the landmarks during navigation. These heuristics, however, often lead to suboptimal choices that may not help the robot find correct data associations. In Chapter 4, we propose a novel method that allows the robot to learn when to place its artificial landmarks in the environment in order to optimize data association performance.

In contrast to most of today's industrial robots that are carefully separated from human workers, the new generation of robots is supposed to share the environment with people. It is desirable that these robots behave in a socially compliant way when interacting with them. For instance, they should be able to communicate with them in a natural and intuitive way. A common conversation that pedestrians have is to give directions to people that are unfamiliar with the environment. Socially compliant mobile robots should understand how humans describe routes to each other in these situations so that they can engage in such conversations just as well. State-of-the-art methods, however, typically provide instructions according to manually programmed rules. Even though the resulting directions are precise, they are not necessarily natural and intuitive. Chapter 5 therefore presents a novel approach that allows a robot to learn how to give natural and intuitive route directions based on a set of directions given by humans.

Furthermore, a robot that autonomously navigates in an environment that is populated by humans should move in a way that does not disturb people nearby. At the same time, the robot should be able to efficiently reach its destination without taking excessive detours in order to avoid pedestrians. For that purpose, the robot needs a model of acceptable behavior. One way to achieve socially compliant mobile robot navigation is to observe how people move in the environment of the robot and then learn a model of their behavior from these observations. To this end, human motion capture provides means of observing and recording the movements of people. On the one hand, there are optical motion capture systems that typically depend on statically mounted, highly specialized cameras that track markers attached to the people. On the other hand, there are inertial motion capture systems that rely on a set of inertial measurement units attached to the people. Whereas optical motion capture systems are restricted to the area that can be captured by the cameras, inertial motion capture systems suffer from drift, which prevents us from properly modeling the behavior of the people in the context of the environment. To combine the advantages of both systems, Chapter 6 introduces a novel approach that allows us to compensate for the drift in inertial motion capture by using a mobile robot to anchor the inertial motion capture estimates in a map of the environment using its on-board laser scanner. By doing this, we obtain highly accurate estimates of the movements of people even in large environments.

Mobile robot navigation in an environment populated by humans is a challenging problem. Most state-of-the-art methods compute a path to the destination of the robot without taking into account the behavior of humans in the vicinity. Even though such methods effectively avoid collisions with nearby humans, the resulting behavior of the robot often does not match the expectations of the humans owing to awkward and unexpected evasive movements of the robot. Humans, on the other hand, are capable of putting themselves in the positions of others to predict their movements. These predictions enable them to pursue safe, smooth, and efficient cooperative navigation strategies. In order to seamlessly blend in with the humans, socially compliant mobile robots therefore need to engage in such cooperative collision avoidance strategies just as well. To this end, the robots need to be able to reason about the intentions of the humans, which requires that the robots have accurate models of their navigation behavior. The behavior of the humans, however, depends on their environment and their culture, which is why it is often infeasible to manually provide these behavior models to the robots in advance. The robots rather need means to autonomously infer such models that capture the behavior of the humans in any given environment. As a main contribution of this thesis, Chapter 7 finally leverages the above-mentioned results by developing a novel probabilistic framework for learning a model of the navigation behavior that underlies observed trajectories of the humans in the environment. By explicitly modeling their cooperative navigation behavior, our method captures the way humans react to each other during encounters, and, additionally, how they react to the robot. The resulting model enables the robot to predict the movements of nearby humans in the current situation and to imitate their behavior in order to seamlessly blend in with them.

1.1 Contributions

We contribute to the field of robotics research by embracing machine learning to infer probabilistic models for mobile robot navigation from noisy sensor measurements. In summary, we propose the following novel approaches:

- Chapter 3 presents a method for discarding uninformative laser range measurements in simultaneous localization and mapping in order to free memory and enable long-term mapping. We apply information-theoretic principles to minimize the uncertainty in the map of the environment.

- Chapter 4 proposes a reinforcement learning approach that allows a mobile robot to learn where to place a limited number of artificial landmarks in the environment during navigation. When the robot comes back to the same location, such a landmark may help the robot solve the data association problem in simultaneous localization and mapping.

- Chapter 5 presents an approach that allows a mobile robot to learn how to give natural and intuitive route directions based on a set of directions given by humans.

- Chapter 6 introduces an approach to human motion capture, which can be used to record the trajectories of pedestrians to learn a model of their navigation behavior. Our method combines locally accurate inertial motion capture with precise global alignment in a map of the environment by means of a localized mobile robot equipped with laser range finders.

- Chapter 7 formulates a probabilistic framework to learn a model of the cooperative navigation behavior of interacting agents, such as pedestrians, from motion capture data. A mobile robot can use that model for socially compliant navigation.

1.2 Publications

The work presented in this thesis is based on the following publications in conference proceedings, workshop proceedings, and journals:

- Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

- Stefan Oßwald, Henrik Kretzschmar, Wolfram Burgard, and Cyrill Stachniss. Learning to give route directions from human demonstrations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014. **Best Cognitive Robotics Paper - Finalist**.

- Markus Kuderer, Henrik Kretzschmar, and Wolfram Burgard. Teaching mobile robots to cooperatively navigate in populated environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.
- Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Predicting human navigation behavior via inverse reinforcement learning. In *The 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, Princeton, NJ, USA, 2013.
- Henrik Kretzschmar and Cyrill Stachniss. Information-theoretic compression of pose graphs for laser-based SLAM. *The International Journal of Robotics Research (IJRR)*, 31:1219–1230, 2012.
- Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Learning navigation policies from human demonstrations. In *Proceedings of the Workshop on Inverse Optimal Control & Robotic Learning from Demonstration at Robotics: Science and Systems (RSS)*, Berlin, Germany, 2013.
- Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Inferring navigation policies for mobile robots from demonstrations. In *Proceedings of the Autonomous Learning Workshop at the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.
- Maximilian Beinhofer, Henrik Kretzschmar, and Wolfram Burgard. Deploying artificial landmarks to foster data association in simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.
- Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Proceedings of Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.
- Jakob Ziegler, Henrik Kretzschmar, Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Accurate human motion capture in large areas by combining IMU- and laser-based people tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, 2011.
- Cyrill Stachniss and Henrik Kretzschmar. Pose graph compression for laser-based SLAM. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, Flagstaff, AZ, USA, 2011. **Invited presentation**.
- Henrik Kretzschmar, Cyrill Stachniss, and Giorgio Grisetti. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, 2011.

The following publications were also written during the author's time at the research group, yet they are not included in this thesis:

- Markus Kuderer, Christoph Sprunk, Henrik Kretzschmar, and Wolfram Burgard. Online generation of homotopically distinct navigation paths. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

- Kai M. Wurm, Henrik Kretzschmar, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. Identifying vegetation from laser data in structured outdoor environments. *Robotics and Autonomous Systems*, 62:675–684, 2014.

- Nichola Abdo, Henrik Kretzschmar, Luciano Spinello, and Cyrill Stachniss. Learning manipulation actions from a few demonstrations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

- Nichola Abdo, Henrik Kretzschmar, and Cyrill Stachniss. From low-level trajectory demonstrations to symbolic actions for planning. In *Proceedings of the ICAPS Workshop on Combining Task and Motion Planning for Real-World Applications (TAMPRA)*, Atibaia, São Paulo, Brazil, 2012.

- Henrik Kretzschmar, Giorgio Grisetti, and Cyrill Stachniss. Lifelong map learning for graph-based SLAM in static environments. *KI – Künstliche Intelligenz*, 24: 199–206, 2010.

- Henrik Kretzschmar, Cyrill Stachniss, Christian Plagemann, and Wolfram Burgard. Estimating landmark locations from geo-referenced photographs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008.

1.3 Collaborations

Parts of this thesis are the result of collaboration with other researchers. The approach to information-theoretic graph pruning in simultaneous localization and mapping described in Chapter 3 was joint work with Cyrill Stachniss. The work on learning to deploy landmarks in simultaneous localization and mapping presented in Chapter 4 is the result of a collaboration with Maximilian Beinhofer. The approach to learning how to give route directions presented in Chapter 5 was joint work with Stefan Oßwald and Cyrill Stachniss. The approach to human motion capture presented in Chapter 6 is an extension of Jakob Ziegler's Master's thesis, which the author co-supervised with Giorgio Grisetti, followed by Cyrill Stachniss. Finally, the work on socially compliant mobile robot navigation presented in Chapter 7 was developed in collaboration with Markus Kuderer.

1.4 Outline

This thesis is structured as follows. Chapter 2 briefly reviews background material that will be leveraged throughout the remainder of this thesis. The subsequent chapters detail the main contributions of this thesis. In Chapter 3, we apply information theory to the problem of discarding laser measurements in simultaneous localization and mapping in a way that minimizes the loss of information about the environment. Chapter 4 uses reinforcement learning in order to compute a policy that allows a mobile robot to deploy artificial landmarks in the environment that help the robot solve the data association problem. In Chapter 5, we formulate the problem of giving route directions as a reinforcement learning problem. We then use inverse reinforcement learning in order to find a reward function that captures the way humans describe routes. In Chapter 6, we use recursive state estimation to compensate for the drift in inertial motion capture by using the laser range measurements obtained by a mobile robot. This allows us to obtain highly accurate estimates of the movements of pedestrians. Chapter 7 formulates a probabilistic framework to learn a model of the cooperative navigation behavior of pedestrians from demonstrated behavior.

Chapter 2

Background

We introduce most of the basic definitions and review some of the theoretical foundations underlying the techniques presented in the remainder of this thesis.

This chapter briefly introduces some definitions and concepts from probability theory and information theory that we will leverage throughout the remainder of this thesis. Probability theory is the analysis of random phenomena, which arise, for instance, owing to noise that affects sensor measurements. Since properly dealing with uncertainty is of pivotal importance when modeling robotic systems, we use a probabilistic approach to modeling the behavior of the robots. We will therefore first take a look at probability theory, and, in particular, discuss random variables as a means of representing uncertainty. We will then introduce some measures from the field of information theory to quantify the uncertainty involved in predicting the values of the random variables. The definitions in this chapter establish a basis for the probabilistic models that we develop in this thesis. The chapter is based on Bishop [21], Cover and Thomas [35], and MacKay [124].

2.1 Probability Theory

We use random variables to describe the uncertainty in the outcomes of sensor measurements. We furthermore use random variables to model human behavior. Formally, a random variable X on a discrete space $\mathcal{X}$ of values can be described by a probability mass function P that assigns a probability $P(X = x)$ to each value $x \in \mathcal{X}$. In contrast to that, a random variable X on a continuous space $\mathcal{X}$ of values can be described in terms of a probability density function p, where the probability that the value of X will be in a particular subset $\mathcal{X}' \subseteq \mathcal{X}$ of values is given by the integral of the probability density over that subset, i. e.,

$$P(X \in \mathcal{X}') = \int_{x \in \mathcal{X}'} p(X = x)\, dx. \tag{2.1}$$

The expected value of a random variable X on a discrete space $\mathcal{X}$ is the probability-weighted average of all possible values, i. e.,

$$\mathbb{E}[X] := \sum_{x \in \mathcal{X}} p(X = x)\, x. \tag{2.2}$$

The expected value of a random variable X on a continuous space $\mathcal{X}$ is defined by

$$\mathbb{E}[X] := \int_{x \in \mathcal{X}} p(X = x)\, x\, dx. \tag{2.3}$$

We will often be interested in the average value of some function f with respect to a probability distribution P. This average value is called the expectation of f and is given by

$$\mathbb{E}[f] := \sum_{x \in \mathcal{X}} P(X = x) f(x). \tag{2.4}$$

When dealing with random variables on continuous spaces, the expectation of function f with respect to a probability distribution with density p is given by

$$\mathbb{E}[f] := \int p(X = x) f(x)\, dx. \tag{2.5}$$

We can approximate the expectation of function f with respect to a probability distribution based on a set $\{x^{(k)}\}_{k=1}^{K}$ of samples drawn independently from the distribution. This approximation is given by

$$\mathbb{E}[f] \approx \frac{1}{K} \sum_{k=1}^{K} f\left(x^{(k)}\right) \tag{2.6}$$

and converges to the true expectation of f as K approaches infinity. The variance of a random variable X is defined by

$$\mathrm{Var}(X) := \mathbb{E}[(X - \mathbb{E}[X])^2]. \tag{2.7}$$

The probability that a random variable X defined on space $\mathcal{X}$ will take the value x and a random variable Y defined on space $\mathcal{Y}$ will take the value y is called the joint probability $P(X = x, Y = y)$. The random variables X and Y are called independent if

$$P(X = x, Y = y) = P(X = x)P(Y = y). \tag{2.8}$$

The sum rule of probability states that

$$P(X = x) = \sum_{y \in \mathcal{Y}} P(X = x, Y = y), \tag{2.9}$$

where, in this context, $P(X = x)$ is called the marginal probability. The probability that random variable X will take the value x given that random variable Y takes the value y is denoted by the conditional probability $P(X = x \mid Y = y)$. The product rule of probability states that

$$P(X = x, Y = y) = P(Y = y \mid X = x)P(X = x). \tag{2.10}$$

Finally, the Bayes' theorem states that

$$P(Y = y \mid X = x) = \frac{P(X = x \mid Y = y)P(Y = y)}{p(X = x)}. \tag{2.11}$$

Up to this point, we have carefully differentiated between random variables and the values they can take. To achieve a concise notation in the remainder of this thesis, we define

$$P(x) := P(X = x) \tag{2.12}$$
$$P(x \mid y) := P(X = x \mid Y = y) \tag{2.13}$$

and

$$p(x) := p(X = x) \tag{2.14}$$
$$p(x \mid y) := p(X = x \mid Y = y). \tag{2.15}$$

2.2 Information Theory

Entropy is a measure of the uncertainty expressed by a random variable. Formally, the entropy H of a random variable X on a discrete space $\mathcal{X}$ with probability mass function P is defined by

$$H(P) := H(X) := -\sum_{x \in \mathcal{X}} P(X = x) \log\left(P(X = x)\right). \tag{2.16}$$

Consequently, the entropy of a random variable only depends on the probabilities of its values and not on the values taken by the random variable. When the logarithm is to base 2, then entropy is expressed in bits. The entropy of a continuous random variable is called the differential entropy. Formally, the differential entropy h of a random variable X on a continuous space $\mathcal{X}$ with probability density function p is defined by

$$h(p) := h(X) := -\int p(X = x) \log\left(p(X = x)\right) dx. \tag{2.17}$$

We can extend this definition of differential entropy to a set $\{X_i\}_1^n$ of random variables, leading to

$$h(X_1, \ldots, X_n) := -\int p(X_1 = x_1, \ldots, X_n = x_n) \log\left(p(X_1 = x_1, \ldots, X_n = x_n)\right) d(x_1, \ldots, x_n). \tag{2.18}$$

If two random variables X and Y on continuous spaces are given by a joint probability density function p, then we can define the conditional differential entropy $h(X \mid Y)$ as

$$h(X \mid Y) := -\int p(X = x, Y = y) \log\left(p(X = x, Y = y)\right) dx\, dy. \tag{2.19}$$

We can rewrite the conditional differential entropy $h(X \mid Y)$ as

$$h(H \mid Y) = h(X, Y) - h(Y). \tag{2.20}$$

The relative entropy $D(f||g)$, which is also called the Kullback-Leibler divergence, between two probability distributions given by the probability density functions f and g is defined by

$$D(f||g) := \int f(X = x) \log \left(\frac{f(X = x)}{g(X = x)} \right) \, dx. \tag{2.21}$$

The mutual information $I(X; Y)$ between two random variables X and Y on continuous spaces with joint probability density $p(X = x, Y = y)$ is defined by the relative entropy between the joint distribution and the product distribution. Hence we have

$$I(X; Y) := \int p(X = x, Y = y) \log \left(\frac{p(X = x, Y = y)}{p(X = x)\, p(Y = y)} \right) \, dx \; dy. \tag{2.22}$$

We can rewrite the mutual information $I(X; Y)$ in terms of the entropy, leading to

$$I(X; Y) = h(X) - h(X \mid Y) \tag{2.23}$$

$$= h(Y) - h(Y \mid X). \tag{2.24}$$

Consequently, the mutual information $I(X; Y)$ corresponds to the reduction in uncertainty of the random variable X owing to the knowledge of Y.

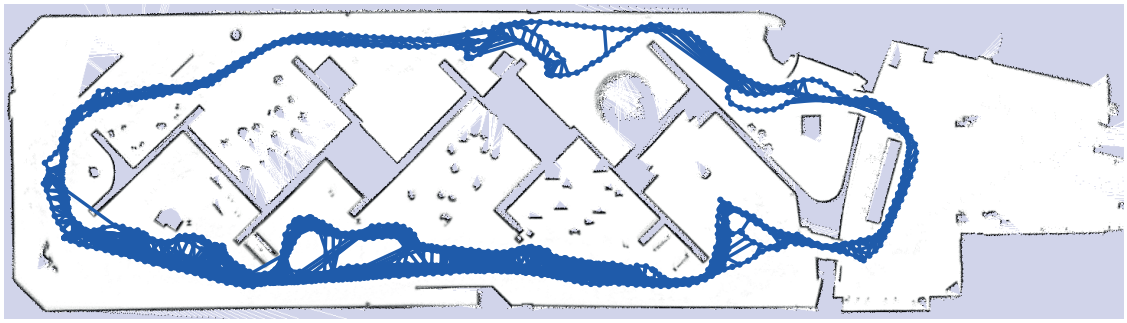
Chapter 3

Discarding Measurements for Efficient Mapping with Mobile Robots

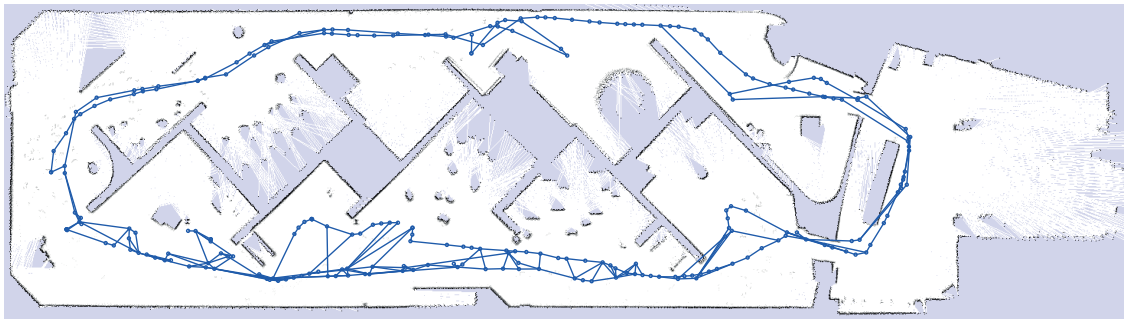
In graph-based simultaneous localization and mapping, the graph that models the belief of the robot grows over time as the robot gathers information about its environment. A strictly growing graph, however, may prevent long-term mapping with mobile robots. We present a novel approach to pruning the graph. Our approach estimates the mutual information between the laser measurements and the grid map in order to discard laser measurements that provide little information about the environment. The nodes in the graph that correspond to the discarded laser measurements can then be marginalized out in a way that preserves the sparsity of the graph. Our method allows a mobile robot to effectively restrict the size of the graph, which enables long-term mobile robot mapping, especially when the robot frequently traverses previously observed parts of the environment. An extensive set of experiments with real mobile robots suggests that our approach discards laser measurements in a way that leads to maps that are highly similar to the maps that result from integrating all laser measurements in the mapping process.

A highly effective approach to simultaneous localization and mapping (SLAM) is to model the belief of the robot in terms of a graph. The nodes in the graph correspond to poses of the robot at discrete points in time, whereas the edges in the graph represent spatial soft constraints between the poses that result from measurements obtained by the robot. The graph effectively is a representation of the optimization problem of estimating the robot poses that are most likely given all the measurements. The robot can then use the solution to this optimization problem to compute a map of the environment.

Most of the existing graph-based approaches to simultaneous localization and mapping assume that map learning is carried out as a preprocessing step and that the robot then uses the acquired model for localization and path planning. Consequently, a robot that needs to extend the map of its environment during long-term operation cannot rely on these methods, since their complexity grows with the length of the robot's trajectory. The



(a) Standard graph-based approach to SLAM. 2049 laser scans.



(b) Our approach to SLAM. 250 laser scans.

Figure 3.1: We prune the graph in simultaneous localization and mapping in order to enable efficient long-term mobile robot mapping. Our method discards laser measurements, here depicted as nodes in the graph, in a way that minimizes the loss of information about the environment. Although in this example our approach discards more than 85 % of the 2049 laser measurements that the robot recorded, only 1.2 % of the grid cells of the resulting occupancy grid map changed their most likely state. The standard SLAM approach required 32 min to process the dataset, whereas our approach only needed 17 min.

reason for this is that standard graph-based approaches to SLAM add more and more new nodes to the graph as the robot records new measurements. A strictly growing graph slows down graph optimization and increases the computational cost of finding spatial constraints between the current pose and former poses, a process known as closing the loop. As a result, memory and computational requirements grow over time, impeding long-term mobile robot mapping applications.

The contribution of this chapter is a novel approach to pruning the graph in SLAM that enables efficient long-term mobile robot mapping by discarding uninformative laser range measurements. See Figure 3.1 for a motivating example. Our method discards uninformative laser measurements in a way that minimizes the loss of information about the environment. To this end, our method explicitly reasons about the effect of the laser measurements on the grid map of the environment. This is in contrast to existing methods for discarding laser measurements while mapping that keep every nth measurement or that aim at equally distributing the measurements in the environment. In contrast to these methods, for instance, our approach implicitly takes into account the field of view of the

robot's laser range finder and its heading when obtaining the measurements. Formally, we are interested in the subset of laser measurements that maximizes the mutual information between the grid map and that subset of laser measurements. In general, computing the mutual information requires solving an integral over the space of all possible laser range measurements, which clearly is infeasible. However, expressing the entropy of the grid map in terms of the effects of the laser measurements on the grid cells allows us to compute the mutual information much more efficiently. The nodes in the graph that correspond to the discarded laser measurements can subsequently be marginalized out by using approximate techniques that preserve the sparsity of the graph [31, 98, 129]. An extensive set of experiments suggests that our approach discards laser measurements in a way that leads to maps that are highly similar to the maps that result from integrating all laser measurements. The experiments furthermore demonstrate that our approach is suitable for long-term mobile robot mapping.

3.1 Finding the Most Informative Measurements

We are interested in finding a subset of the recorded laser measurements of a given size that maximizes the information about the environment. Formally, we define the map $\mathbf{M}$ as a random variable that describes the state of the world. The map is highly correlated with the random variables $\mathbf{Z}_{1:T}$ that describe the laser measurement outcomes $\mathbf{z}_{1:T}$ obtained by the robot at the poses $\mathbf{x}_{1:T}$, respectively. We use Z_i^j to refer to the random variable of an individual beam of laser scan $\mathbf{Z}_i$. To estimate the state of the world, we consider the posterior probability distribution of the map $\mathbf{M}$ conditioned on the laser measurement outcomes $\mathbf{z}_{1:T}$.

We are interested in finding the subset $\mathbf{Z}^\star \subseteq \mathbf{Z}_{1:T}$ consisting of at most n laser measurements that minimizes the uncertainty in the map $\mathbf{M}$. The average reduction in uncertainty in the map $\mathbf{M}$ resulting from observing a set $\mathbf{Z}$ of laser measurements is captured by the mutual information I between the map and the laser measurements, leading to

$$I(\mathbf{M}; \mathbf{Z}) = \int_{\mathbf{M}} \int_{\mathbf{Z}} p(\mathbf{M} = \mathbf{m}, \mathbf{Z} = \mathbf{z}) \, \log \left(\frac{p(\mathbf{M} = \mathbf{m}, \mathbf{Z} = \mathbf{z})}{p(\mathbf{M} = \mathbf{m}) \, p(\mathbf{Z} = \mathbf{z})} \right) d\mathbf{z} \, d\mathbf{m}. \tag{3.1}$$

For a formal definition of the mutual information, see Section 2.2. We can equivalently express mutual information in terms of entropy, leading to

$$I(\mathbf{M}; \mathbf{Z}) = h(\mathbf{M}) - h(\mathbf{M} \mid \mathbf{Z}), \tag{3.2}$$

where h refers to the differential entropy of continuous random variables, defined in Section 2.2. Consequently, we are interested in the subset $\mathbf{Z}^\star \subseteq \mathbf{Z}_{1:T}$ of at most n laser measurements such that the mutual information between the map $\mathbf{M}$ and $\mathbf{Z}^\star$ is maximized, yielding

$$\mathbf{Z}^\star = \operatorname*{argmax}_{\mathbf{Z} \subseteq \mathbf{Z}_{1:T}, |\mathbf{Z}| \leq n} h(\mathbf{M}) - h(\mathbf{M} \mid \mathbf{Z}). \tag{3.3}$$

The conditional entropy $h(\mathbf{M} \mid \mathbf{Z})$ of the map $\mathbf{M}$ given the set $\mathbf{Z}$ of measurements is the expected value of the conditional entropy of the map given the measurement outcomes $\mathbf{z}$ with respect to the distribution over the space of all possible measurement outcomes, i. e.,

$$h(\mathbf{M} \mid \mathbf{Z}) = \int p(\mathbf{z})\, h(\mathbf{M} \mid \mathbf{Z} = \mathbf{z})\, d\mathbf{z}. \tag{3.4}$$

Computing this expectation is infeasible in general because integrating over the space of laser measurements is computationally intractable. Moreover, evaluating the entropy $h(\mathbf{M} \mid \mathbf{Z} = \mathbf{z})$ of the map $\mathbf{M}$ given a set of measurement outcomes $\mathbf{z}$ requires model assumptions about the world. In the next section, we will make such model assumptions that will allow us to approximate the entropy of the map.

3.2 Model Assumptions

The following assumptions allow us to approximate the entropy of the map of the environment. First, we assume the environment to be static. We approximate the probability distribution over the robot poses with a Dirac delta function at the most likely estimate. Similar to most approaches in robotics (see Thrun et al. [164] for an overview), we assume that the laser measurements and the individual laser beams are independent of each other. Moreover, we model the map $\mathbf{M}$ as a standard occupancy grid map. An occupancy grid map is a grid of independent discrete binary random variables C that take the values $\text{Val}(C) = \{\text{“free”}, \text{“occupied”}\}$. Therefore, in the following, we will turn our attention to the entropy H of discrete random variables, as opposed to the differential entropy h of continuous random variables. The entropy of a grid cell C given a set of measurement outcomes $\mathbf{z}$ is given by

$$H(C \mid \mathbf{Z} = \mathbf{z}) = -\sum_{c \in \text{Val}(C)} P(C = c \mid \mathbf{z}) \log P(C = c \mid \mathbf{z}), \tag{3.5}$$

where the probability $P(C = \text{occupied} \mid \mathbf{z})$ that the cell is actually occupied is typically estimated by means of a Bayes filter. More specifically, we rely on a standard inverse measurement model for laser range scanners, which updates each cell based on three values l_{free}, l_{occ}, and l_0. Figure 3.2 depicts the entropy $H(C \mid \mathbf{Z} = \mathbf{z})$ of a grid cell C given a set of measurement outcomes $\mathbf{z}$ as a function of the probability of the cell being occupied given the measurement outcomes. Since the joint entropy of mutually independent random variables is given by the sum of the entropies of the random variables, the entropy of an occupancy grid map $\mathbf{M}$ given a set of measurement outcomes $\mathbf{z}$ is then given by

$$H(\mathbf{M} \mid \mathbf{Z} = \mathbf{z}) = \sum_{C \in \mathbf{M}} H(C \mid \mathbf{Z} = \mathbf{z}). \tag{3.6}$$

Furthermore, to reason about the mutual information between the map and the measurements, we need to make assumptions about the likelihood of sensing an object at a certain range. One approach is to assume that the likelihood of sensing an object decreases with

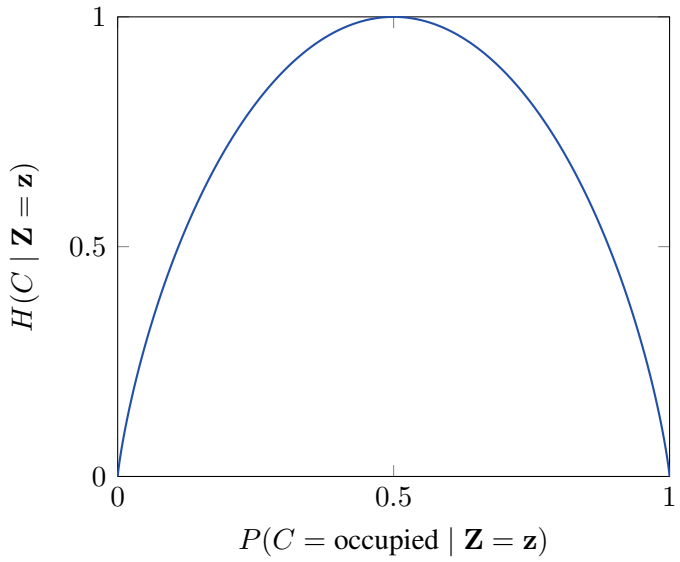


Figure 3.2: Entropy $H(C \mid \mathbf{Z} = \mathbf{z})$ of a grid cell C given a set of measurement outcomes $\mathbf{z}$ as a function of the probability $P(C = \text{occupied} \mid \mathbf{Z} = \mathbf{z})$ that the cell is occupied.

range [164]. The prior probability distribution of ranges being measured can then be described independently of the map $\mathbf{M}$ and the estimate of the robot pose $\mathbf{X}_t$ in terms of an exponential distribution. This leads to the density

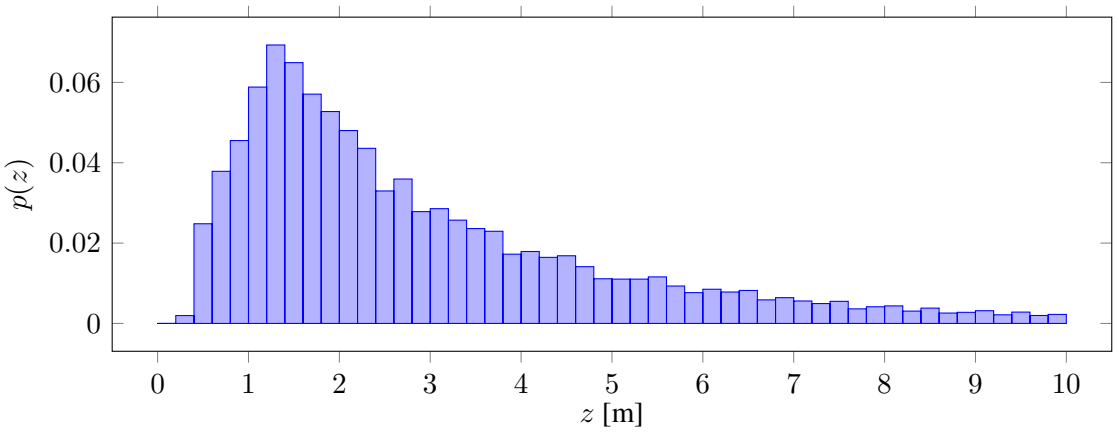
$$p(Z_i^j = z_i^j) = \begin{cases} \eta\lambda e^{-\lambda z_i^j} & \text{if } z_i^j \leq z_{\max} \\ 0 & \text{if } z_i^j > z_{\max}, \end{cases} \tag{3.7}$$

where z_i^j refers to the range reading corresponding to beam j of measurement z_i, and $z_{\max}$ corresponds to the maximum range of the scanner. Moreover, λ is an intrinsic parameter of the measurement model, and η is a normalizing constant.

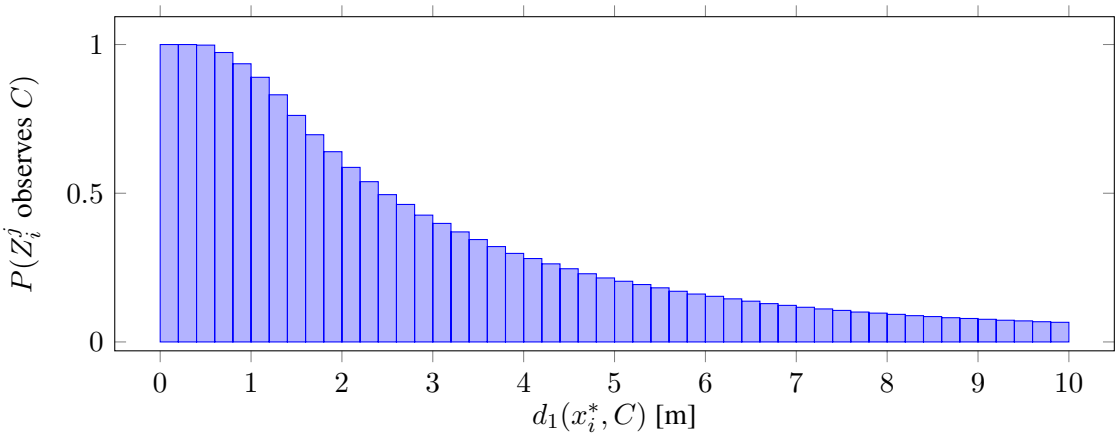
We can also learn this distribution from data. To this end, we computed a statistic of laser range measurements averaged over all beams, scans and datasets based on a variety of publicly available datasets found in the Radish dataset repository provided by Howard and Roy [72]. Figure 3.3 depicts the distribution of laser range measurements found in the datasets. The plots suggest that for range readings beyond 1.5 m, the learned distribution is fairly similar to the exponential distribution given in Equation (3.7). In the experimental evaluation, we still consider both measurement models.

3.3 Approximating Mutual Information

Computing the map entropy in Equation (3.4) by integrating over the space of all possible range measurements is infeasible. However, formulating the probability distribution of the map in terms of the effects that the laser measurement outcomes have on the individual grid cells allows us to compute the map entropy much more efficiently.



(a) Learned probability distribution $p(z)$ of the laser range readings.



(b) Probability of measuring a cell C that is $d_1(x_i^\star, C)$ meters away from the sensor location $x_i^\star$.

Figure 3.3: Statistics of laser range measurements extracted from a series of robotic datasets.

Specifically, there are three possible measurement outcomes of a laser beam with respect to a particular grid cell that is located along the ray of the beam. The laser beam either traverses the cell and thus observes the cell as free, the laser beam ends in the cell and thus observes the cell as occupied, or the laser beam does not observe the cell. Given no prior map information, the probability distribution of the outcome can be computed by integrating over the density $p(z_i^j)$, leading to

$$P(Z_i^j \text{ does not observe } C) = \int_0^{d_1(x_i^\star, C)} p(z_i^j)\, dz_i^j \tag{3.8}$$

$$P(Z_i^j \text{ observes } C \text{ as occupied}) = \int_{d_1(x_i^\star, C)}^{d_2(x_i^\star, C)} p(z_i^j)\, dz_i^j \tag{3.9}$$

$$P(Z_i^j \text{ observes } C \text{ as free}) = \int_{d_2(x_i^\star, C)}^{z_{\max}} p(z_i^j)\, dz_i^j, \tag{3.10}$$

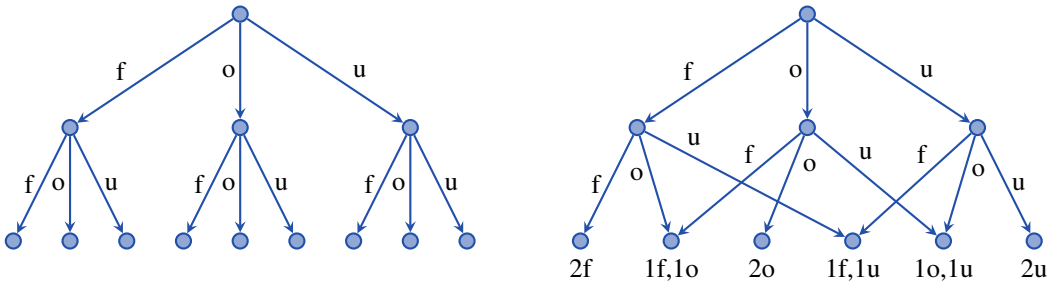


Figure 3.4: Space of measurement outcomes of a grid cell given k laser scans, where $k = 2$. Here, "free" is denoted by f, "occupied" by o, and "unknown" by u. Left: In general, there are 3^k measurement outcomes, which is exponential in k. Right: The number of measurement outcomes is quadratic in k when exploiting the standard measurement model.

where $d_1(x_i^\star, C)$ is the distance between the pose $x_i^\star$ from which the laser scan Z_i is taken and the closest border of the grid cell C in the direction of the j^{th} beam, i. e. the border at which the beam enters the cell. Similarly, $d_2(x_i^\star, C)$ is the distance to the border of the grid cell C in the direction of the j^{th} beam that is furthest away from $x_i^\star$, i. e. the border at which the beam leaves the cell.

Exploiting the probability distribution of the measurement outcome expressed in Equations (3.8), (3.9), and (3.10) allows us to avoid integrating over the space of all laser range measurements. Instead, we consider the effects of the measurements on the estimates of the grid cells, yielding

$$I(\mathbf{M}; \mathbf{Z}) = \sum_{C \in \mathbf{M}} I(C; \mathbf{Z}) \tag{3.11}$$

with

$$I(C; \mathbf{Z}) = H(C) - \sum_{\mathbf{z}' \in \mathcal{A}_\mathbf{Z}} P(\mathbf{Z} = \mathbf{z}') H(C \mid \mathbf{Z} = \mathbf{z}'), \tag{3.12}$$

where $\mathcal{A}_\mathbf{Z}$ refers to the set of measurement outcomes with respect to the grid cell C due to the laser scans that are recorded close enough to potentially measure the cell.

As illustrated in Figure 3.4, the number of possible combinations of grid cell measurement outcomes is exponential in the number k of laser scans in general. It is therefore infeasible to enumerate all the combinations in a tree. However, our approach relies on a measurement model that updates each cell based on the three values l_{free}, l_{occ}, and l_0, which do not depend on the sensed range. Therefore, the effect of a set of observations on a particular cell exclusively depends on the number of free and occupied observations, i. e., the histogram of measurement outcomes. Furthermore, the effect does not depend on the order in which the measurements were obtained. This allows us to combine all nodes in the tree that represent the same histogram of measurement outcomes. The number $A(k)$ of measurement outcomes is given by the number of ways to sample k elements from the set of three laser measurement outcomes with replacement and disregarding different

Algorithm 1 Compute the probabilities of the measurement outcomes for a grid cell.

Input: set of k laser measurements $\mathbf{Z}$, cell C
Output: probabilities $P(\langle \cdot, \cdot, \cdot \rangle)$ of the outcomes ("free", "occupied", "not observed")

```
P(⟨0, 0, 0⟩) = 1
for r = 1 ... k do
  for all ⟨f, o, u⟩ with f + o + u == r do
    P(⟨f + 1, o, u⟩) += P(⟨f, o, u⟩)P(Z_i^j observes C as free)
    P(⟨f, o + 1, u⟩) += P(⟨f, o, u⟩)P(Z_i^j observes C as occupied)
    P(⟨f, o, u + 1⟩) += P(⟨f, o, u⟩)P(Z_i^j does not observe C)
  end for
end for
```

orderings, leading to

$$A(k) = \sum_{i=0}^{k} \left(\sum_{j=0}^{k-i} 1 \right) = \frac{k^2}{2} + \frac{3}{2}k + 1 \quad \in \mathcal{O}(k^2), \tag{3.13}$$

which is quadratic in k.

Note that the probability of each of these outcomes would be given by a multinomial distribution if the free and occupied measurements were equally likely for all laser scans. However, Figure 3.3 suggests that the probability of observing a cell depends on the distance from the sensor to the cell. Algorithm 1 computes the probabilities of the outcomes by propagating the probability mass through the graph, using a hash table $P(\langle \cdot, \cdot, \cdot \rangle)$ that is indexed by the measurement histograms. The number of probabilities that the algorithm needs to compute in total is given by

$$3 \sum_{i=1}^{k} A(i) = \frac{3}{2}k^3 + \frac{9}{2}k^2 + 3k \quad \in \mathcal{O}(k^3), \tag{3.14}$$

leading to cubic complexity in the number k of measurements that potentially observe grid cell C.

The maximum measurement range of the laser scanner restricts the set of scans that can measure a cell. We can reduce the computational burden by only considering the $l < k$ laser scans that most affect the mutual information between the measurements and the grid cell, i. e., the scans with the highest likelihood of measuring C. These are the l scans that are recorded closest to C. The approximation of using only l instead of k scans is appealing because it yields a complexity that is linear in k since choosing these l scans is linear in k. However, the approximation neglects a subset of the scans when computing the mutual information between the measurements and the grid cell.

3.4 Discarding Laser Measurements

Computing the optimal subset $\mathbf{Z}^\star$ of n measurements that most reduces the uncertainty of an estimate of the state of the world is at least NP-hard, as shown by Krause and

Algorithm 2 Compress the graph such that it has at most n laser scans.

Input: set of laser scans $\mathbf{Z}$, maximum number of scans n
Output: set of laser scans $\mathbf{Z}$ such that $|\mathbf{Z}| \leq n$

```
while |Z| > n do
  for all laser scans Z_i ∈ Z do
    IG_{Z_i} ← I(M; Z) − I(M; Z \ {Z_i})        // see Equation (3.11) and Algorithm 1
  end for
  Z̃ ← argmin_{Z_i} IG_{Z_i}
  Z ← Z \ {Z̃}                                  // discard laser scan Z̃
  if desired, marginalize out node corresponding to Z̃
end while
```

Algorithm 3 Discard laser scans less informative than ε.

Input: set of laser scans $\mathbf{Z}$, threshold ε
Output: set of laser scans $\mathbf{Z}$ such that the information gain of each scan is at least ε

```
repeat
  for all laser scans Z_i ∈ Z do
    IG_{Z_i} ← I(M; Z) − I(M; Z \ {Z_i})        // see Equation (3.11) and Algorithm 1
  end for
  Z̃ ← argmin_{Z_i} IG_{Z_i}
  if IG_{Z̃} < ε then
    Z ← Z \ {Z̃}                                // discard laser scan Z̃
    if desired, marginalize out node corresponding to Z̃
  end if
until IG_{Z̃} ≥ ε
```

Guestrin [97]. Fortunately, as they pointed out, the problem is submodular. Hence, greedily selecting measurements results in obtaining a set of measurements that is at most a constant factor, namely $1 - e^{-1} \approx 0.63$, worse than the optimal set. Motivated by this insight, our approach estimates the subset $\mathbf{Z}^\star$ by successively discarding laser scans. In each iteration, our method discards the laser scan that is expected to be least informative.

Our approach is able to restrict the size of the graph, resulting in an any-space SLAM system. Alternatively, our approach is capable of discarding uninformative laser scans by setting a threshold on the expected information gain of laser scans. Algorithm 2 and Algorithm 3 sketch pseudocode of our greedy approach to discarding laser scans during online operation. In practice, most of the mutual information values $I(C; \mathbf{Z})$ do not change when incorporating new laser scans. Therefore, an efficient implementation of the algorithm can cache these values and does not need to recompute them in each step. The implementation then only needs to recompute $I(C; \mathbf{Z})$ for cells that are in the measurement range of laser scans that are incorporated into the belief or discarded from the belief of the robot.

Figure 3.5: Mobile robot ActivMedia Pioneer.

3.5 Experimental Evaluation

We carried out a set of experiments using an ActivMedia Pioneer mobile robot equipped with a SICK laser range finder, which is depicted in Figure 3.5. In addition to that, we applied our method to a set of popular benchmark datasets that are publicly available. We specifically compare the performance of our SLAM implementation when using our method to discard laser scans with the performance of the same SLAM implementation when no scans are discarded. We refer to the latter as the "standard approach". We evaluate our approach using four distinct datasets. First, we use a dataset that we recorded with the Pioneer mobile robot traveling in our lab environment in building 079 on the campus of the University of Freiburg for an extended period of time. In addition to that, we use a dataset of a robot moving in building 101 at the University of Freiburg, which was provided by Grisetti et al. [61]. Furthermore, we use the FHW dataset and the Intel Research Lab dataset, which were both provided by Dirk Hähnel.

3.5.1 Memory Requirements and Runtime

Our approach to selecting informative measurements causes an overhead in computation time that only pays off when the robot frequently traverses previously visited parts of the environment. More specifically, when the robot stays in the same environment and rarely explores new terrain, the loop closing component of the SLAM front end needs to consider an ever increasing number of laser scans in each step. The reason for this is that the front end performs scan matching to find constraints between the current scan and all former scans that were recorded in the vicinity of the robot. Furthermore, the runtime of graph optimization scales with the number of edges, which tends to increase quadratically when

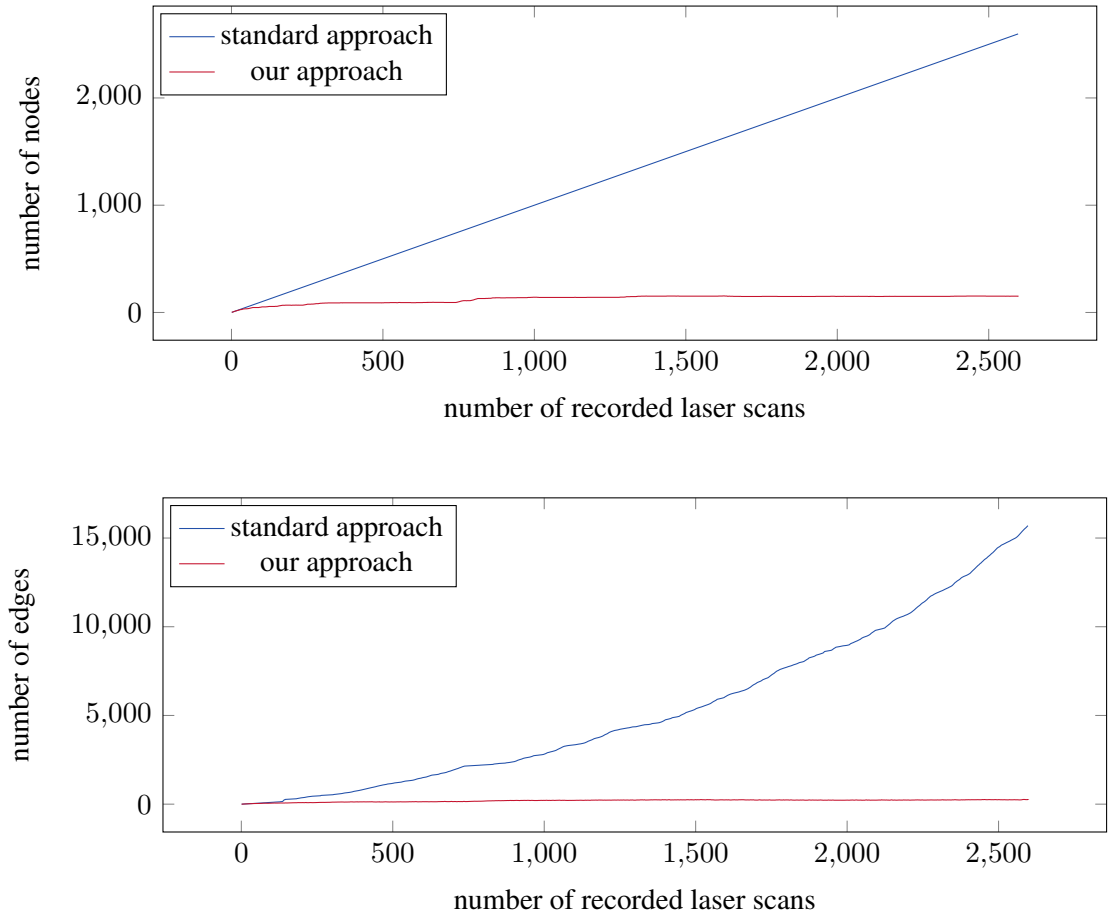


Figure 3.6: Evolution of the size of the graph while the robot moves through our office environment for an extended period of time. In this experiment, our algorithm discarded all laser scans whose expected information gain was below a given threshold.

traversing previously observed areas. In contrast to the standard approach, our approach discards redundant laser measurements, and, as a result, the number of nodes in the graph stays small. In this section, we evaluate the memory requirements of our approach in terms of the size of the graph. We furthermore provide runtime measurements of our SLAM system running on a 2.6 GHz Intel CPU system with 8 GB RAM. Note, however, that our current implementation is not optimized for speed. We expect a significant reduction in runtime when caching intermediate results, as discussed in Section 3.4.

Figure 3.6 shows the evolution of the number of nodes and edges in the graph while the robot moves through our office environment for an extended period of time. The plots clearly suggest that the experimental setup leads to an explosion in terms of memory requirements when relying on the standard approach. In contrast to that, when using our method, the graph does not grow as long as the robot does not explore new terrain. More specifically, in this experiment, our approach kept only 148 of 2597 laser scans. The standard approach required a total of 129 min to process the dataset. Despite the overhead in computation time, the same SLAM implementation had successfully computed a map

	number of scans		
dataset	standard approach	our approach	changed grid cells
Intel Research Lab	1802	349	0.9 %
Freiburg, building 079	2597	148	1.6 %
FHW Museum	2049	250	1.2 %

Table 3.1: Effects on the most likely states of the grid cells when discarding laser measurements using our approach. Even though our method discards most of the laser measurements obtained by the mobile robot, only a small fraction of the grid cells changes their most likely state due to our method.

after only 87 min when taking advantage of our graph compression technique.

To process the FHW dataset, which is depicted in Figure 3.1, the standard approach required 32 min, whereas our approach needed 17 min. Our approach retained 250 of 2049 laser scans and 324 of 7190 edges.

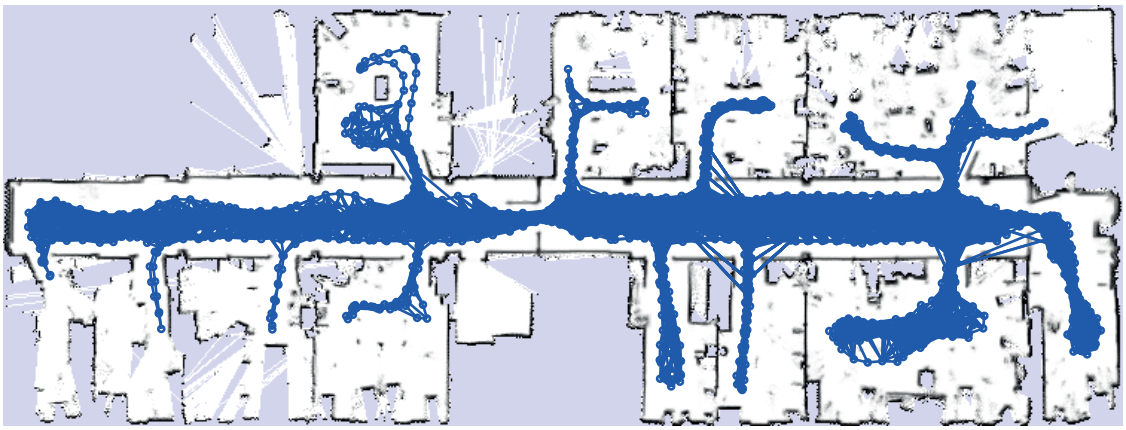
For the Intel Research Lab dataset, which is depicted in Figure 3.9, the standard approach was done after 9 min, whereas our approach needed 26 min. Our approach kept 250 of 1,802 laser scans and 349 of 3,916 edges. In this case, the computation time overhead of our method did not pay off because the robot mostly explores new terrain and only revisits the relatively small corridor.

3.5.2 Effects on the Most Likely Grid Map

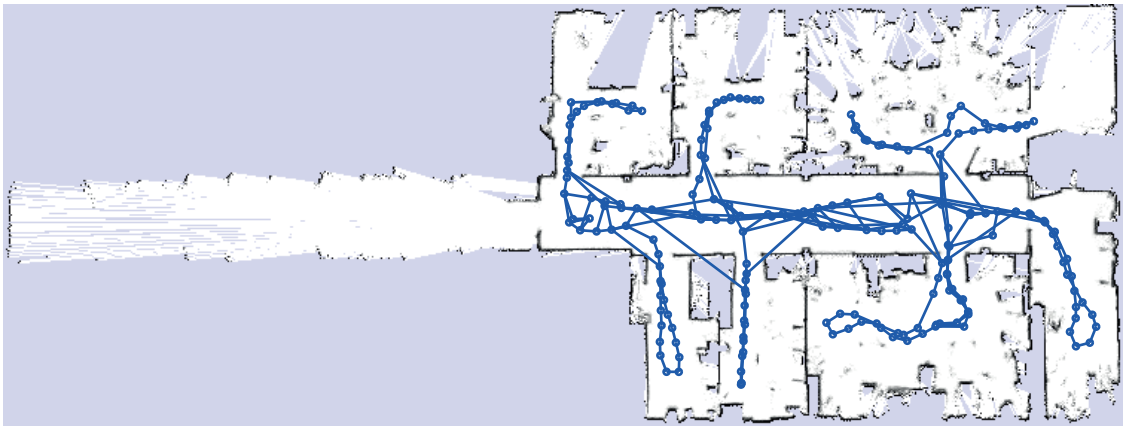
We furthermore analyzed the effects of our approach on the resulting occupancy grid maps. We therefore compared the grid maps at a resolution of 10 cm and counted the number of cells that changed their most likely state, as in "free", "occupied", and "unknown", due to our method. More specifically, we evaluated the grid maps using three different datasets. Table 3.1 summarizes the results. Although our approach retained only 349 of 1802 laser scans of the Intel Research Lab dataset, only 0.9 % of the cells changed their most likely state. In the long-term experiment conducted in building 079 at the University of Freiburg, our method kept 148 of 2597 laser scans, and 1.6 % of the cells changed their most likely state. When mapping the FHW, our approach maintained 250 of 2049 scans, and 1.2 % of the cells changed their most likely state. These results suggest that the changes in the most likely maps are small. Most of the cells that changed their most likely state were seen as free before applying our method and are seen as unknown after applying our technique because no beam covers the cells anymore.

3.5.3 Measurement Model

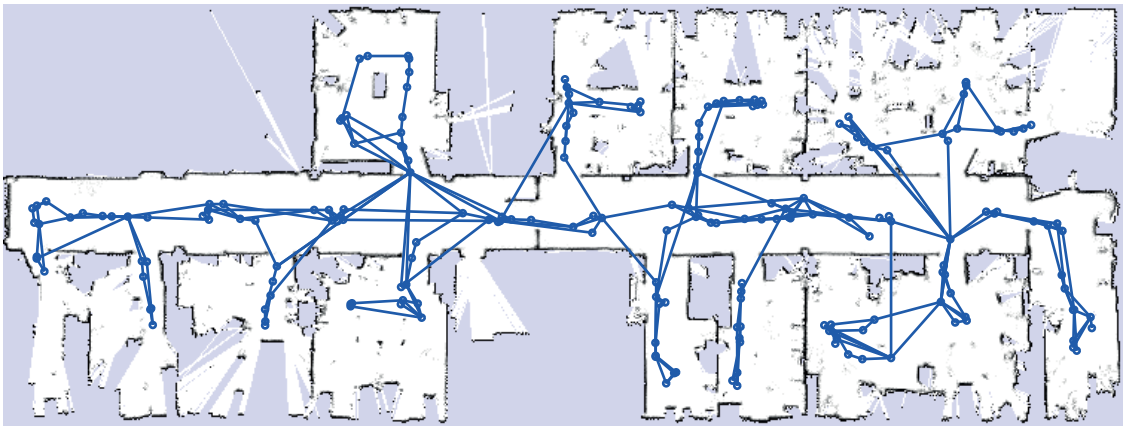
We briefly analyzed the effects on the compressed graph when using different measurement models, i. e., different models for describing $p(z_i^j)$. To this end, we performed



(a) Standard approach. 2597 laser scans, 15 695 edges.



(b) Our approach at an intermediate time step. 200 laser scans, 264 edges.



(c) Our approach. 200 laser scans, 315 edges.

Figure 3.7: University of Freiburg, building 079. A mobile robot moves around in an office environment for an extended period of time, visiting the rooms and the corridor over and over again. The map resulting from the 200 laser scans that our approach has chosen is fairly similar to the map resulting from all 2597 laser scans that the robot obtained throughout the entire experiment.

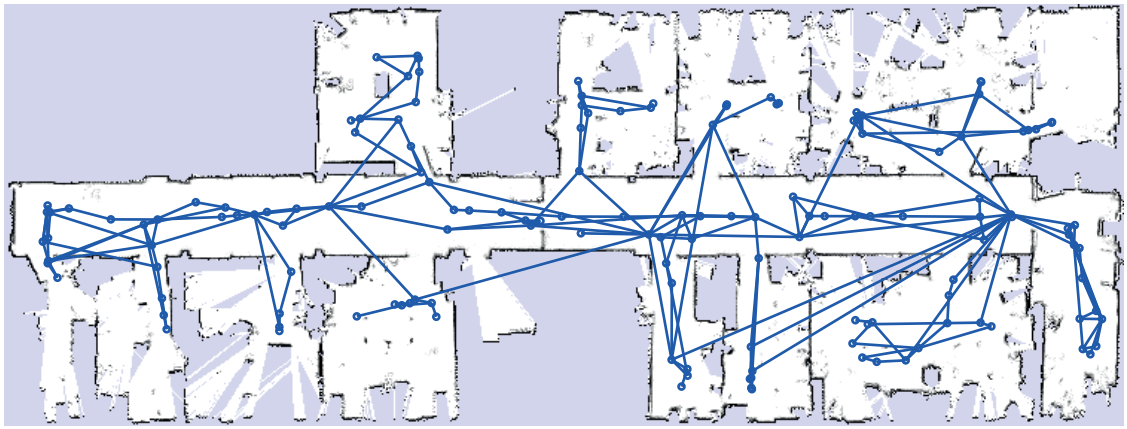
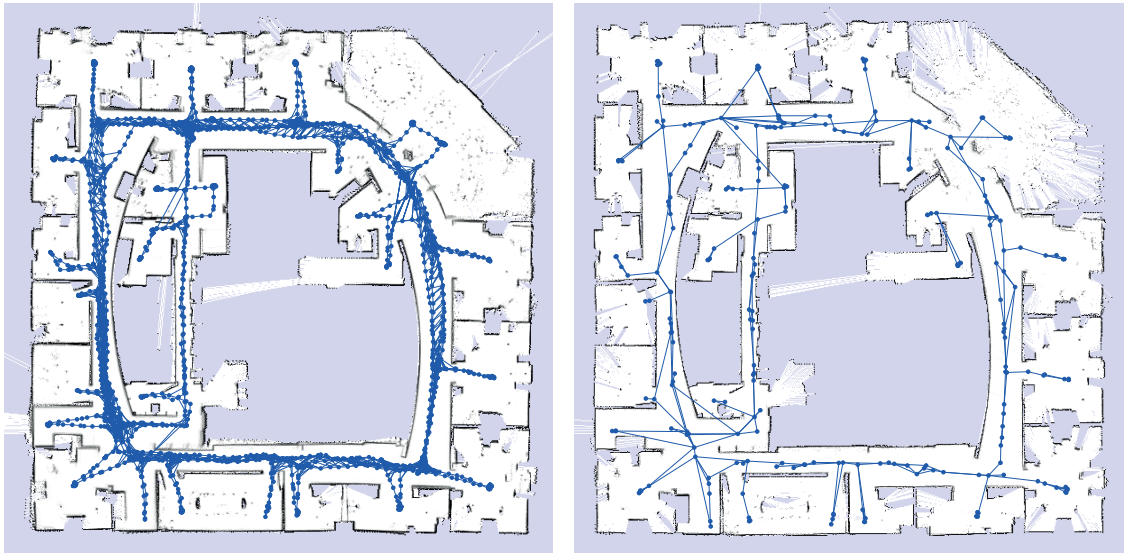
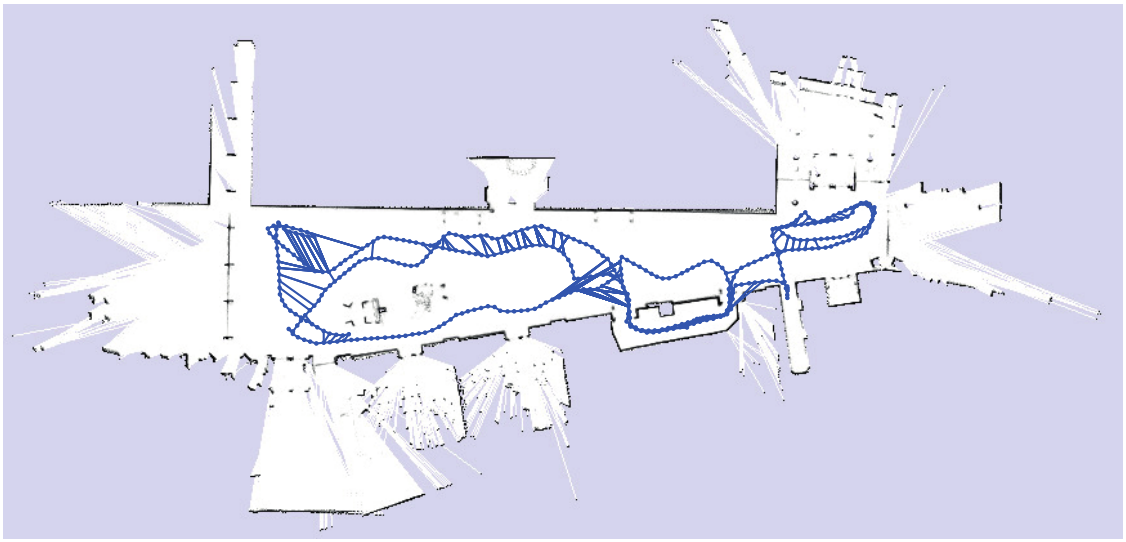


Figure 3.8: University of Freiburg, building 079. In this experiment, we set a threshold for the mutual information between the map and the laser scans. As a result, our approach only keeps 148 of the 2597 laser scans, yet the resulting map is fairly similar to the map computed based on all laser scans.

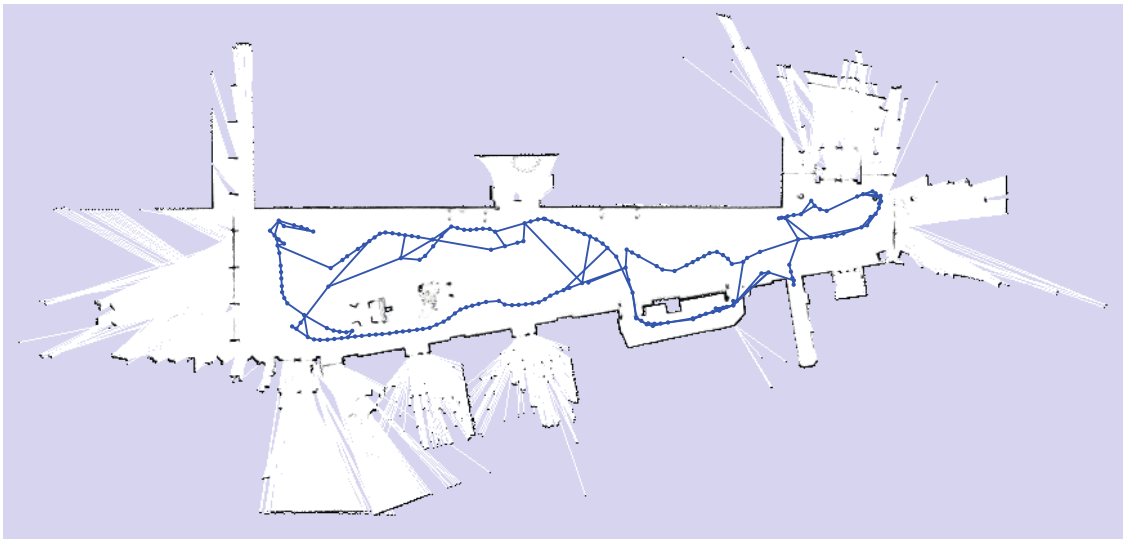


(a) Standard approach. **(b)** Our approach.

Figure 3.9: Intel Research Lab. The standard approach incorporates 1802 laser scans and 3916 edges. Our approach only retains 250 laser scans and 349 edges, yet the resulting maps are fairly similar.



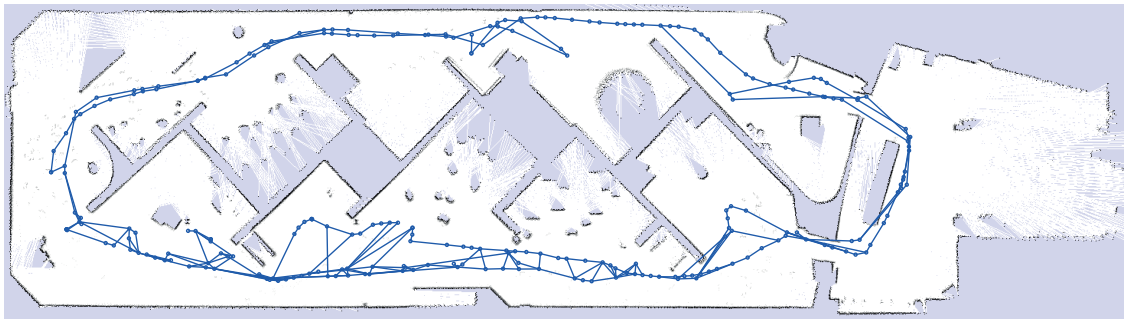
(a) Standard approach. 408 nodes, 723 edges.



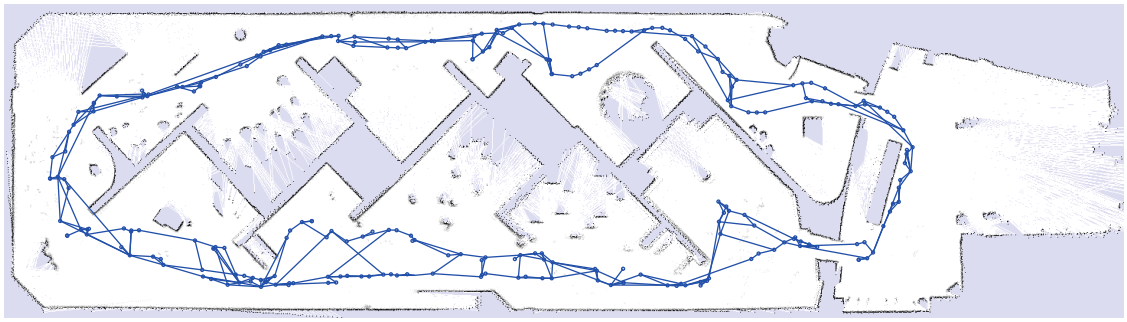
(b) Our approach. 200 nodes, 246 edges.

Figure 3.10: University of Freiburg, building 101. Since the robot does not frequently traverse previously visited terrain, our approach only discards few scans in this experiment.

the same mapping experiment using the exponential probability distribution given in Equation (3.7) as well as using the measurement model depicted in Figure 3.3, which was learned from robotics datasets. Since the exponential model and the learned model are fairly similar for ranges beyond 1.5 m, we expected that the effect on the resulting grid map and the graph is small. In fact, as can be seen in Figure 3.11, different nodes have been selected, but the overall structure of both graphs is similar in terms of topology and local density of the nodes. The same applies to the resulting occupancy grid maps. Hence, our experiments do not suggest that either method perform better.



(a) Measurement model based on the exponential probability distribution given in Equation (3.7)



(b) Measurement model learned from robotic datasets, which is plotted in Figure 3.3.

Figure 3.11: Maps and graphs obtained by using different measurement models, which are given by $p(z_i^j)$, when processing the FHW dataset. The resulting maps and the node densities are fairly similar in both settings. However, some areas of the environment are not fully explained by these maps. The states of the corresponding grid cells are unknown according to the map. This suggests that more laser scans might be needed to fully capture the environment in this experiment.

3.5.4 Practical Considerations

In this section, we discuss the effects of discarding laser measurements while mapping on the quality of the resulting grid maps. In principle, the more measurements are taken into account, the better is the estimate of the state of the world. Ignoring measurements leads to a belief of higher uncertainty. However, today's occupancy grid-based mapping approaches typically rely on some kind of scan alignment technique, which aligns the scans to extract spatial constraints. It turns out that these systems have disadvantages when it comes to long-term mapping. Whenever the robot obtains a laser measurement, the scan matcher aims at aligning the new scan with existing scans in order to solve the data association problem. The probability that the scan matcher thereby makes a small alignment error is nonzero. A scan that is incorporated at a slightly wrong position blurs the map. As a result, the probability that the scan alignment procedure poorly aligns subsequent scans increases since matching is performed based on misaligned scans. Hence, the probability of making alignment errors increases with the number of incorporated scans. In the long run, the map tends to become increasingly blurred, and the mapping approach can diverge.

Figure 3.7 depicts the grid maps resulting from the long-term experiment that we conducted in our office environment. The grid map generated by the standard approach that incorporated all the laser measurements exhibits more blur in some parts of the map, which can be attributed to errors in the scan alignment procedure as described above. In general, the more often the robot traverses already visited terrain, the more blur is added to the maps. In contrast to the standard approach, by incorporating fewer laser measurements into the grid map, our method produces maps with less blurred obstacle boundaries, particularly when the robot frequently traverses already visited places. On the one hand, however, these maps that take into account fewer measurements do not represent better estimates of the world, strictly speaking. In practice, on the other hand, it turns out that our method facilitates scan matching and therefore can mitigate the risk of divergence in the mapping process.

3.6 Related Work

The pivotal importance of maps to mobile robots has given rise to a large number of mapping techniques. Lu and Milios [120] were the first to model the belief of the robot in terms of a graph in order to perform global map optimization. Since then, the robotics community has presented a large variety of methods to minimize the error in graphs. Popular approaches include but are not limited to the methods proposed by Kaess et al. [79], Grisetti et al. [60], Kümmerle et al. [110], Kaess et al. [80], Olson and Agarwal [139], and Agarwal et al. [5].

Most of the approaches to SLAM, however, do not provide means to effectively prune the graph. Instead, the methods tend to add more and more nodes to the graph as the robot obtains new measurements. Consequently, the runtime complexity per timestep increases with the number of measurements that the robot has obtained. There are approaches to address the strictly increasing complexity. For instance, Folkesson et al. [51] group nodes into rigid local submaps. Bosse et al. [23] subdivide the map into connected frames that contain maps that capture the local environment. These methods, however, typically do not discard nodes that store information about the environment. As a result, the above-mentioned techniques do not effectively prevent the graph from growing.

An effective approach to reducing the number of nodes in the graph is to sample the trajectory of the robot at an appropriate spatial decimation, as proposed by Eustice et al. [48]. Similarly, Konolige and Agrawal [95] only add a new node to the graph if it is not spatially close to any existing node. Konolige and Bowman [96] present an approach to lifelong mapping that only updates the map using a stereo camera when the environment changes. Their method discards views based on a least-recently-used policy. Similarly, hierarchical techniques, such as the techniques proposed by Grisetti et al. [60], Estrada et al. [47], and Ni et al. [136], seek to bound the computational requirements by optimizing only higher levels of the hierarchy. However, the above-mentioned techniques do not rely on information-theoretic concepts to determine which sensor measurements to discard from the belief in order to free up resources.

In contrast to that, Davison [40] analyzes mutual information, particularly in the case

of Gaussian probability distributions, to guide image processing. In the vision community, Snavely et al. [154] are interested in finding a skeletal subgraph with the minimum number of interior nodes that spans the full graph while achieving a bound on the full covariance. Their technique is used for reconstructing scenes based on large, redundant photo collections.

Kaess and Dellaert [78] consider the information gain of measurements in the state estimate within the iSAM framework. In contrast to that, our approach estimates the mutual information between laser range scans and the occupancy grid map in order to explicitly reason about the effect of the measurements on the resulting grid map. Ila et al. [75] propose to only incorporate non-redundant poses and informative constraints based on the relative distance between poses in information space and the expected information gain of candidate loop closures. As opposed to our maximum-likelihood approach to SLAM based on graphs, their method applies an information filter and does not marginalize out poses that were already added. Eade et al. [45] presented a view-based monocular SLAM system that reduces the complexity of the graph by marginalization and subsequent suppression of edges incident to nodes of high degrees. Their heuristic discards the constraints that most agree with the current state estimate, which, however, introduces a bias into the system.

Kim and Eustice [84] combine information gain with visual saliency in order to find loop closures. Bachrach et al. [13], Kollar and Roy [92], and Stachniss et al. [157] use information gain to evaluate target locations during exploration. He et al. [67] are interested in planning robot paths in a way that minimizes localization uncertainty in the context of a quadrotor helicopter.

When discarding laser range scans, the pose nodes that correspond to the discarded laser measurements can be marginalized out. Exact marginalization, however, results in a dense graph, which hampers efficient mapping. Kretzschmar et al. [101] therefore rely on Chow-Liu tree approximation [33] to sparsify the Markov blanket of each marginalized pose node. Subsequently, Carlevaris-Bianco and Eustice [29, 30] propose generic linear constraints to approximate the information matrix of the Markov blanket. Mazuran et al. [129] pose a similar optimization problem to approximate the original distribution with a set of nonlinear measurements. Other researchers within the field of robotics use Chow-Liu trees to approximate multidimensional probability distributions. For instance, Cummins and Newman [38] extend the bag-of-words approach by learning a model for the sensory data in the form of a Chow-Liu tree. Furthermore, Chli and Davison [32] use a Chow-Liu tree approximation to divide large visual maps into a fully hierarchical correlation and clustering structure.

3.7 Conclusion

We presented a novel approach to pruning the graph in simultaneous localization and mapping that enables efficient long-term mobile robot mapping by discarding measurements. Our method discards laser range measurements in a way that minimizes the loss of information about the environment. This represents a more principled approach

to discarding laser measurements than existing methods that discard all but every nth measurement or that aim at equally distributing the measurements in the environment. Maximizing the mutual information between the grid map and the laser measurements, however, is infeasible in general. We demonstrate how to overcome this by expressing the entropy of the grid map in terms of the effects of the laser measurements on the grid cells.

An extensive set of experiments suggests that our approach leads to maps that are highly similar to the maps that result from integrating all laser measurements in the mapping process, even when our method discards most the laser measurements that the mobile robot obtained. The experiments furthermore demonstrate that our approach reduces the complexity of mobile robot mapping in settings where the robot spends most of the time moving in previously explored areas. In fact, the complexity of our method scales with the area of the explored environment, which is in contrast to standard graph-based approaches to simultaneous localization and mapping, which scale with the length of the trajectory. The work presented in this chapter can therefore be considered to be a step towards long-term mobile robot mapping. The method, however, assumes that the environment is static. In future work, it would be interesting to investigate how to maintain a map of a changing environment.

Maps of the environment play a prominent role throughout this thesis owing to their importance to mobile robot navigation. The next chapter will present a method that helps a mobile robot build a consistent map by deploying artificial landmarks in the environment, which is particularly useful in ambiguous environments. In Chapter 5, the robot will use an annotated map of the environment to give natural and intuitive directions to people unfamiliar with the environment. In Chapter 6 and Chapter 7, the robot will use a map of the environment, which can be generated by the approach presented in this chapter, for autonomous navigation.

Chapter 4

Learning to Deploy Landmarks to Foster Data Association in SLAM

Approaches to simultaneous localization and mapping, such as the method introduced in the previous chapter, require that the robot be able to solve the data association problem while mapping the environment. Finding the correct correspondences between sensor measurements and environment features, however, can be challenging, especially in ambiguous environments. A robot can mitigate the data association problem by deploying uniquely identifiable artificial landmarks in the environment and by using these landmarks as fixed anchors in order to resolve ambiguities. We use Monte Carlo reinforcement learning to compute a landmark deployment policy that optimizes data association performance. Our experimental evaluation in simulation and on a real mobile robot suggests that our approach leads to significantly more accurate maps. We furthermore demonstrate that our method significantly outperforms baseline landmark deployment strategies.

The last chapter developed a method for discarding sensor measurements during simultaneous localization and mapping in order to reduce redundancy in the model, thereby freeing memory for coming measurements. To build a consistent map of the environment that is suitable for navigation, however, the robot needs to be able to recognize previously observed places based on its sensor measurements. The complexity of this data association problem grows exponentially with the number of feature observations. Finding data associations can therefore become intractable in large, ambiguous environments, even for state-of-the-art techniques.

One way to mitigate the data association problem in practice is to have the robot deploy artificial landmarks in the environment. When the robot returns to a place where it has deployed such a landmark, the robot can recognize the landmark and use it as a fixed anchor to resolve ambiguities in the environment, which is known as closing the loop. This approach typically relies on robots that can drop radio-frequency identification (RFID) tags or similar uniquely identifiable landmarks. The question arises as to when the robot should deploy these landmarks while mapping. Most state-of-the-art techniques rely

on handcrafted heuristics to decide when to deploy the landmarks. These heuristics, however, often lead to suboptimal choices that may not help the robot find the correct data associations.

In this chapter, we propose a novel method that allows the robot to learn when to place its artificial landmarks in the environment during navigation in order to optimize data association performance. Our approach is designed to foster data associations without interfering with the navigation task of the robot. The robot therefore does not need to make any detours for proper landmark deployment. We compute an optimal landmark deployment policy by applying actor-critic Monte Carlo reinforcement learning. To this end, we simulate episodes of robot navigation tasks and evaluate the data associations that the robot suggests. The reinforcement learning approach optimizes a numerical reward that captures the negative number of incorrectly estimated feature correspondences. In order to obtain a policy that generalizes to new environments, our approach relies on features, such as the remaining battery life time, the number of landmarks left on board, the distance to the closest deployed landmark, and a feature capturing the abstract local structure of the environment. Our method allows the robot to learn a policy for placing artificial landmarks that fosters data association performance. See Figure 4.1 for an example of a simulated mobile robot mapping a Manhattan-like world.

4.1 Data Association in Simultaneous Localization and Mapping

Simultaneous localization and mapping can be cast as the problem of estimating the joint posterior probability distribution

$$p(\mathbf{x}_{1:T}, \mathbf{m}_{1:n}, c_{1:T} \mid \mathbf{u}_{1:T}, \mathbf{z}_{1:T}) \tag{4.1}$$

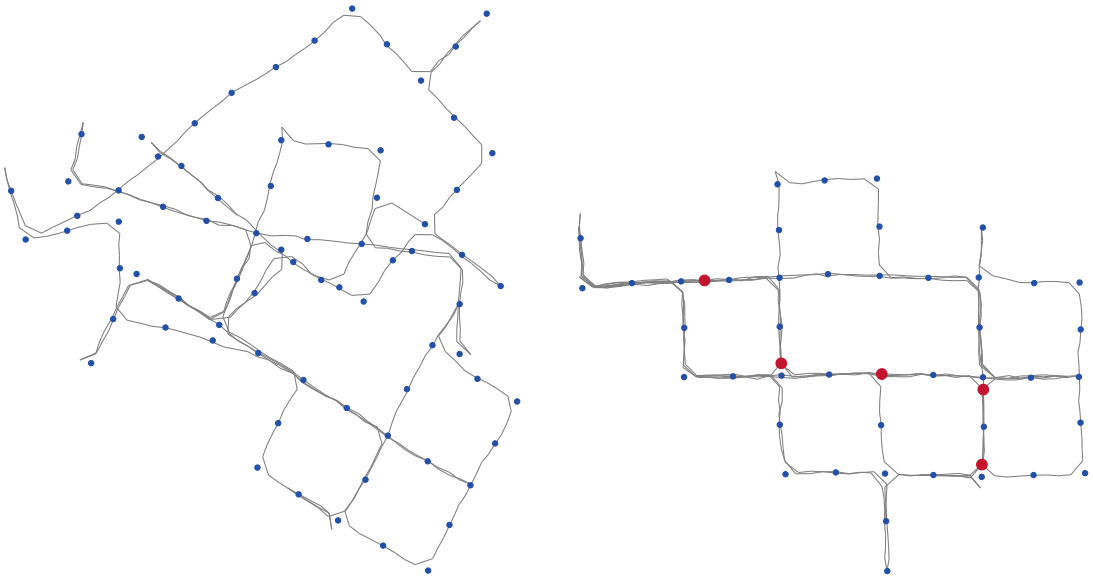
of the robot's poses $\mathbf{x}_{1:T}$ and the map $\mathbf{m}$ given a sequence of robot motion commands $\mathbf{u}_{1:T}$ and feature observations $\mathbf{z}_{1:T}$, where the map is given by the map features $\mathbf{m}_{1:n}$. In many applications, however, the features in the environment are not clearly distinguishable from each other. In these cases, a key challenge is to correctly estimate the data associations $c_{1:T}$ that identify the map features that are perceived in the observations, where

$$c_t(\mathbf{z}_t) = \mathbf{m}_i \tag{4.2}$$

indicates that measurement $\mathbf{z}_t$ is an observation of map feature $\mathbf{m}_i$ according to data association c_t. Integrating out the unknown data associations $c_{1:T}$ leads to the posterior distribution

$$p(\mathbf{x}_{1:T}, \mathbf{m}_{1:n} \mid \mathbf{u}_{1:T}, \mathbf{z}_{1:T}) = \sum_{c_1} \sum_{c_2} \dots \sum_{c_T} p(\mathbf{x}_{1:T}, \mathbf{m}_{1:n}, c_{1:T} \mid \mathbf{u}_{1:T}, \mathbf{z}_{1:T}). \tag{4.3}$$

Hence, the number of possible data associations grows exponentially with the number of feature observations, which is why solving the data association problem is challenging in



(a) No artificial landmarks. **(b)** Artificial landmarks.

Figure 4.1: A simulated robot travels through a Manhattan-like world. The figures depict the positions of the landmarks in the environment, here shown in blue, as estimated by an approach to simultaneous localization and mapping that relies on a gated nearest neighbor filter to compute the data associations. (a): The robot only relies on landmarks that are already in the environment to address the data association problem. The resulting wrong data associations lead to an inconsistent map that is not suitable for mobile robot navigation. (b): The robot deploys additional uniquely identifiable artificial landmarks in the environment, here shown in red, in order to mitigate the data association problem. As a result, the robot is able to compute a consistent map of the environment.

general, even for state-of-the-art methods. It is, however, essential to correctly compute the data associations, since a single incorrect data association may distort the entire map estimate, leading to an unusable map.

Most approaches to solving the data association problem evaluate the compatibility between sensor measurements and map features. To this end, the methods consider the difference between a measurement $\mathbf{z}_i$ that the robot has obtained and the predicted measurement $\hat{\mathbf{z}}_i$ given a specific data association $c(\mathbf{z}_i)$. This difference is commonly known as the innovation. For a Gaussian innovation, the squared Mahalanobis distance $D_M^2(\mathbf{z}_i, \hat{\mathbf{z}}_i)$ is distributed according to the χ_d^2 distribution, where d is the dimensionality of the innovation. The widely-used gated nearest neighbor filter [14] computes a set of compatible candidate features whose innovation is within a certain region of the χ^2 distribution. The filter then chooses the candidate feature that best matches the measurement or initializes a new map feature if there are no candidate features.

4.2 Deploying Landmarks to Foster Data Association

In this chapter, we consider a mobile robot that is able to deploy a limited number k of uniquely identifiable landmarks in the environment during navigation in order to mitigate the data association problem in simultaneous localization and mapping. Observations of such artificial landmarks that the robot has previously deployed in the environment in conjunction with the known data associations of these observations refine the posterior distribution given by Equation (4.3), leading to the posterior

$$p(\mathbf{x}_{1:T}, \mathbf{m}_{1:n}, c_{1:T}, \boldsymbol{\ell}_{1:k} \mid \mathbf{u}_{1:T}, \mathbf{z}_{1:T}, \mathbf{z}^{\boldsymbol{\ell}}_{1:T}, c^{\boldsymbol{\ell}}_{1:T}), \tag{4.4}$$

where $\boldsymbol{\ell}_{1:k}$ are the positions of the artificial landmarks, and $c^{\boldsymbol{\ell}}_{1:T}$ refer to the known identities of the artificial landmarks observed in the measurements $\mathbf{z}^{\boldsymbol{\ell}}_{1:T}$. In particular, since the robot can uniquely identify the artificial landmarks, the robot can resolve ambiguities in the environment by using observations of these landmarks as anchors when returning to previously observed places. Hence, the artificial landmarks potentially help the robot solve the data association problem in simultaneous localization and mapping.

4.3 Evaluating Data Associations

The goal of our method is to have the robot deploy a limited number of artificial landmarks in the environment in a way that optimizes the performance of the data association algorithm. To this end, we evaluate the data association performance in terms of the number of incorrect data associations. A data association algorithm either assigns a feature observation to an existing feature in the map, or the algorithm creates a new map feature when the feature observation cannot be associated with an existing map feature. It is important to distinguish between environment features that are present in the environment and the map features that the algorithm maintains in its map. We denote the environment features by $\mathbf{m}^{\star}_{1:n^{\star}}$ and the map features as $\mathbf{m}_{1:n}$. In addition to that, let $c^{\star}_t$ be the true data association that indicates that observation $\mathbf{z}_t$ corresponds to environment feature $\mathbf{m}^{\star}_i$. In contrast to that, let c_t be the data association as estimated by the algorithm that indicates that observation $\mathbf{z}_t$ originates from map feature $\mathbf{m}_j$. For every environment feature $\mathbf{m}^{\star}_i$, we compute the number $N(\mathbf{m}^{\star}_i)$ of map features $\mathbf{m}_j$ that the data association algorithm associated with the environment feature $m^{\star}_i$ at least once. More formally, we define

$$N(\mathbf{m}^{\star}_i) := \Big| \big\{ \mathbf{m}_j \in \mathbf{m}_{1:n} \mid \exists t \in [1, T] : c^{\star}_t(\mathbf{z}_t) = \mathbf{m}^{\star}_i \wedge c_t(\mathbf{z}_t) = \mathbf{m}_j \big\} \Big|. \tag{4.5}$$

In case the estimated data associations are correct, all observations of the same environment feature are associated with the same map feature, and we obtain $N(\mathbf{m}^{\star}_i) = 1$. The total number of incorrectly estimated feature correspondences is then given by

$$E(c^{\star}_{1:T}, c_{1:T}) := \sum_{i=1}^{n^{\star}} \Big(N(\mathbf{m}^{\star}_i) - 1 \Big). \tag{4.6}$$

Note that this function counts wrong data associations that occur when the algorithm associates multiple observations of the same environment feature to different map features. Data association errors that occur when the algorithm associates every observation of different environment features with the same map feature, however, are not captured by the function, yet our experimental evaluation demonstrates that policies that minimize the function lead to maps that are significantly more accurate than the maps that result from using baseline policies.

4.4 Deploying Landmarks as a Reinforcement Learning Problem

The benefit of having deployed an artificial landmark at a certain location in the environment depends on to what extent the robot can use that landmark later on during navigation to refine its pose estimate. Hence, it is desirable that the robot deploy the artificial landmarks such that they foster data associations. Our goal, however, is to deploy the artificial landmarks in a way that does not interfere with the navigation task of the robot, for example by having the robot make a detour to place a landmark at a specific location.

We use reinforcement learning to compute a landmark deployment policy that optimizes the performance of data association. In reinforcement learning [160], an agent interacts with its environment to learn how to behave so as to maximize a numerical reward. We formulate this reinforcement learning environment in terms of a discrete set $\mathcal{S}$ of states $s \in \mathcal{S}$, a discrete set $\mathcal{A}$ of actions $a \in \mathcal{A}$, a transition model that describes how the state changes when carrying out an action, and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that determines the numerical reward that the agent receives for executing action $a \in \mathcal{A}$ when in state $s \in \mathcal{S}$. The agent behaves according to a stochastic policy π, which is given by a probability distribution $\pi(s, a)$ of choosing action a when in state s. An episode consists of a finite sequence of T time steps. The return $R(s_t, a_t)$ of executing action a_t in state s_t at time t is given by the sum of the rewards gathered at the individual time steps during one episode after having executed that action a_t, giving

$$R(s_t, a_t) := \sum_{t'=t+1}^{T} r(s_{t'}, a_{t'}). \tag{4.7}$$

Reinforcement learning aims to find a policy $\pi^\star$ that maximizes the expected return

$$Q^\pi(s, a) := \mathbb{E}_\pi[R(s_t, a_t) \mid s_t = s, a_t = a] \tag{4.8}$$

for all states $s \in \mathcal{S}$ and all actions $a \in \mathcal{A}$.

In each episode during the learning phase, the actor performs a randomly sampled, simulated navigation task in the environment. At each time step t of an episode, the agent decides whether to deploy one of its artificial landmarks in the current state s_t

or not according to the current policy $\pi(s_t, a_t)$. In our case, the action space $\mathcal{A}$ of the reinforcement learning environment is therefore given by

$$\mathcal{A} = \{\text{deploy}, \text{keep}\}. \tag{4.9}$$

The robot relies on a graph-based approach to simultaneous localization and mapping to compute a map of the environment based on features that are in the environment and based on the artificial landmarks. At each time step t, the actor receives the reward

$$r(s_t, a_t) := \begin{cases} 0 & \text{if } t < T \\ -E(c^\star_{1:T}, c_{1:T}) & \text{if } t = T, \end{cases} \tag{4.10}$$

where $E(c^\star_{1:T}, c_{1:T})$ is defined in Equation (4.6). According to Equation (4.7), the rewards then lead to the returns

$$R(s_t, a_t) = \sum_{t'=t+1}^{T} r(s_{t'}, a_{t'}) = -E(c^\star_{1:T}, c_{1:T}) \tag{4.11}$$

for all state-action pairs (s_t, a_t) that occur during the episode.

4.4.1 Features

In order to learn policies that generalize to new environments, we represent the state of the robot in the environment in terms of features. These features need to capture all the information that is relevant for deciding whether to deploy an artificial landmark or not. One of our features therefore captures the remaining battery life time in percent. Another feature counts the artificial landmarks that can still be deployed by the robot. Yet another feature captures the distance to the artificial landmark that has been deployed closest to the robot. In addition to that, we use a spatial feature that captures the structure of the environment in the vicinity of the robot. To this end, we classify the location of the robot into the categories “room”, “doorway”, “hallway”, and “junction” based on laser range scans obtained by the robot. Robust approaches to this classification task have been presented by Friedman et al. [54] and Martínez Mozos et al. [128] for laser range scans and by Luo et al. [123] for vision-based systems. To efficiently represent the state-action space, we divide the space into bins.

4.4.2 Actor-Critic Monte Carlo Reinforcement Learning

When the state transition model of the reinforcement learning environment is known, an optimal policy that optimizes the expected return can be estimated by means of dynamic programming [160]. When the state transition model, however, is unknown, an optimal policy can be estimated by means of Monte Carlo reinforcement learning [160]. To this end, Monte Carlo reinforcement learning methods estimate $Q^\pi(s, a)$ by averaging the return over a number of sample episodes. The first-visit-only Monte Carlo method thereby

only considers the first occurrence of each state-action pair (s, a) in each episode e. This leads to the estimator

$$\hat{Q}^{\pi}(s,a) := \frac{1}{|\mathcal{F}(s,a)|} \sum_{e \in \mathcal{F}(s,a)} R^{e}_{\text{first}}(s,a)\,, \tag{4.12}$$

where

$$\mathcal{F}(s,a) := \{e \mid (s,a) \in e\} \tag{4.13}$$

is the set of episodes that contains an occurrence of the state-action pair (s, a), and $R^{e}_{\text{first}}(s, a)$ is the return corresponding to the first occurrence of the state-action pair (s, a) in episode e. The estimator $\hat{Q}^{\pi}$ converges to the true Q-function Q^{π} if all state-action pairs occur with nonzero probability when choosing actions according to policy π. To satisfy this condition, we use a so-called softmax policy of the form

$$\pi(s,a) = \frac{\exp\left(\hat{Q}^{\pi}(s,a)/\tau\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\hat{Q}^{\pi}(s,a')/\tau\right)}, \tag{4.14}$$

where $\tau \in \mathbb{R}^{+}$ is known as the temperature, which determines how peaked the distribution is.

In order to estimate the optimal policy in an efficient way, we use an actor-critic approach to Monte Carlo reinforcement learning. In actor-critic reinforcement learning [160], the so-called actor behaves according to the current policy, while the so-called critic attempts to estimate the Q-function induced by this policy. As soon as the critic has observed enough episodes to learn the Q-function, the critic becomes the actor and a new critic is initialized. The advantage of actor-critic learning is that the Q-function can be estimated from a policy that does not change in the process.

4.5 Experimental Evaluation

We conducted experiments in simulation and using a real mobile robot to evaluate the performance of our approach. We consider a robot that travels through environments with indistinguishable features. The robot is able to carry and deploy five artificial landmarks. Furthermore, the robot is equipped with a noisy odometer and a sensor that provides noisy measurements of the relative positions of the indistinguishable environment features and the uniquely identifiable artificial landmarks. To build a map of the environment, the robot relies on a graph-based approach to simultaneous localization and mapping using the framework for graph optimization presented by Kümmerle et al. [110]. To address the data association problem, the robot uses a gated nearest neighbor filter to compute the data associations.

4.5.1 Mapping an Office Environment

To evaluate how our method performs on a real robot, we conducted an experiment with the ActivMedia Pioneer robot depicted in Figure 4.2. We had the robot build

Figure 4.2: Mobile robot ActivMedia Pioneer equipped with a device to deploy artificial landmarks, which, however, is not visible in this figure, and a SICK RFID reader to perceive these artificial landmarks and environment features.

a map of the office environment in building 079 at the University of Freiburg that is depicted in Figure 3.7(b). The robot was equipped with a custom-made device to deploy five RFID landmarks and a SICK RFI641 RFID reader, which was able to perceive the landmarks later on in a uniquely identifiable manner. The RFID reader had a circular field of view and a range of 0.9 m. In addition to that, the robot was equipped with a SICK S300 laser range finder with a field of view of 270°, which the robot used to compute the spatial features discussed in Section 4.4.1. To compute the spatial features, we relied on a straightforward heuristic that looks for local minima in the laser scans in order to detect door posts and that extracts straight lines in order to detect walls of the hallway. Using a more sophisticated technique, such as the methods proposed by Friedman et al. [54] or Martínez Mozos et al. [128], would possibly further improve our results. We placed 70 RFID tags at randomly selected locations in the environment, which served as environment features. The robot was able to perceive these features using its RFID reader. Although the reader was able to uniquely identify these landmarks, their true identities were hidden from the robot in order to simulate indistinguishable environment features. We rather used their identities in order to evaluate the performance of the robot in terms of the data association error E defined in Section 4.3.

We first learned a policy using our approach based on randomly generated, simulated navigation tasks. Our method simulated 2800 episodes, which took 37.48 min using our multithreaded implementation running on an Intel Core i7 system with 2.8 GHz.

We then evaluated the data association performance and the resulting maps when using the learned policy to deploy artificial landmarks. We compared the performance of our method with the performance when only relying on the indistinguishable environment

	error E	translational error	rotational error
our approach	19.30	0.88 m	0.09 rad
no artificial landmarks	23.10	3.68 m	0.24 rad

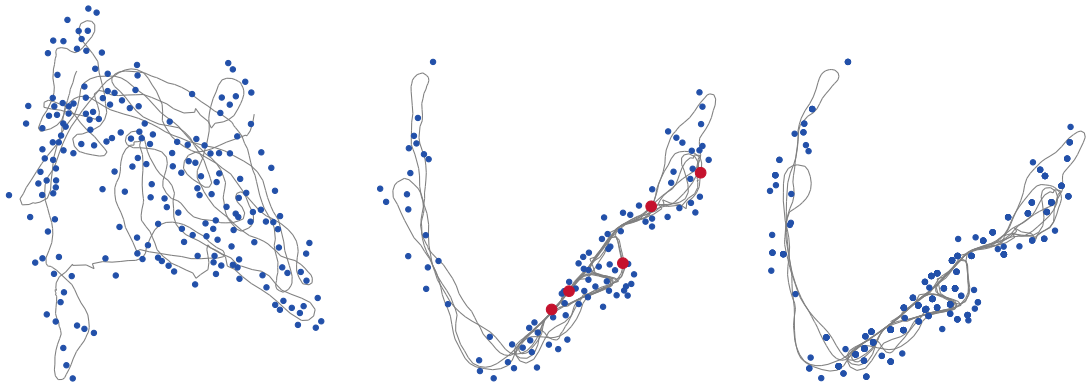
Table 4.1: An ActivMedia Pioneer mobile robot maps an office environment that comprises indistinguishable features. The table evaluates the effects of our method on the resulting estimated data associations and the resulting estimates of the robot's poses in terms of the error E, which is defined in Section 4.3, and the translational and rotational errors, proposed by Burgard et al. [26].

features. We evaluated the accuracy of the resulting estimates of the robot's poses by means of the framework presented by Burgard et al. [26], using laser-based Monte Carlo localization to obtain reference positions. Table 4.1 shows the results averaged over ten runs. Our method yields a moderate reduction in the error E, which translates to greatly improved estimates of the robot's poses in terms of translational error and rotational error.

4.5.2 Mapping a Large Outdoor Environment

In another experiment, we apply our approach to the well-known Victoria Park dataset. This dataset contains recorded sensor readings of a mobile robot traveling through Victoria Park in Sydney, which covers an area of more than 200 m $\times$ 200 m. During navigation, the robot observes the tree trunks that are present in the park. These tree trunks are indistinguishable to the robot. In this scenario, the robot does not rely on any spatial features. Note that owing to the nature of the data, we learn and evaluate the policy on the same run, i. e., the run recorded in the dataset, adding simulated observations of the deployed artificial landmarks on top of the observations recorded in the dataset. Figure 4.3 depicts the path of the robot as estimated by an approach to simultaneous localization and mapping that relies on a gated nearest neighbor filter for data association. The figure illustrates the effects of the artificial landmarks deployed by our policy. The figure shows the path of the robot as estimated when integrating and when not integrating the simulated observations of the deployed artificial landmarks. In addition to that, the figure depicts the path estimated when relying on the true data associations, which are publicly available. See Kümmerle et al. [110] for more information on the dataset.

The figure suggests that the gated nearest neighbor filter performs poorly at handling this scenario when not taking into account the deployed artificial landmarks. Using a more sophisticated data association approach, such as the joint compatibility branch and bound algorithm [134], certainly would increase the performance here. Our approach was able to reduce the number of incorrectly estimated feature correspondences E to 10, while the "equidistant" policy resulted in a value of 20, the "always" policy resulted in a value of 26, the "random" policy yielded a value of 78, and the "never" policy led to a value of 130. The "density" heuristic is not applicable because the obstacle density cannot be calculated in this scenario. The results suggest that our approach works well even in large outdoor environments without spatial features.



(a) No artificial landmarks. **(b)** Artificial landmarks. **(c)** Ground truth.

Figure 4.3: Victoria Park. The path of the robot and the locations of the environment features as estimated by an approach to simultaneous localization and mapping that relies on a gated nearest neighbor filter to address the data association problem. (a): The robot only uses the environment features to address the data association problem. The nearest neighbor filter fails to solve the data association problem and, as a consequence, the estimate diverges and the resulting map is not suitable for navigation. (b): The robot deploys artificial landmarks during navigation according to a policy learned by our approach. These landmarks help the robot solve the data association problem. (c): The robot is provided with the true data associations.

4.5.3 Comparison with Baseline Methods

In yet another experiment, we simulate a robot that maps a Manhattan-like world in order to compare the performance of our approach with the performance of five baseline methods. To perceive landmarks, the simulated robot is equipped with a noisy sensor that has a circular field of view and a range of 2 m. We annotated the map of the environment with the spatial feature values "corridor" and "junction". We compare our learned policies with four naïve baseline techniques. The approach "equidistant" deploys the artificial landmarks equidistantly in time. The "random" method deploys the landmarks at random time steps. Whereas "always" deploys the landmarks at every time step until all landmarks have been deployed, the method "never" does not deploy any landmarks at all. In addition to that, we evaluated a heuristic in the sense of Kleiner et al. [88], which we will refer to as "density". This heuristic computes the obstacle density to the left and to the right of the robot by using kernel density estimation based on a simulated laser range scan. The heuristic furthermore uses a kernel density estimation in order to compute the density of the artificial landmarks that the robot has already deployed at the current location of the robot. The heuristic then uses the densities to decide whether to drop a landmark. We thereby manually tuned the parameters to optimize the performance of this heuristic.

Using each of the above-mentioned approaches to landmark deployment, the simulated robot performed 100 randomly sampled navigation tasks in a 6×6 Manhattan-like environment that comprises 96 indistinguishable environment features. See Figure 4.1 for a sample run. Figure 4.4 summarizes the results of the experiment. The results suggest that our approach significantly outperforms all the other methods in terms of the error E,

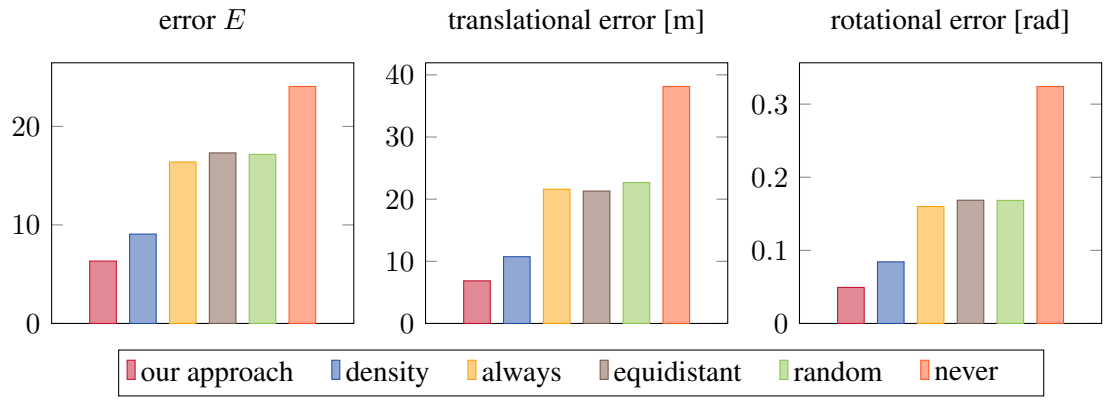


Figure 4.4: A simulated robot maps a Manhattan-like environment that comprises indistinguishable environment features. The plots depict the feature correspondence error E, the translational error in the robot's pose, and the rotational error in the robot's pose averaged over 100 sample runs for our method and for five baseline methods.

the translational error, and the rotational error according to two-sided t-tests on a 95% confidence level.

4.6 Related Work

Data association in simultaneous localization and mapping has been studied thoroughly in the past, yet it remains a bottleneck to the development of robust approaches to mobile robot mapping. A widely-used approach to data association is the gated nearest neighbor algorithm [14]. This method relies on the normalized squared innovation test to decide whether a measurement is considered a candidate and then chooses the candidate with the smallest Mahalanobis distance if there is a candidate. The gated nearest neighbor algorithm, however, may perform suboptimally in challenging environments since it is prone to accepting wrong data associations as long as they are mutually consistent. The joint compatibility branch and bound method proposed by Neira and Tardós [134] therefore explicitly considers the correlations between the innovations to determine the compatibility of a set of associations. Olson [138] applies single cluster graph partitioning to the data association hypotheses in order to construct a pair-wise consistency graph. Olson then rejects those associations that are not consistent on a global scale.

Data association, however, is challenging and quickly becomes intractable in practice. The reason for this is that the number of associations that need to be considered grows exponentially with the number of feature observations. The complexity of the data association problem motivates the usage of artificial landmarks.

Most approaches to SLAM that rely on deployable landmarks consider deterministically observable landmarks in graph-like worlds. See Dudek et al. [43] and Bender et al. [17] for examples. Wang et al. [170] prove that the simultaneous localization and mapping in an undirected graph can be solved in a deterministic way if the robot can drop a deterministically observable directional marker. In contrast to these methods, our method

uses a probabilistic model to deal with noisy motion and measurements.

Batalin and Sukhatme [15] present a coverage strategy for a mobile robot that can deploy active markers and use them later to move in the direction suggested by them. Kleiner et al. [88] propose a manually designed heuristic based on the obstacle density and the marker density in order to deploy RFID markers during SLAM. In contrast to these approaches, our method uses Monte Carlo reinforcement learning to infer a landmark deployment policy based on simulated runs.

Strasdat et al. [158] use reinforcement learning to find a policy for feature selection that minimizes the distance between the final position of the robot and its goal position. Since they consider obstacle-free worlds, they do not need to incorporate information about the spatial structure of the environment into the learning method. Similarly, in the context of mobile robot localization, Thrun [162] decides which features to consider based on the average posterior localization error.

4.7 Conclusion

Data association is a bottleneck to robust simultaneous localization and mapping with mobile robots, especially in ambiguous environments. A robot that can deploy uniquely identifiable artificial landmarks in the environment can resolve ambiguities in the data associations by recognizing the artificial landmarks later on. In this chapter, we presented a novel method for learning a landmark deployment policy that maximizes data association performance in simultaneous localization and mapping. Our actor-critic Monte Carlo reinforcement learning approach relies on features that generalize to new environments. Our method does not interfere with the navigation task of the robot, for example by requiring the robot to make detours to place landmarks. As a result, our technique can be seamlessly integrated into most robotic systems. We implemented our method and tested it both in simulation and on a real mobile robot. Our experimental evaluation demonstrates that the policies learned by our approach significantly outperform four baseline policies and one state-of-the-art technique, leading to consistent and significantly more accurate maps. Such high quality maps are needed for mobile robot navigation and for other map-based services. In the next chapter, the robot will use an annotated map of the environment to give directions to goal locations in a natural and intuitive way. In the remaining chapters, the robot will rely on a map for autonomous navigation.

Chapter 5

Learning to Give Natural and Intuitive Route Directions

The previous two chapters addressed the problem of learning a map of a previously unknown environment with a mobile robot. In this chapter, we present a novel approach that allows a robot to use an annotated map of the environment to learn how to give natural route directions to a goal location based on a corpus of directions given by humans. We assume that humans attempt to describe routes in a way that optimizes some reward function. We use inverse reinforcement learning to unveil a reward function that explains the given route descriptions. This allows us to compute a model of the style of the descriptions at hand. More specifically, we formulate the process of describing a route as a Markov decision process, where the actions correspond to instructions that guide the user along the route towards the destination. Since the reward function depends on features of the instructions, the resulting models generalize to new environments and therefore allow a mobile robot to give natural and intuitive directions to people. We carried out user studies that suggest that the directions given by our approach are perceived as significantly more human compared to the directions given by a state-of-the-art method.

The previous two chapters addressed the problem of learning a map of the environment for autonomous mobile robot navigation. Mobile robots that freely move in environments populated by humans should be able to communicate with the humans in a natural and intuitive way. A common conversation that pedestrians face as they travel in public spaces is to give directions to people that are unfamiliar with the environment. To engage in such conversations, mobile robots need to understand how humans describe routes to each other in these situations. A mobile robot that is able to mimic the way people describe routes is not only able to help people that are foreign to a city. The robot is also able to inform people about its destination in a natural, intuitive, and effective manner.

Existing commericial services, such as Google Maps [59] and Bing Maps [130], typically provide turn-by-turn instructions according to a manually programmed set of

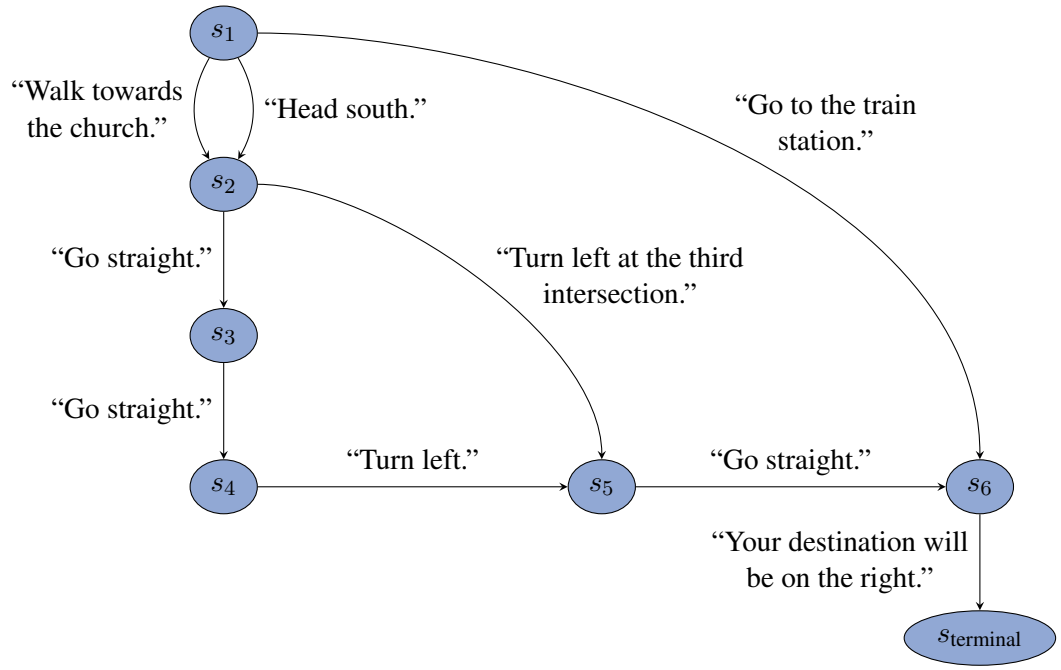


Figure 5.1: Example Markov decision process that captures valid descriptions of a given route.

rules. Although the resulting instructions are precise, they are not necessarily natural and intuitive. For instance, the instructions tend to heavily rely on precise metric information, as in "turn left after 260 meters." In contrast to that, humans rather refer to salient landmarks that are on the way, as in "turn left after the bridge." In addition to that, the instructions generated by these state-of-the-art techniques differ from the directions given by humans in many much more subtle ways, and mostly it is easy to tell that they are produced by a machine. Imagine the awkward situation that would arise when a person on the street gave directions to a foreigner in a way similar to these services.

The contribution of this chapter is a novel approach that allows a robot to learn how to give natural and intuitive route directions. We formulate the problem of describing a route as a Markov decision process and assume that humans attempt to optimize some reward function when giving directions. Inverse reinforcement learning then allows us to capture the style of a set of directions in terms of a reward function. The reward function depends on features that capture relevant properties of the route descriptions, such as cultural preferences. See Figure 5.1 for an example Markov decision process. Our experimental evaluation suggests that the directions generated by our approach are perceived as natural, intuitive and significantly more human-like than directions provided by state-of-the-art methods.

5.1 Giving Directions as a Reinforcement Learning Problem

We model the problem of giving route directions as a reinforcement learning problem. In reinforcement learning [160], an agent interacts with an environment that gives rise to rewards that the agent tries to maximize by carefully choosing its actions. Similar to Section 4.4, we formulate this reinforcement learning environment as a Markov decision process $\mathcal{M}_{\text{MDP}}$ given by a discrete set $\mathcal{S}$ of states $s \in \mathcal{S}$, a discrete set $\mathcal{A}$ of actions $a \in \mathcal{A}$, a deterministic transition model $P(s' \mid s, a)$ of the agent's next state s' given that the agent carries out action a in its current state s, and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that determines the reward that is given to the agent. The agent behaves according to a stochastic policy π, which is given by a probability distribution $\pi(a \mid s)$ over actions a to take when in state s. A path ζ in the Markov decision process is then determined by the agent's initial state and the agent's action choices. In the context of giving directions, an action corresponds to an atomic instruction, such as "turn left", that guides the user along the given route towards the target location, whereas a path consists of a sequence of such instructions. Consequently, the Markov decision process encodes the space of directions that successfully guide the user all the way to the target location along a given route, and a policy corresponds to a specific style of giving directions.

We assume that humans attempt to optimize some reward function when giving directions. The goal of reinforcement learning is to compute a policy $\pi(a \mid s)$ that maximizes the expected cumulative reward given a specific reward function. In contrast to that, inverse reinforcement learning is the problem of recovering an unknown reward function given a set $\mathcal{D} = \{\zeta_k\}_{k=1}^n$ of demonstrations in the form of paths in the Markov decision process. In the following, we will use inverse reinforcement learning to recover the reward function that humans attempt to optimize in order to learn a model of a specific human style of giving directions.

5.2 Learning from Demonstrations

When casting the problem of giving directions as a reinforcement learning problem, learning a particular style of giving directions from human demonstrations translates to finding a reward function that explains the demonstrated human style. To compute the reward function based on demonstrated behavior, we use maximum entropy inverse reinforcement learning [179, 180].

We assume that the style is governed by a probability distribution over paths in the Markov decision process that depends on features $\mathbf{f}$ of the paths. The empirical feature values

$$\mathbf{f}_{\mathcal{D}} := \frac{1}{|\mathcal{D}|} \sum_{\zeta_k \in \mathcal{D}} \mathbf{f}(\zeta_k) \tag{5.1}$$

then capture relevant properties of the demonstrations $\mathcal{D} = \{\zeta_k\}_{k=1}^n$. Our goal is to find a distribution $P^\star$ that leads to the style of the demonstrations in terms of the features.

Hence, we require that the feature expectations of the model and the empirical feature values of the demonstrations match, giving

$$\mathbb{E}_{P^\star}[\mathbf{f}] = \sum_{\zeta} P^\star(\zeta)\mathbf{f}(\zeta) = \mathbf{f}_{\mathcal{D}}. \tag{5.2}$$

Matching the feature expectations, however, is ambiguous. Maximum entropy inverse reinforcement learning relies on the principle of maximum entropy [76] to resolve the ambiguity. The principle of maximum entropy suggests that we choose the distribution with the highest entropy because that distribution does not favor any particular style beyond the style suggested by the observed feature constraints. Accordingly, maximum entropy inverse reinforcement learning maximizes the Shannon entropy $H(P)$ of the distribution, leading to the optimization problem

$$P^\star = \operatorname*{argmax}_P H(P) \tag{5.3}$$

subject to the constraints

$$\mathbb{E}_P[\mathbf{f}] = \mathbf{f}_{\mathcal{D}}. \tag{5.4}$$

According to Ziebart et al. [180], the resulting distribution $P^\star$ is then given by

$$P_{\boldsymbol{\theta}}(\zeta) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(\zeta)\right) \tag{5.5}$$

with

$$Z(\boldsymbol{\theta}) = \sum_{\zeta} \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(\zeta)\right), \tag{5.6}$$

where $Z(\boldsymbol{\theta})$ is a normalization constant.

The resulting optimization problem is equivalent to maximizing the natural logarithm of the likelihood $\mathcal{L}$ of the model parameters $\boldsymbol{\theta}$ under the maximum entropy distribution given the demonstrations $\mathcal{D}$. The optimization problem is convex for deterministic Markov decision processes, and the gradient with respect to the model parameters $\boldsymbol{\theta}$ is given by

$$\nabla_{\boldsymbol{\theta}} \ln \mathcal{L} = \mathbb{E}_{P_{\boldsymbol{\theta}}}[\mathbf{f}] - \mathbf{f}_{\mathcal{D}}. \tag{5.7}$$

Consequently, we can apply gradient-based optimization to iteratively refine the model parameters $\boldsymbol{\theta}$ to match the feature expectations. In Section 5.5, we discuss an algorithm to compute the feature expectations $\mathbb{E}_{P_{\boldsymbol{\theta}}}[\mathbf{f}]$ induced by the current model parameters $\boldsymbol{\theta}$.

5.3 Features

The goal of this work is to learn a model of a specific style of giving directions from demonstrations. To capture the style of the directions, the model relies on features

$$f : \mathcal{A} \to \mathbb{R} \tag{5.8}$$

of instructions that map actions $a \in \mathcal{A}$ in the Markov decision process to real numbers $f(a)$. The feature value of a path ζ is then given by the sum of the feature values of the actions along the path, giving

$$f(\zeta) = \sum_{a_k \in \zeta} f(a_k). \tag{5.9}$$

In this section, we propose a set of such features of instructions grouped into three categories. For specific applications, other features are possible.

5.3.1 Detail

The information that the directions convey typically depends on the situation. For instance, printed directions that the user can take along tend to be more detailed than verbal directions given to someone on the street that need to be memorized. We propose two features to capture the detail of the directions. The first feature computes the total number of instructions given to the user. The second feature counts the units of information given to the user, which is similar to the concept of slots presented by Mark [126]. For instance, according to our definition, the instruction "turn right at the second intersection" conveys two units of information.

5.3.2 Saliency

Directions given by humans typically rely on landmarks, especially when referring to decision points. The question arises as to how humans choose the landmarks that they refer to when giving directions. We assume that humans tend to point out landmarks that are salient in the context of their environment. For instance, people seem to be much more likely to give directions such as "go down the road and turn left at the tree" when talking about a road with a single tree that indicates where the recipient needs to make the left turn, as opposed to a boulevard with many trees. In addition to that, some landmarks, such as the Eiffel Tower, are well known and highly distinctive.

Our approach uses features that capture the saliency of landmarks, the distances between landmarks and decision points, and the directions of landmarks relative to the user. To compute the saliency, our method groups the landmarks into categories and counts the references to these landmarks in the directions given by the participants, which constitute the training data. The resulting frequencies yield a first saliency score for each landmark category.

In addition to that, our approach assumes that landmarks that are often mentioned on the world wide web tend to be better known and more salient. Therefore, our method queries a web search engine to estimate the saliency of landmarks with proper names, such as "Eiffel Tower", based on the number of hits.

5.3.3 Abstraction

Turn-by-turn instructions typically refer to one decision point along the route at a time, such as "turn right" and "go straight". Chunked instructions guide the user across several decision points, such as "go straight until you get to the church" and "turn right at the next traffic light". Destination descriptions send the user to an intermediate target location without providing specific directions how to get there, such as "go to the train station". We propose features that compute the frequencies of the instructions of these categories. In addition to that, we propose to count the decision points that are mentioned in the instructions. Furthermore, we propose to consider the lengths of the route segments covered by the instructions.

5.3.4 Reference Frame

Directions can refer to landmarks, as in "go to the bridge", street names, as in "turn into Broadway", or they can rely on distances, as in "turn left after 100 meters". More specifically, cardinal directions send the user in the directions of north, east, south, or west. Allocentric directions rely on a global map and encode information about the location of a landmark relative to other landmarks. In contrast to that, egocentric directions refer to landmarks using a personal reference frame, such as "left" or "right". We use features that capture the frequencies of all these types of directions.

5.4 Contexts

Directions typically depend on contextual factors, such as the complexity of the route that is to be described. For example, instructing the user to turn at a complex intersection tends to require more information than simply telling the user to go straight ahead. We therefore consider contexts that capture the complexity of intersections in terms of the number of streets that are connected by the intersection. In addition to that, we consider the direction in which the route continues at the intersections. Moreover, we consider the category of the street, such as footpath or surface street, the saliency of street names, and whether there are significant landmarks at the intersection, among other things.

Similar to Ziebart et al. [181], our method is aware of the context by considering duplicate features, where each of the duplicates is only active in a specific context, leading to

$$f^k(a) := \begin{cases} f(a) & \text{if } c(a) = k \\ 0 & \text{if } c(a) \neq k, \end{cases} \tag{5.10}$$

where $c(a)$ captures the context of action a. In contrast to features, contexts do not depend on the actions, i. e., the directions.

However, learning the weights of each pair of features and contexts might require a large amount of training data or might lead to overfitting. To cope with small training sets, we aim to assess how much the features depend on the contexts. To this end, we

consider the mutual information

$$I(F;C) = H(F) - H(F \mid C) \tag{5.11}$$

between the features and the contexts for each pair of features and contexts, where H is the entropy, and F and C refer to random variables that capture the frequencies of the feature values and the contexts in the training data, respectively. As discussed in Section 2.2, the mutual information between two random variables is a measure of their mutual dependence. More specifically, the mutual information between a context and a feature measures the average reduction in uncertainty about the feature value that results from knowing the context. For instance, knowing that we are describing a turn yields a high information gain about the value of the feature "street name reference". Following Quinlan [146], in order to build a compact model, we combine the features with the contexts that lead to the highest mutual information and, hence, the highest information gain.

5.5 Computing Feature Expectations

Learning the model parameters $\boldsymbol{\theta}$ from training data requires computing the feature expectations $\mathbb{E}_{P_{\boldsymbol{\theta}}}[\mathbf{f}]$. Similarly, in Chapter 7 of this thesis, we are concerned with learning a model of navigation behavior from demonstrations in order to predict the trajectories of interacting agents such as pedestrians. To approximate the feature expectations with respect to the resulting probability distributions over continuous composite trajectories, we rely on Markov chain Monte Carlo sampling techniques. In this chapter, however, we are dealing with Markov decision processes with discrete state spaces and discrete action spaces. As a consequence, we can apply efficient dynamic programming techniques to compute the feature expectations. To this end, Ziebart et al. [180] propose to express the feature expectations in terms of expected action frequencies D_a in the Markov decision process, giving

$$\mathbb{E}_{P_{\boldsymbol{\theta}}}[\mathbf{f}] = \sum_{\zeta} P(\zeta \mid \boldsymbol{\theta}) \mathbf{f}(\zeta) = \sum_{a \in \mathcal{A}} D_a \mathbf{f}(a). \tag{5.12}$$

However, computing these action frequencies D_a by enumerating all possible paths in the Markov decision process is not computationally tractable in general.

Ziebart et al. therefore propose an efficient method to approximate these action frequencies, which is given in Algorithm 4. The notation of Algorithm 4 makes use of the indicator function $\mathbf{1}$ of a subset A, which is a function $\mathbf{1}_A : A \to \{1, 0\}$ that is defined as

$$\mathbf{1}_A(x) := \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases} \tag{5.13}$$

Furthermore, the notation refers to actions that transition the agent from state s_i to state s_j as actions $a_{i,j}$. The algorithm approximates the action frequencies based on a fixed time horizon T. More specifically, the method first computes the partition function $Z(\boldsymbol{\theta})$ given

Algorithm 4 Compute expected action visitation frequencies.

Input: MDP $\mathcal{M}_{\text{MDP}}$, model parameters $\boldsymbol{\theta}$, initial state s_{initial}, terminal states $\mathcal{S}_{\text{terminal}}$
Output: expected action visitation frequencies $D_{a_{i,j}}$

// Backward pass
for all states $s_i \in \mathcal{M}_{\text{MDP}}$ **do**
  $Z_{s_i} \leftarrow \mathbf{1}_{\mathcal{S}_{\text{terminal}}}(s_i)$
end for
for t = 1 ... T **do**
  for all actions $a_{i,j} \in \mathcal{M}_{\text{MDP}}$ **do**
    $Z_{a_{i,j}} \leftarrow \sum_{s_k} P(s_k \mid s_i, a_{i,j}) \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(a_{i,j})\right) Z_{s_k}$
    $Z_{s_i} \leftarrow \sum_{a_{i,j}} Z_{a_{i,j}} + \mathbf{1}_{\mathcal{S}_{\text{terminal}}}(s_i)$
  end for
end for

// Forward pass
for all states $s_i \in \mathcal{M}_{\text{MDP}}$ **do**
  $Z'_{s_i} \leftarrow \mathbf{1}_{\{s_{\text{initial}}\}}(s_i)$
end for
for t = 1 ... T **do**
  for all actions $a_{i,j} \in \mathcal{M}_{\text{MDP}}$ **do**
    $Z'_{a_{i,j}} \leftarrow Z'_{s_i} \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(a_{i,j})\right)$
    $Z'_{s_i} \leftarrow \sum_{a_{j,i}} Z'_{a_{j,i}} + \mathbf{1}_{\{s_{\text{initial}}\}}(s_i)$
  end for
end for

// Summing frequencies
for all actions $a_{i,j} \in \mathcal{M}_{\text{MDP}}$ **do**
  $D_{a_{i,j}} \leftarrow \frac{Z'_{s_i} \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(a_{i,j})\right) Z_{s_j}}{Z_{s_{\text{initial}}}}$
end for

in Equation (5.6) at each action $a_{i,j}$ by propagating the probability mass backwards from the terminal states to the initial state. The algorithm then propagates the probability mass from the initial state forward to the terminal states. Finally, the algorithm combines these results to compute the expected action frequencies.

5.6 Experimental Evaluation

We conducted two user studies to evaluate our approach. In the first user study, we asked our participants to give directions for a set of example routes. We then used our approach to learn a model of the style of the recorded directions. In the second study, we presented directions written by humans and computer generated directions to other participants and asked them to rate how natural the directions appear to them.

5.6.1 Acquiring Training Data

In order to acquire training data to learn a model of giving directions, we asked 13 participants of a web-based user study to give directions for several routes in Freiburg. All the participants were familiar with Freiburg and fluent in English. We presented them an unmodified, interactive OpenStreetMap map and asked them to give directions in English for a predefined route, which was clearly highlighted in the map. In total, we collected a corpus of 28 directions for ten routes each of lengths between 0.6 km and 2.9 km in an urban environment. The wording of the instructions given to the participants was the following:

> "Imagine the following situation. After a successful meeting, your business partner from abroad decides to walk to his next meeting and asks you for directions. Your business partner is nonlocal. However, he had visited the city a few times before, so he has some rudimentary knowledge about the city. He is supposed to stick to the route shown in blue. He will walk to his next meeting. You will print out the text and give it to him. He will take along your text, but he will not have a map of the area. Write in English in simple but complete sentences."

5.6.2 Learning a Model from the Training Data

As natural language processing is beyond the scope of this work, we manually processed the corpus of written route directions that we collected from the participants of the user study. More specifically, we inspected the directions and extracted a set of instructions that seemed to be frequently used in route directions in general, such as "turn left". Our approach subsequently used these instructions to formulate a Markov decision process for each of the given routes based on an annotated map. The Markov decision processes that we used in our experiments were composed of between 20 states and 50 states and between 20 000 actions and 100 000 actions, depending on the complexity of the corresponding route.

We then manually matched each of the route descriptions written by the participants to the path in the corresponding Markov decision process that yields the most similar description. This step implicitly corrects obvious mistakes in the directions written by the participants because all the paths in the Markov decision process correspond to correct descriptions. For instance, we mapped the user directions

> "Walk through the Eisenbahnstraße, on your left side you must pass near the bank Volksbank, and after a while on your right side you will pass the post office and the Edeka shop. Walk straight until you reach the semaphore, a nice park should be on your right side. Then turn right to the Rotteckring street and walk straight until you reach the tram trails. Then turn left and walk along the tram trail, you must have pharmacy on your left side. Pass the post office that is also on your left side and then turn right. You will be in the small street that has a lot of bars. Go straight until you reach the street

> with the tram trails and turn right. Walk straight, after the coffee bar you will reach the crossing where is your goal."

to the path in the Markov decision process that corresponds to the directions

> "Walk straight ahead onto Eisenbahnstraße. Continuing straight ahead, you'll pass Hauptpost on your right. You should see Edeka on your right. Pass Colombipark and continue. Go ahead until you get to a crossing. Turn right onto Rotteckring. At the tram line you have to turn left. Go along the trail of the tram, then at the next intersection take a right. When you get to the tram rails, turn right. Go straight ahead until you reach your goal."

To give another example, we mapped the user directions

> "Cross the street here at the huge pedestrian crossing and go straight ahead through the Eisenbahnstrasse. After a while, you see a public park on the left hand side. Continue straight ahead until again you reach a pedestrian crossing. There, turn right and follow the street named Rotteckring until you reach a crossing where the tramway passes. Turn left and follow the tram line along the Bertoldstrasse. Continue along the Bertoldstrasse until you reach a big square where tramlines go into every direction. Here, turn to the right and follow the Kaiser-Joseph-Strasse, which is a big shopping street, until you reach a historic building with an arc through which the tramway passes. This building is called the Martinstor. It is your destination."

to the path in the Markov decision process that yields the directions

> "Start heading towards the huge crossing. Go straight ahead across a traffic signal onto Eisenbahnstraße. Pass by Colombipark. Go straight until you reach a traffic light and turn right onto Rotteckring. Go ahead until you get to the rail of the tramway. Turn left onto Bertoldstraße. After a while, you will encounter Café Schmidt. In front of Café Schmidt turn right onto Niemensstraße. Turn right at the tram line onto Kaiser-Joseph-Straße. You have reached your final destination when you get to Martinstor."

The resulting paths in the Markov decision processes allow our approach to compute the empirical feature values $\mathbf{f}_{\mathcal{D}}$ of the demonstrations and to learn a model of the style of the directions that were given by the participants. To tackle the optimization problem given in Equation (5.3), our implementation relies on the RPROP algorithm [151]. In order to avoid overfitting to the training data, we add a penalty of complexity to the objective function by means of L_2 regularization. The resulting models generalize to new routes in new environments. Sampling from the models yields new route descriptions of the same style.

5.6.3 Evaluating the Learned Model

We evaluated how natural the directions generated by our method sound to humans by conducting another web-based user study. Prior to the user study, we used our approach to learn a model of the style of giving directions from the corpus described in Section 5.6.1. We then presented an interactive OpenStreetMap map along with directions of a clearly highlighted route in that map to the participants of the study. The directions presented to the participants were either written by humans, which we gathered in the first study, or generated by a machine. The directions generated by a machine were either directions generated by our method or directions generated by one of three popular web services. Specifically, we relied on Google Maps [59], Bing Maps [130], and MapQuest [9]. We asked the participants to rate how natural the presented directions sound. The wording of the question presented to the participants was: "How natural does this description sound to you?" The participants rated the directions by adjusting a continuous slider, where the two extremes were labeled as "sounds like a computer" and "sounds natural (human-like)." Each of the 12 participants rated descriptions of twelve distinct routes in random order. The participants were allowed to revise their ratings until the end of the study. To give an intuition, for one of the routes, these directions were given by a human:

> "Start going through the Katharinenstraße. When you get to a huge intersection, turn left onto the big street. Go straight until you come to the trail of the tram. Turn right there. Walk along the trail of the tram, then at the Bertoldsbrunnen you have to turn left onto the Salzstraße. Walk up to the Augustinermuseum, where you have to turn right and enter the Augustinerplatz."

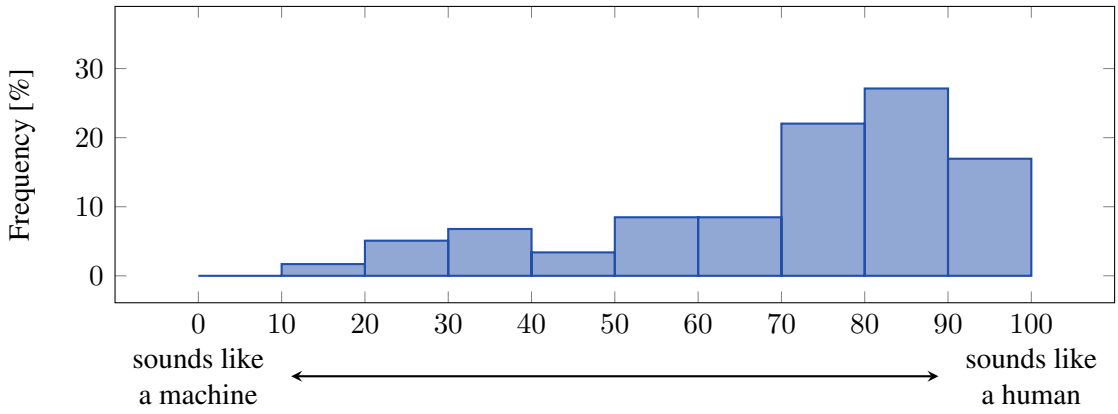
These directions were generated by our approach:

> "On Katharinenstraße start heading towards the Altstadt. Go along the street until you encounter a large intersection. Turn left onto the Friedrichstraße, and pass the Vapiano on your right. Go ahead until you come to the tram track. Turn right there onto the Kaiser-Joseph-Straße. Turn left after the Drogerie Müller. Turn right at the second possibility and enter the Augustinerplatz. You are at your target when you get there."

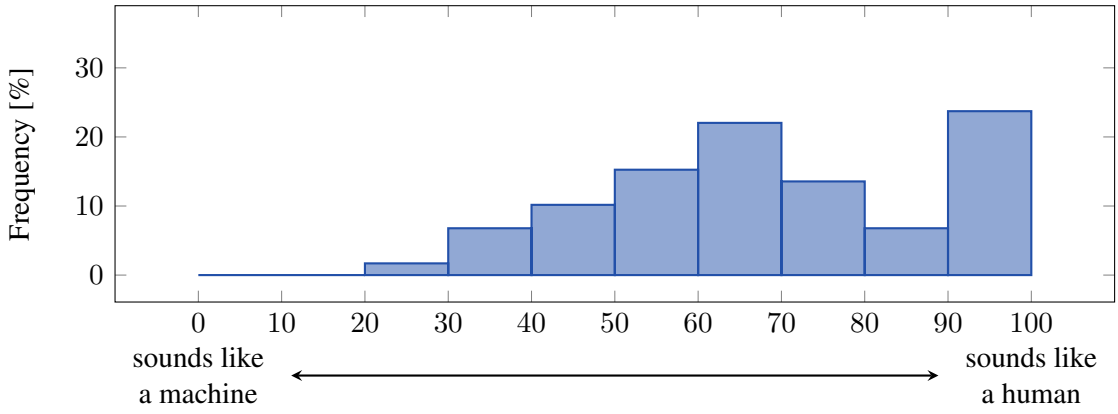
Finally, these directions were output by Google Maps [59]:

> "Head north on Katharinenstraße toward Rheinstraße. Turn right onto Rheinstraße. Turn right onto Merianstraße. Continue onto Rathausplatz. Continue onto Universitätsstraße. Turn left onto Bertoldstraße. Continue onto Salzstraße. Turn right onto Augustinerplatz."

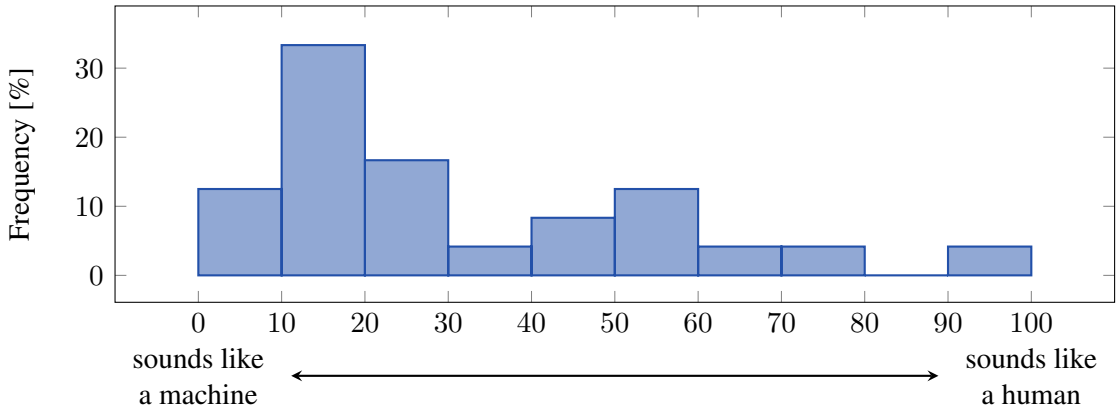
Figure 5.2 summarizes the ratings that the participants of the second user study assigned to the directions that were presented to them. The participants rated the directions generated by our method as more human compared to the directions generated by the other methods. The difference is significant according to a t-test at a 99 % confidence



(a) Directions given by humans.



(b) Directions given by our approach.



(c) Directions given by state-of-the-art routing web services.

Figure 5.2: Results of a user study that we conducted to evaluate whether the directions given by our approach qualify as human. The plots show histograms of the user ratings, which range from 0, which translates to "sounds like a machine", to 100, which translates to "sounds like a human". The results suggest that directions given by our approach are perceived as significantly more natural than the directions given by state-of-the-art routing web services [9, 59, 130].

level. Moreover, the participants rated the directions written by humans as more natural than the directions given by our approach. However, our experiments do not suggest that the difference is statistically significant.

5.7 Related Work

Researchers of various fields such as psychology, cognition science, geo sciences, linguistics, and computer science have investigated how humans give directions. For instance, Richter [149] provides a survey of findings on route directions generated by machines and directions given by humans and discusses the underlying cognitive principles. Allen [6] and Lovelace et al. [119] identify common characteristics of directions given by humans. Dale et al. [39] manually analyzed a corpus of route directions written by humans to derive a set of rules for generating natural and effective directions.

More specifically, several authors discuss the importance of references to landmarks in route directions. For instance, Lovelace et al. [119], Denis et al. [41], and Burnett [27] point out that directions that incorporate landmarks tend to be more effective. Waller and Lippa [169] discuss the importance of landmarks when learning routes. However, there does not seem to be a clear consensus on how humans choose landmarks to enrich route descriptions. Our approach learns a model from demonstrations that aims to imitate the way people choose the landmarks that they refer to when giving directions.

Various factors seem to affect the directions that humans give, such as the cultural background [74], the perspective of the recipient [74], gender [171], and, most important, the familiarity with the environment [119]. It is infeasible to model all these factors to generate appropriate directions for the application at hand. We therefore propose to learn a model of the directions from human demonstrations in order to imitate the style of a given group of people in a given setting.

Several researchers developed approaches to natural language processing to understand spatial language commands [91, 93, 94, 125], which can be seen as the inverse problem to the problem of generating directions. In contrast to these methods, automatic natural language processing to understand the directions written by humans in order to automatically formulate the Markov decision processes that model the space of route directions for given routes is, however, beyond the scope of our work.

Look [118] leverages findings from cognitive psychology about the limitations and organizational schemas of humans in order to structure route directions such that they are easier to understand. For instance, the resulting directions comprise overview descriptions of the neighborhoods the recipient will pass through on the way to the target location in addition to classical turn-by-turn instructions.

Some approaches to describing how to get to a given target location deliberately choose a route that is easy to describe even though it might not be the shortest route to the target location. For instance, Haque et al. [65] compute a route that minimizes the number of intersections that seem difficult to navigate. Richter and Duckham [150] compute a route that leads to the simplest directions. Goeddel and Olson [56] compute a route that maximizes the probability that the user actually manages to arrive at the target location.

They run a particle filter to simulate the user's ability to follow the directions. Cuayáhuitl et al. [36] propose a hierarchical reinforcement learning approach to choosing a route and then generating directions conditioned on that route. The algorithm minimizes the number of instructions and "user confusion" based on hand-crafted reward functions and transition dynamics.

However, research conducted by Hund et al. [73], for instance, suggests that there does not seem to be a clear consensus on what a good route description constitutes. They report on participants of a user study who navigated significantly faster in a model town when following route directions that other participants of that very study had rated to be the least effective directions. In contrast to Haque et al. [65], Richter and Duckham [150], and Goeddel and Olson [56], the directions generated by our approach are not necessarily optimal with respect to simplicity or reliability. Instead, our approach aims to imitate the style of directions written by humans, aiming to achieve natural, human-like route directions.

Many researchers have investigated ways to improve human-robot interaction. For instance, Kühnlenz et al. [109] aim at improving human-robot interaction by adapting the behavior of the robot to the mood of the user in terms of pleasure, arousal, and dominance. Gonsior et al. [58] combine human-like feedback, such as speech and facial expressions, with non-human-like feedback, such as a screen mounted on the robot that displays text. Gonsior et al. [57] explore dialog strategies in order to cope with erroneous speech recognition.

The robotics community has used inverse reinforcement learning [1] techniques to address a variety of imitation learning problems, such as autonomous helicopter aerobatics [2], learning pedestrian navigation behavior [86, 105, 106, 182], learning the preferences of cab drivers [181], and learning to play the game ball-in-a-cup [24]. For instance, Chapter 7 of this thesis presents an approach to learning a model of the navigation behavior of ineracting agents, such as pedestrians, that is inspired by inverse reinforcement learning techniques. In this chapter, we cast the problem of giving route directions as a reinforcement learning problem [160] and use maximum entropy inverse reinforcement learning [180] to learn how to give directions of routes from directions given by humans. As a result, our approach is able to imitate the style of the human directions in order to generate natural, human directions of routes.

5.8 Conclusion

In this chapter, we presented a novel approach to learning how to give natural route directions from a corpus of directions given by humans. We formulate the process of describing a route as a Markov decision process. The actions in this Markov decision process correspond to instructions that guide the user along the route towards the target location. Hence, the Markov decision process encodes the space of consistent directions that guide the user all the way to the target location. We assume that humans attempt to optimize some reward function when giving route directions. However, manually specifying this reward function to model a specific style of giving directions is cumbersome. We

therefore use inverse reinforcement learning to recover the reward function based on a corpus of directions written by humans. The reward function depends on features of the instructions. These features capture information conveyed by the instructions, such as descriptions of landmarks, street names, and the complexity of the resulting directions, among many other things. We carried out user studies that suggest that the directions given by our approach are perceived as significantly more human-like compared to the directions given by a state-of-the-art method. The resulting models generalize to new environments and therefore allow a robot to give natural and intuitive directions to goal destinations that are covered by its map of the environment. For instance, a mobile robot that travels in populated environments can use such a model to tell humans in its vicinity where it is currently heading to. In the remaining chapters, we will investigate how the robot can autonomously move to its target location in a socially compliant way that seamlessly blends in with the people nearby.

Chapter 6

Human Motion Capture with Mobile Robots

Having dealt with the problem of learning maps of the environment, we will now leverage the maps for autonomous mobile robot navigation. A robot that moves in an environment that is populated by humans should navigate in a socially compliant way. Our approach to socially compliant mobile robot navigation is to observe how people move in the environment of the robot and then learn a model of their behavior from these observations. In this chapter, we therefore present a novel method for human motion capture. Our technique combines locally accurate inertial motion capture with precise global alignment in the environment. We use a particle filter to track the pose of a target person. The motion model is based on inertial measurement units worn by the person. The measurement model is based on a laser range finder mounted on a mobile robot that uses a map of the environment for localization and to follow the person. To refine the trajectory estimates, our method solves a least squares problem based on the measurements of the inertial motion capture data and the laser measurements obtained by the mobile robot. The resulting estimates of the trajectories of pedestrians can be used to compute a model of their navigation behavior, a problem which we will address in the next chapter.

Unlike today's robots that mostly operate in factories, the next generation of robots is expected to be mobile and to move freely in open environments where people live and work. In the first chapters of this thesis, we addressed the problem of learning maps of the environment, thereby establishing a basis for autonomous robot navigation. To collaborate with people in a natural and intuitive way, however, the robots need to understand the behavior of the people they are surrounded by. In particular, socially compliant mobile robot navigation, which we will discuss in the next chapter, benefits from a model of the navigation behavior of the people in the environment. To learn such a behavior model, we need accurate training data of the movements of the people. The process of recording such training data is known as human motion capture.

On the one hand, there are optical motion capture systems that rely on a set of statically mounted cameras that track the positions of markers worn by the target persons. The most popular optical motion capture systems are provided by Motion Analysis [7] and Vicon [161]. These systems are highly accurate, but they are restricted to the area that can be captured by their cameras. Given the high price tags of these highly specialized cameras, optical motion capture systems are hardly ever suitable for motion capture in large areas.

On the other hand, there are inertial motion capture systems that attach a set of inertial measurement units to the target persons. For instance, the Xsens MVN [175] is an inertial motion capture suit that comprises 17 inertial measurement units, which enables full body human motion capture. Such systems lead to accurate estimates of the poses of the target persons on a local scale. However, the position estimates are subject to drift. The drift accumulates over time and therefore becomes more and more apparent as the person moves through the environment. As a result, the drift leads to inconsistencies in the position estimate of the person and the map of the environment. In the worst case, these inconsistencies prevent the method from properly capturing the person in the context of the environment and from properly capturing interactions between multiple persons. As a result, inertial motion capture tends to yield training data that is not suitable for socially compliant mobile robot navigation methods.

The contribution of this chapter is a novel approach that combines locally accurate inertial motion capture with precise global alignment in the environment. See Figure 6.1 for an illustration. Our method uses a particle filter to track the pose of a target person. The motion model of the particle filter is based on inertial measurement units worn by the person. The measurement model is based on a laser range finder mounted on a mobile robot. During navigation, the robot estimates its pose in a map of the environment, which has been built by means of simultaneous localization and mapping techniques, such as the method described in Chapter 3. To refine the trajectory estimates, our method solves a least squares problem based on the inertial motion capture data and the laser measurements obtained by the mobile robot. Our experimental evaluation based on an inertial motion capture suit suggests that our approach provides estimates of the person's full body posture that are, both, locally precise and globally aligned with the environment.

6.1 Tracking People with Particle Filters

We use recursive Bayesian state estimation [164] to track the poses of pedestrians with a mobile robot. More specifically, we use a particle filter to estimate the posterior probability distribution of a person's pose $\mathbf{x}_t$ in a planar environment with respect to a global reference frame at time t conditioned on the measurements $\mathbf{z}_{1:t}$ and the control commands $\mathbf{u}_{1:t}$ up to time t, which is given by

$$p(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \propto p(\mathbf{z}_t \mid \mathbf{x}_t) \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t)\, p(\mathbf{x}_{t-1} \mid \mathbf{z}_{1:t-1}, \mathbf{u}_{1:t-1})\, d\mathbf{x}_{t-1}. \quad (6.1)$$

Figure 6.1: We compensate for drift in locally accurate inertial motion capture by using a mobile robot equipped with a laser range finder.

The particle filter estimates this probability distribution using the motion model, which is given by the density $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t)$, and the measurement model, which is given by the density $p(\mathbf{z}_t \mid \mathbf{x}_t)$. The motion model is based on the odometry data $\mathbf{u}_t$ reported by the inertial measurement units worn by the person. The measurement model is based on laser measurements $\mathbf{z}_t$ of the person obtained by a localized mobile robot. In the following, we describe the motion model and the measurement model.

6.1.1 Motion Model

The motion model is given by the posterior probability distribution of the pose $\mathbf{x}_t$ at time t conditioned on the previous pose $\mathbf{x}_{t-1}$ and the control command $\mathbf{u}_t$, which is given by the density $p(\mathbf{x}_t \mid \mathbf{u}_t, \mathbf{x}_{t-1})$. Our current implementation relies on the motion capture suit Xsens MVN [175] that estimates the person's full body posture using a set of inertial measurement units. These inertial measurement units provide an estimate of the motion of the target person in the environment. We consider these measurements as control commands in the Bayes filter framework. The measurements of the inertial measurement units are subject to noise and drift. Consequently, the motion estimate is inaccurate on a global scale due to the accumulating pose error. However, the estimate is typically rather

precise on a local scale and therefore well-suited for the motion model. The pose estimate

$$\bar{\mathbf{x}}_t = (\bar{x}_t, \bar{y}_t, \bar{\theta}_t)^\top \tag{6.2}$$

provided by the inertial measurement units at time t is embedded in an internal reference frame whose relation to the map reference frame is unknown. Following Thrun et al. [164], the motion information $\mathbf{u}_t$ is then determined by the pose estimates $\bar{\mathbf{x}}_{t-1}$ and $\bar{\mathbf{x}}_t$ which are reported by the inertial measurement units at the two consecutive time steps $t-1$ and t. We conducted a statistical evaluation of the motion estimates provided by the inertial motion capture suit. As a result, we express the motion information $\mathbf{u}_t$ in terms of a translation δ_{trans} and a rotation δ_{rot}. Our probabilistic motion model assumes that the translation and the rotation are both subject to independent zero-mean Gaussian noise. The variances σ^2_{trans} and σ^2_{rot} of the normal distributions that model the noise depend on a weighted sum of the norm of the predicted translation and the absolute value of the rotation, yielding

$$\sigma^2_{\text{trans}} = \alpha_1 \|\delta_{\text{trans}}\|_2 + \alpha_2 |\delta_{\text{rot}}| \tag{6.3}$$

$$\sigma^2_{\text{rot}} = \alpha_3 \|\delta_{\text{trans}}\|_2 + \alpha_4 |\delta_{\text{rot}}|, \tag{6.4}$$

where the weights α_1, α_2, α_3, and α_4 are learned from manually labeled training data.

6.1.2 Measurement Model

The particle filter relies on the measurement model given by $p(\mathbf{z}_t \mid \mathbf{x}_t)$ to correct the prediction given by the motion model. Our measurement model compares the laser measurements that the mobile robot actually records with the expected measurements of the target person conditioned on the pose hypotheses. The measurement model can be tailored to the robot setup at hand depending on the laser range finder and the inertial measurement units. In our case, the laser range finder is mounted on the mobile robot such that the scanner observes the legs of the person. Our current implementation therefore specifically reasons about the appearance of the person's legs in the scans.

To this end, our measurement model constructs a three-dimensional skeleton model of the target person based on the pose hypothesis of the person in the planar environment and based on the current estimate of the person's full body posture, as provided by the inertial motion capture suit. The expected laser measurement is then given by the intersection of the plane of the two-dimensional laser measurement space and the skeleton model. Our method looks for potential legs in the current laser range scan to evaluate how well the expected measurement matches the actual measurement. We then consider the Euclidean distances between the expected locations of the legs and the locations of the detected legs. We furthermore assume that these distances are corrupted by zero-mean Gaussian noise.

Similar to Kluge et al. [89], our method detects legs in the laser range scans by grouping the laser endpoints into segments based on the Euclidean distances between endpoints of adjacent laser beams. Our method subsequently examines the widths and shapes of the resulting segments and considers those segments as leg candidates whose widths and shapes suggest that they might correspond to human legs.

Let $P(\text{left})$ and $P(\text{right})$ be the probabilities that the detector actually identifies the left and the right leg of the person, respectively, which we assume to be independent from the state $\mathbf{x}_t$ and the measurement $\mathbf{z}_t$. Since the data associations are unknown, we sum over all possible associations. We then assume

$$\begin{aligned} p(\mathbf{z}_t \mid \mathbf{x}_t) \propto\; & P(\neg\text{left})\, P(\neg\text{right}) \\ & + P(\text{left})\, P(\text{right}) \sum_{l_1 \neq l_2 \in L(\mathbf{z}_t)} \exp\left(-\lambda \left(\|\hat{l}_{\text{left}} - l_1\|_2 + \|\hat{l}_{\text{right}} - l_2\|_2\right)^2\right) \\ & + P(\text{left})\, P(\neg\text{right}) \sum_{l \in L(\mathbf{z}_t)} \exp\left(-\lambda \left(\|\hat{l}_{\text{left}} - l\|_2\right)^2\right) \\ & + P(\neg\text{left})\, P(\text{right}) \sum_{l \in L(\mathbf{z}_t)} \exp\left(-\lambda \left(\|\hat{l}_{\text{right}} - l\|_2\right)^2\right), \end{aligned} \tag{6.5}$$

where $\hat{l}_{\text{left}}$ and $\hat{l}_{\text{right}}$ refer to the positions of the left and the right leg of the skeleton, respectively, according to the current particle. Furthermore, $L(\mathbf{z}_t)$ refers to the set of legs detected in the current range scan, and λ is a scaling parameter that determines how peaked the distribution is. For each particle i, the filter then updates the weight w^i by

$$w_t^i \leftarrow \eta\, w_{t-1}^i\, p(\mathbf{z}_t \mid \mathbf{x}_t^i), \tag{6.6}$$

where η is a normalization constant that is computed by summing up the weights of all the particles. Our approach selectively carries out the particle filter resampling step by considering the effective number of particles, similar to Liu [117].

6.2 Robustly Tracking People in Populated Environments

Robustly tracking a target person in a populated environment is a challenging problem for various reasons. On the one hand, the tracker faces the risk of losing the target person when the mobile robot is unable to perceive the person using its on-board laser range finders. For instance, the robot might not be able to perceive the target person during occlusions due to walls or other people. On the other hand, the tracker faces the risk of mistaking the target person for a different person that is in the vicinity of the target person. A measurement that shows the target person as absent when it is present is called a false negative. A measurement that shows the target person at a location where it is absent is called a false positive. In general, it is difficult to design a measurement model based on two-dimensional laser range scans that avoids false negatives and false positives altogether.

In addition to that, our approach needs to perform global localization, which refers to the problem of localizing the target person without knowledge of the person's position. This problem arises during initialization when the relation between the reference frame internal to the inertial measurement units and the map reference frame is still unknown. Our approach can also recover from having lost the target person by means of global

localization. A common approach to global localization by means of particle filters is to inject particles into the filter that correspond to poses uniformly sampled from the entire state space, which, however, tends to be rather inefficient. An alternative method is to add samples to the particle filter such that the probability of adding a particular sample is proportional to its observation likelihood, which is similar to an algorithm proposed by Lenser and Veloso [114]. In our application, however, such a technique is likely to make the particle filter diverge, since the robot may observe other people while the person that is supposed to be tracked is outside the robot's field of view.

To achieve robust tracking in populated environments, our approach constantly evaluates candidate poses based on the measurements. In contrast to Lenser and Veloso [114], however, we require that a candidate be consistent with the measurements for some time before incorporating the hypothesis into the particle filter. To this end, we sample from the distribution of poses $\mathbf{x}_t$ conditioned on the last k motion estimates and the last k laser measurements, which is given by the density $p(\mathbf{x}_t \mid \mathbf{z}_{t-k+1:t}, \mathbf{u}_{t-k+1:t})$. By doing this, our algorithm seeks to prevent samples that correspond to a different person from being added to the particle filter. To efficiently sample from the distribution with density $p(\mathbf{x}_t \mid \mathbf{z}_{t-k+1:t}, \mathbf{u}_{t-k+1:t})$, we use a particle filter to track each of the candidates. We only incorporate samples into the main particle filter that correspond to those candidate poses whose movements are consistent with the measurements during the last k time steps.

6.3 Human Motion Capture in Large Areas

Our approach is able to track a target person in large areas over a long period of time. To this end, the mobile robot follows the person in order to obtain laser measurements, which allow the robot to correct the pose estimates obtained from inertial motion capture. To follow the target person, our current implementation uses a standard mobile robot path planning approach to guide the robot to the vicinity of the person. The robot performs Monte Carlo localization to estimate its pose in the map.

6.4 Optimizing Trajectories

The estimate of the target person's trajectory as computed by the particle filter is subject to discontinuities and not necessarily smooth. As a result, the trajectories do not accurately reflect the smooth motion of the person that the inertial measurement units originally captured. For instance, when rendering a virtual character based on the resulting trajectory estimates, such discontinuities translate to the character being teleported between nearby locations. Using these trajectory estimates as training data for approaches to socially compliant mobile robot navigation, such as the method that we present in Chapter 7 of this thesis, leads to suboptimal models. To bridge the gap between local accuracy and global alignment, we refine the trajectory estimate provided by the particle filter.

To this end, we are interested in the estimate $\mathbf{x}^{\star}_{1:T}$ of the target person's trajectory that is most likely given all the measurements $\mathbf{z}_{1:T}$ obtained by the robot at the poses $\mathbf{r}_{1:T}$ and the motion capture data $\mathbf{u}_{1:T}$, yielding

$$\mathbf{x}^{\star}_{1:T} = \operatorname*{argmax}_{\mathbf{x}_{1:T}} p(\mathbf{x}_{1:T}, \mathbf{r}_{1:T} \mid \mathbf{z}_{1:T}, \mathbf{u}_{1:T}). \tag{6.7}$$

We assume that the measurements $\mathbf{z}_{1:T}$, the motion capture data $\mathbf{u}_{1:T}$, and the robot poses that are estimated by means of Monte Carlo localization, are corrupted by independent Gaussian noise, which is a common assumption in the robotics community [164]. We can then compute the maximum likelihood trajectory estimate $\mathbf{x}^{\star}_{1:T}$ by means of least squares error minimization. More specifically, we are interested in finding the minimum of a function $F : \mathbb{R}^n \to \mathbb{R}$ of the form

$$F(\mathbf{x}_{1:T}, \mathbf{r}_{1:T}) = \sum_{t=1}^{T} e_{\mathbf{z}_t}^{\top} \Omega_{\mathbf{z}_t} e_{\mathbf{z}_t} + \sum_{t=2}^{T} e_{\mathbf{u}_t}^{\top} \Omega_{\mathbf{u}_t} e_{\mathbf{u}_t} + \sum_{t=1}^{T} e_{\mathbf{r}_t}^{\top} \Omega_{\mathbf{r}_t} e_{\mathbf{r}_t}. \tag{6.8}$$

The function F comprises error terms $e_{\mathbf{z}_t}$, $e_{\mathbf{u}_t}$, and $e_{\mathbf{r}_t}$ that correspond to the laser measurements $\mathbf{z}_{1:T}$, to the inertial motion capture data $\mathbf{u}_{1:T}$, and to the robot poses $\mathbf{r}_{1:T}$. These error terms are defined as

$$e_{\mathbf{z}_t} := (\mathbf{r}_t \oplus \mathbf{z}_t) \ominus \mathbf{x}_t \tag{6.9}$$

$$e_{\mathbf{u}_t} := (\mathbf{x}_{t-1} \oplus \mathbf{u}_t) \ominus \mathbf{x}_t \tag{6.10}$$

$$e_{\mathbf{r}_t} := \mathbf{r}_t \ominus \hat{\mathbf{r}}_t, \tag{6.11}$$

where $\hat{\mathbf{r}}_t$ refers to the robot pose as estimated by the Monte Carlo localization. The information matrices $\Omega_{\mathbf{z}_t}$, $\Omega_{\mathbf{u}_t}$, and $\Omega_{\mathbf{r}_t}$ model the uncertainty of the error functions corresponding to the error terms $e_{\mathbf{z}_t}$, $e_{\mathbf{u}_t}$, and $e_{\mathbf{r}_t}$, respectively. The resulting optimization problem is given by

$$\mathbf{x}^{\star}_{1:T} = \operatorname*{argmin}_{\mathbf{x}_{1:T}} F(\mathbf{x}_{1:T}, \mathbf{r}_{1:T}). \tag{6.12}$$

In general, solving Equation (6.12) is not feasible in closed form. However, we can compute the gradient of F with respect to $\mathbf{x}_{1:T}$ by means of a Taylor expansion and then apply gradient-based optimization techniques, such as the Gauss-Newton algorithm [145]. In practice, our implementation relies on g^{2}o, which is a general framework for such graph optimization proposed by Kümmerle et al. [110].

6.5 Experimental Evaluation

We carried out several experiments to evaluate the approach presented in this chapter. Our experiments rely on an inertial motion capture suit. However, depending on the application at hand, the inertial motion capture suit can be replaced with other devices that are equipped with inertial measurement units. For instance, exploiting data from sensors built into modern mobile phones is particularly interesting given the prevalent use of

(a) Inertial motion capture suit Xsens MVN. **(b)** Mobile robot ActivMedia Powerbot.

Figure 6.2: The robot setup of our experimental evaluation comprises an inertial motion capture suit and a mobile robot equipped with laser range finders. Left image courtesy of Xsens [175].

these devices nowadays. First, we present an experiment that analyzes the accuracy of the inertial motion capture suit. The estimates provided by the inertial motion capture suit are subject to drift. We evaluate how that drift affects the estimates over time. Subsequently, we evaluate the performance of our particle filter approach. We measure the position error at various checkpoints and compare the estimates of our approach to the raw estimates provided by the inertial measurement units. In addition to that, we evaluate the effects of the motion model on the tracker estimates and the effects of trajectory optimization.

6.5.1 Experimental Setup

Our experimental setup is based on the inertial motion capture suit Xsens MVN [175], which is depicted in Figure 6.2(a). The Xsens MVN suit consists of 17 inertial measurement units to estimate the full body posture of the person wearing the suit. The setup furthermore comprises the ActivMedia PowerBot mobile robot, which is depicted in Figure 6.2(b). The robot is equipped with a SICK LMS laser ranger finder for localization and perception.

6.5.2 Accuracy of the Inertial Motion Capture Suit

We conducted a first experiment to evaluate the accuracy of the raw position estimate provided by the inertial motion capture suit Xsens MVN. To this end, we had a human subject who was wearing the motion capture suit walk around our campus. The human

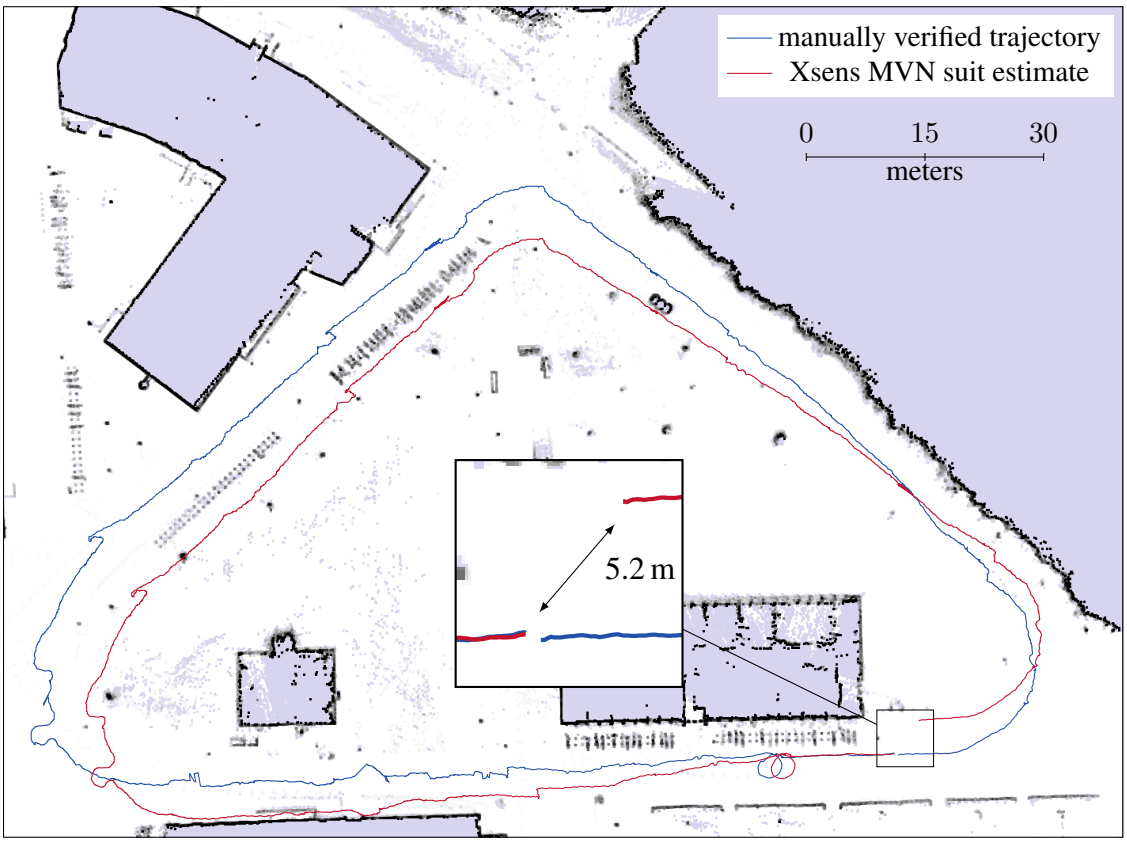


Figure 6.3: Accuracy of the raw position estimate provided by the inertial motion capture suit Xsens MVN during an experiment that we conducted on our campus. The red line visualizes the trajectory of a human subject as estimated by the motion capture suit. The blue line visualizes the true trajectory estimated by manually aligning the laser scans recorded by a mobile robot that was following the person. To assess the global drift of the position estimate reported by the motion capture suit, the start location and the end location were identical.

subject walked a loop in a way that the subject passed ten predefined checkpoints. We asked the human subject to visit each of the checkpoints, which allowed us to measure the position error of the motion capture suit. After having walked for around 45 minutes, the subject had completed ten such rounds and had traveled approximately 3000 m. Figure 6.3 depicts the trajectory recorded during one of the rounds, where the red solid line visualizes the raw estimate of the Xsens MVN suit, and the black dashed line visualizes a reference trajectory that was recorded by an accurately localized mobile robot. We obtained this reference trajectory by manually inspecting every single laser scan recorded by the mobile robot in order to provide the data associations between the scans, the human subject, and the environment. We measured an average position error of 5.2 m per round. Figure 6.4 depicts the position error at each of the ten checkpoints averaged over the ten rounds, and Figure 6.5 visualizes the evolution of the position error during the entire experiment.

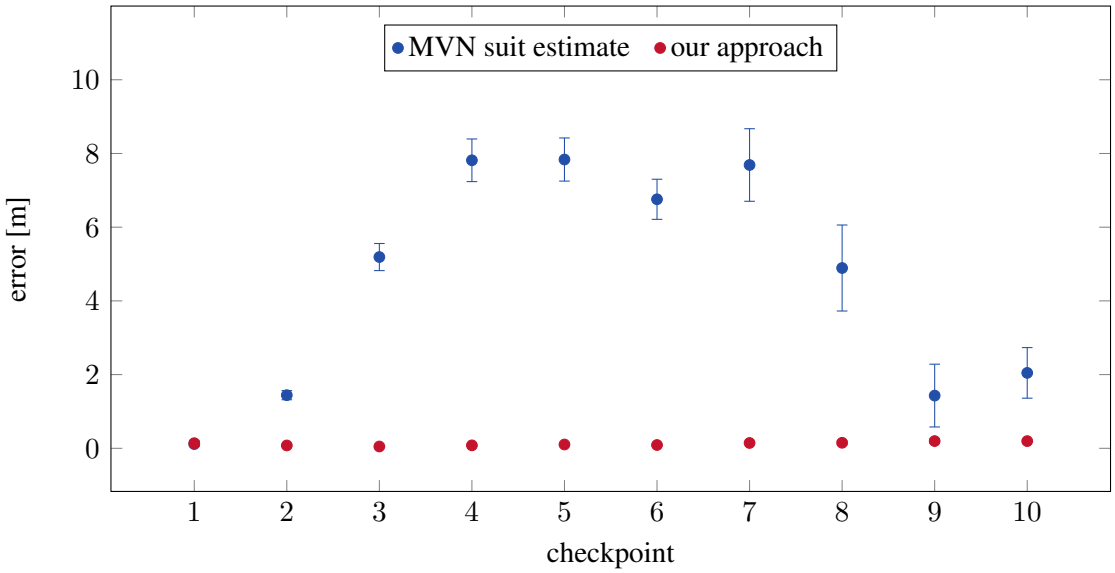


Figure 6.4: The error of the position of a human subject as estimated by the motion capture suit Xsens MVN, here depicted in blue, and as estimated by our approach, here depicted in red. The errors are given at predefined checkpoints along a 300 m loop averaged over ten rounds.

6.5.3 Large-Scale Human Motion Capture with a Mobile Robot

We evaluated the accuracy of the pose estimate computed by our method based on the setup described in the previous section. We evaluated the position error at the above-mentioned checkpoints and the evolution of the error during the experiment. Figure 6.4 depicts the position error at each of the ten checkpoints averaged over the ten rounds, where the error bars show the standard deviation of the error. In addition to that, Figure 6.5 assesses the evolution of the position error during the entire experiment. We carried out a paired sample t-test based on the position errors at all the checkpoints. The results indicate that the position error of our approach is significantly smaller at a 99 % confidence level.

We furthermore evaluated the ability of our approach to track a person wearing an inertial motion capture suit by having the mobile robot follow the person in a large area over a long period of time. We therefore used our approach to track a human subject that walked around our campus. To create a realistic setting, three other persons accompanied the target person who was wearing the inertial motion capture suit. To challenge our approach, we instructed these three persons to walk next to the target person and to frequently block the field of view of the mobile robot. In addition to that, several people who were not part of the experimental setup happened to show up and interacted with our human subjects in the course of the experiment. In total, the mobile robot depicted in Figure 6.2(b) autonomously followed the target person over a distance of about 1 km. Figure 6.6 shows the trajectory of the target person, where the red line refers to the estimate computed by our approach. The raw position estimate provided by the inertial motion capture suit is depicted in blue. Note that the raw trajectory estimate provided by the inertial motion capture suit is not compatible with the map despite an accurate

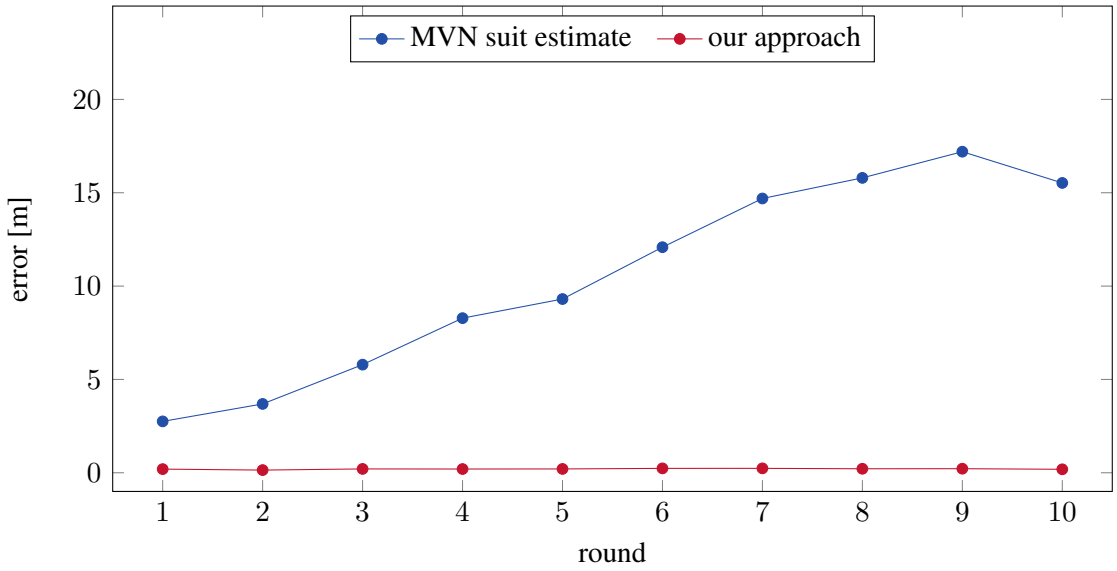


Figure 6.5: Evolution of the position error of the inertial motion capture suit Xsens MVN and our approach. In this experiment, a human subject walked a 300 m loop ten times. After each round, we assessed the error of the position estimate of the Xsens MVN and of our approach with respect to a known reference position marked on the ground. The results clearly suggest that the position estimate provided by the inertial motion capture suit drifts in the course of the experiment. In contrast to that, the position error of our tracking method is significantly smaller. More specifically, the position error was below 0.2 m at all times during the entire experiment.

pose initialization. As a result, the raw position estimate is not suitable for applications that require an accurate estimate of the position of pedestrians, such as the approach to socially compliant mobile robot navigation presented in Chapter 7 of this thesis. In contrast to that, our approach successfully tracked and followed the target person during the entire experiment, which provides evidence of the robustness of our approach.

6.5.4 Evaluating Motion Models

In this section, we compare the performance of the people tracker when using three different motion models. More specifically, we evaluate the performance based on two standard motion models, namely a constant velocity motion model and a Brownian noise motion model. In addition to that, we evaluate the performance based on the inertial motion capture suit motion model.

In this experiment, a mobile robot aims to track a target person walking in a group of three people. The results, which are depicted in Figure 6.7, suggest that the method fails to track the person when relying on the Brownian noise motion model or the constant velocity motion model, especially in the presence of other people who walk close to the target person or even occlude the target person. In contrast to that, when relying on the motion model that is based on the inertial motion capture suit, the method is able to correctly track the right person even in populated environments.

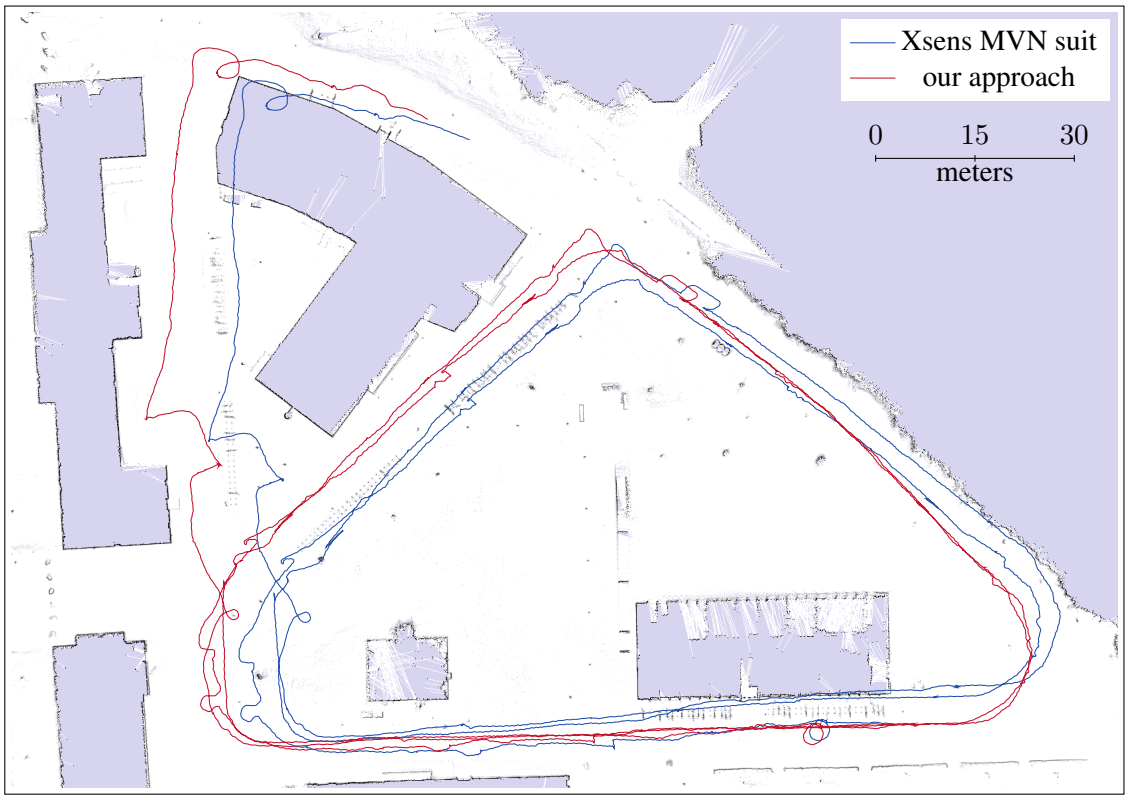


Figure 6.6: Trajectory of a target person as estimated by the inertial motion capture suit Xsens MVN and as estimated by our approach. In this experiment, a person walked around our campus, while a mobile robot tracks and follows the person. To create a realistic setting, several people accompanied the target person, frequently blocking the field of view of the mobile robot in the process. The trajectory estimate provided by the inertial motion capture suit is depicted in blue, whereas the trajectory estimate computed by our approach is shown in red. Note that the trajectory estimate provided by the inertial motion capture suit is not compatible with the map despite an accurate pose initialization.

6.5.5 Optimizing Trajectories

In this section, we evaluate the effects of the trajectory optimization technique presented in Section 6.4. We first analyze the local smoothness of the trajectories obtained by our approach. To this end, we chose two situations from the dataset depicted in Figure 6.6. The first situation, which is depicted in Figure 6.8, is an example of an erroneous leg detection that led to a local inconsistency in the estimate of the person's trajectory. Such an inconsistency can have negative effects on the application. For instance, when using the trajectory estimate to animate a movie character, the resulting non-continuous trajectory of the character is visually undesirable. When using the trajectory estimate to learn a navigation policy, as in Chapter 7 of this thesis, the excessive accelerations can lead to suboptimal navigation strategies. Our smoothing approach is able to compensate for that by exploiting the high local accuracy of the estimate of the inertial motion capture suit.

The second situation, which is depicted in Figure 6.9, illustrates the fact that the

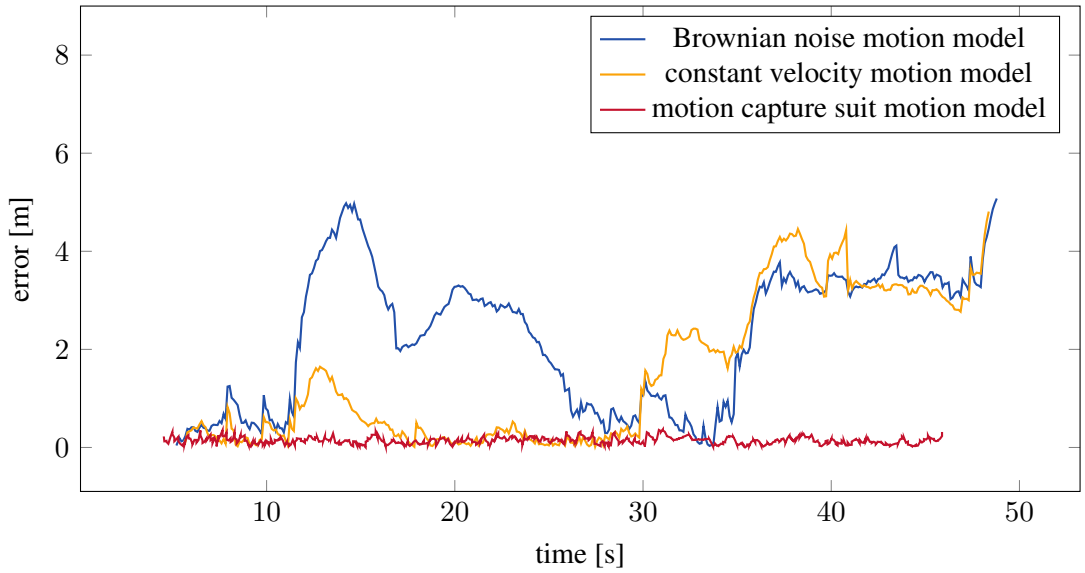


Figure 6.7: Performance of the tracker in the presence of a group of three people in the scene when relying on a Brownian noise motion model, a constant velocity motion model, and the inertial motion capture suit motion model. The plot suggests that the Brownian noise motion model and the constant velocity motion model both fail to accurately track the target person. In contrast to that, the position error of the tracker estimate when using the inertial motion capture suit motion model stays close to zero at all times.

trajectory optimization preserves local movements of the person. The estimate of the inertial motion capture suit, which is depicted in blue, and the estimate of our approach, which is depicted in red, locally exhibit the same behavior. This suggests that our approach appropriately preserves local movements by means of least squares optimization.

6.6 Related Work

Tracking people in the environment is a well studied problem in robotics and related disciplines. Robotics approaches to people tracking can be classified according to the sensor that is being used. Common methods rely on laser scanners [11, 50], monocular cameras [20], stereo cameras [46], inertial measurement units [137], or combinations of these sensors [28, 87, 90, 173, 185]. Key techniques found in most approaches to people tracking are Kalman filters [37, 50, 55], particle filters [132, 152, 173], and hidden Markov models [18]. To tackle the data association problem, most methods rely on nearest neighbor filters [132], the Hungarian method [20], multi-hypothesis data association filters [11], probabilistic data association filters [90], or the joint probabilistic data association filter [18, 153].

Montemerlo et al. [132] present an approach to simultaneously localizing a mobile robot and people in its vicinity. Similarly, our method also jointly estimates the trajectory

(a) Inertial motion capture suit Xsens MVN.

(b) Our particle filter.

(c) Trajectory smoothing based on the estimate of the particle filter.

Figure 6.8: Comparison of three estimates of the trajectory of a pedestrian roughly walking on a straight line. Notably, smoothing the trajectory estimate globally aligns the trajectory while preserving the local structure of the inertial motion capture suit estimate.

of a robot and a person. In contrast to Montemerlo et al. [132], however, our approach does not only track the person but also smoothes the trajectory estimate in order to obtain a more accurate maximum likelihood solution based on all measurements. Our method, however, does not take into account that the laser beams reflected by a person may negatively influence the pose estimate of the robot itself. Grzonka et al. [62] estimate the trajectory of a person wearing an inertial motion capture suit by incorporating information about key events, such as the person opening a door. They estimate the trajectory of the person as well as a topological-metric map.

The robotics community has proposed various motion models to improve people tracking. For instance, Luber et al. [121] incorporate a motion model that is based on the social forces concept [69] into a multi-hypothesis laser-based people tracker. Luber et al. [122] propose a place-dependent motion model that is based on spatial distributions of where people tend to walk. Bennewitz et al. [19] learn common motion patterns of pedestrians by means of expectation maximization clustering of demonstrated trajectories. Bruce and Gordon [25] learn to predict target locations of pedestrians by clustering demonstrations of trajectories. They subsequently use a path planner to predict the trajectories of the pedestrians to the target locations. To track pedestrians, they use a particle filter whose motion model relies on these predicted trajectories. In contrast to that, we use a motion model that is based on measurements obtained by inertial measurement units, such as the measurements reported by the inertial motion capture suit that we used in our experiments.

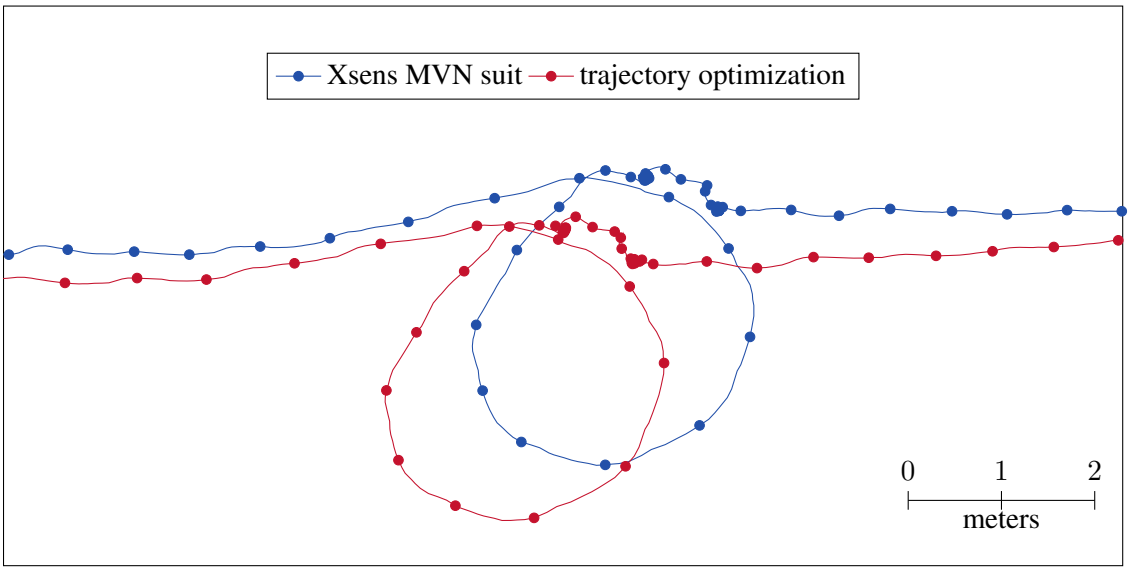


Figure 6.9: Close-up of two estimates of the trajectory of a pedestrian. The smoothed trajectory estimate, here depicted in red, preserves local movements, such as this small loop, in the raw trajectory estimate, here depicted in blue.

6.7 Conclusion

In this chapter, we presented a novel approach to human motion capture. Our method allows a mobile robot that is equipped with a laser range finder to spot and track pedestrians in large-scale environments. Our method uses a particle filter to track the poses of the pedestrians. In contrast to common approaches to people tracking whose motion models are based on Brownian motion or constant velocity assumptions, however, our approach relies on inertial measurement units that are worn by the target persons. These inertial measurement units are, for instance, worn by actors in the form of specialized motion capture suits for full-body human motion capture in the movie industry. Alternatively, exploiting data from inertial measurement units built into modern mobile phones is particularly interesting given the prevalent use of these devices nowadays. The measurement model of the particle filter is based on measurements of the pedestrians that a mobile robot obtains using its on-board laser range finders. These measurements allow our approach to compensate for drift in the motion predictions, which enables human motion capture applications in large areas over extended periods of time.

Our approach effectively performs global localization of the target persons by selectively injecting pose hypotheses into the particle filter that prove to be consistent with respect to motion predictions and measurement updates over some period of time. To obtain a smooth estimate of the trajectories of the persons, we formulate a joint least squares optimization problem that leads to a maximum likelihood estimate of the trajectory conditioned on all previous motion predictions and laser measurements. Our experimental evaluation suggests that our method yields smooth yet globally aligned estimates of the trajectories of pedestrians. Globally consistent pose estimates are particularly relevant to

applications where the target persons interact with each other or with the environment.

When using inertial motion capture suits, such as the Xsens MVN, which we used in our experiments, our method is suitable for large-scale motion capture of the full body posture of actors, for instance in the context of movie production. Moreover, our approach leads to estimates of the trajectories of pedestrians that can be used to train models of their navigation behavior, which we will discuss in the next chapter.

Chapter 7

Learning Navigation Behavior from Demonstrations

We leverage the techniques presented up to this point by developing a novel approach to learning a model of the cooperative navigation behavior of humans based on observations of their movements in the environment. We model the behavior in terms of a mixture distribution over composite trajectories that captures both the discrete navigation decisions as well as the natural variance of the continuous trajectories. Our approach learns the model parameters of this distribution that match, in expectation, the observed behavior in terms of user-defined features. To compute the feature expectations with respect to the continuous probability distributions over trajectories, we use highly efficient Hamiltonian Markov chain Monte Carlo sampling. Our method is able to imitate the behavior of pedestrians or, alternatively, to replicate a specific behavior that is taught by teleoperation in the target environment. We implemented our approach on two mobile robots and demonstrate that they are able to successfully navigate in an office environment in the presence of humans. An extensive set of experiments suggests that our method outperforms state-of-the-art approaches to modeling the behavior of pedestrians, which makes it also applicable to fields such as behavioral science or computer graphics.

Mobile robots are expected to increasingly populate our human environments in the near future. Applications range from domestic robots, flexible coworkers in factories, assistive robots in health care to autonomous cars. As a result, new challenges in the field of mobile robot navigation arise. The ability to put oneself in the position of others allows humans to navigate cooperatively, mutually evading each other in a safe and highly efficient way even in fast-paced, crowded environments. In order to seamlessly blend in with the humans, socially compliant mobile robots therefore need to be able to perceive the humans' intentions and predict their actions to engage in such cooperative collision avoidance just as well. To this end, the robots need accurate models of the navigation behavior of humans. The behavior of the humans in the environment of the robot, however,

specifically depends on the environment and the culture of the humans. For example, doctors in a hospital behave differently compared to pedestrians in a shopping mall. Mobile robots therefore need flexible means to infer a model of the human navigation behavior in any given environment.

Up to this point, we have developed robust probabilistic techniques that allow a mobile robot to learn a map of the environment, even during long-term operation. We have furthermore presented a method for accurately recording the movements of people. In this chapter, we finally leverage these results by using statistical inference to learn a model of the underlying human navigation behavior from such recordings. We specifically aim at modeling aforesaid cooperative navigation. To this end, our model jointly predicts the human actions in terms of a joint probability distribution over the space of trajectories that lead them from their current positions to their target positions, which we assume to be known. Although the space of trajectories is continuous, discrete navigation decisions, such as the question as to whether to pass an obstacle or another person on the left or on the right side, partition the space into homotopy classes. To explicitly account for these discrete navigation decisions, our model relies on a mixture distribution. This mixture distribution combines continuous probability distributions that each model the continuous navigation decisions for one of the homotopy classes. The distribution thereby implicitly reflects the natural variance in the trajectories, as observed in the observations. We propose a feature-based maximum entropy learning approach to fit the model to the observed behavior. More specifically, our goal is to match the feature expectations induced by our model to the empirical feature values of the demonstrations. To compute these feature expectations, we use Markov chain Monte Carlo sampling. Our spline-based representation of the trajectories allows us to use Hybrid Monte Carlo, which leads to highly efficient sampling. A mobile robot can use our approach to learn a model of the behavior of pedestrians in its environment. The robot can then use the learned model during navigation to predict the trajectories of nearby pedestrians. These predictions allow the robot to navigate in a safe, efficient, and socially compliant way.

The contribution of this chapter is a probabilistic framework to learn a model of the cooperative navigation behavior of physical agents, such as pedestrians, from demonstration. A key challenge of such a learning approach is the so-called forward problem, which involves computing the feature expectations with respect to a given distribution over the space of trajectories. Our representation of the probability distribution allows us to approximate these feature expectations using highly efficient Markov chain Monte Carlo sampling. Specifically, our continuous spline-based trajectory representation allows us to exploit the gradient of the probability density function to guide the sampling process towards regions of high probability by means of Hybrid Monte Carlo sampling. As a result, our approach is able to capture the stochasticity of the observed trajectories, which is in contrast to existing methods that learn deterministic models, which do not replicate well the stochastic behavior of natural agents. Furthermore, our technique reasons about trajectories in continuous spaces, which is in contrast to existing methods that perform trajectory inference based on a grid-based representation. An extensive set of experiments suggests that our model outperforms state-of-the-art methods to model pedestrian naviga-

tion behavior. The experiments include a Turing test in which the trajectories computed by our approach were perceived as more human than the trajectories computed by other methods. Moreover, the experiments demonstrate that our approach is suitable for mobile robot navigation in environments populated by pedestrians.

7.1 Modeling Predictive and Cooperative Navigation

The objective of this work is to learn a model that predicts the trajectories of physical agents such as pedestrians in a given situation. To account for nondeterministic behavior, we model the agents' navigation decision process as a probability distribution over their trajectories. Our probabilistic model reasons about the agents' trajectories from their current positions all the way to their target positions. We are particularly interested in modeling predictive and cooperative navigation. Predictive agents reason about the trajectories of other agents. In addition to that, cooperative agents give way to each other in order to maximize overall comfort. Moreover, when planning trajectories, cooperative agents assume that other agents just as well behave cooperatively and will also give way to them. See Figure 7.1 for an illustration. Specifically, the predictive, non-cooperative agent shown in red in Figure 7.1(c) is unable to plan a trajectory to pass the three agents shown in blue, since the agent does not account for the fact that the agents will step aside to let him pass. In contrast to that, the cooperative agent shown in red in Figure 7.1(d) assumes the other agents to behave cooperatively and is therefore able to plan a trajectory down the corridor. To account for such cooperation, we propose to jointly predict the trajectories of all the agents, as illustrated in Figure 7.1 for the agent shown in red. In summary, our model predicts the agents' behavior in a given situation in terms of a joint probability distribution over their trajectories to their target positions.

7.2 Learning from Demonstrations

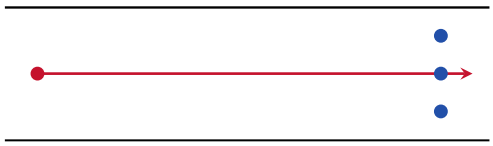
Learning from demonstrations refers to finding a model that explains observed demonstrations and generalizes to new situations. Similar to Section 5.2, when modeling the observed behavior in terms of a probability distribution over trajectories, learning can be cast as finding the distribution from which a set $\mathcal{D}$ of observed samples is drawn, where the distribution depends on features of these trajectories. Let $\mathbf{f}$ be a vector of features with

$$\mathbf{f} : \mathcal{T} \to \mathbb{R}^n \tag{7.1}$$

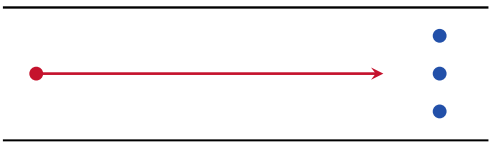
that maps trajectories $\boldsymbol{\tau} \in \mathcal{T}$ to a vector $\mathbf{f}(\boldsymbol{\tau})$ of real feature values. Similar to Equation (5.1), the empirical feature values

$$\mathbf{f}_{\mathcal{D}} := \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{\tau}_k \in \mathcal{D}} \mathbf{f}(\boldsymbol{\tau}_k) \tag{7.2}$$

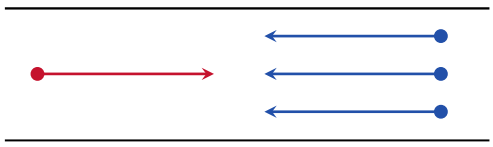
then capture relevant properties of the demonstrations $\mathcal{D}$.



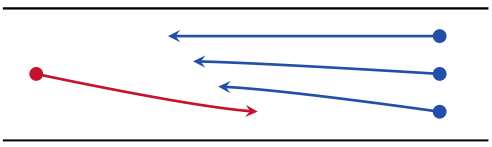
(a) No collision avoidance.



(b) Local collision avoidance.



(c) Predictive, non-cooperative navigation.



(d) Predictive, cooperative navigation.

Figure 7.1: Predictive and cooperative navigation strategies. The figures depict the trajectories of all the agents as predicted by the agent shown in red during an example encounter with the three agents shown in blue. (a): An agent without any means of collision avoidance plans a trajectory to its target location without taking into account other agents. This behavior might lead to collisions if the other agents do not evade early enough. (b): A non-predictive agent that relies on local collision avoidance treats other agents as static obstacles and is therefore unable to plan a trajectory to go past the other agents. (c): A predictive, non-cooperative agent predicts the trajectories of the other agents, for example by means of a constant velocity model. However, the agent does not assume the other agents to behave cooperatively. As a result, the agent is not able to plan a trajectory past the blue agents. (d): A predictive, cooperative agent predicts the behavior of other agents and assumes the other agents to also behave cooperatively. Hence, cooperative agents mutually give way to each other. As a result, the agent is able to plan a trajectory past the blue agents.

Similar to Section 5.2, we aim to find the distribution p that matches these empirical feature values in expectation, giving

$$\mathbb{E}_p[\mathbf{f}] = \int p(\boldsymbol{\tau})\mathbf{f}(\boldsymbol{\tau})\, d\boldsymbol{\tau} = \mathbf{f}_{\mathcal{D}}. \tag{7.3}$$

In general, however, there is not a unique distribution that satisfies the constraint expressed in Equation (7.3). The question arises as to which distribution to choose among all the distributions that satisfy the constraint. The principle of maximum entropy states that the distribution with the highest entropy represents the given information best, since it does not favor any particular outcome besides the observed constraints. In Chapter 5, we therefore considered maximum entropy distributions in the context of inverse reinforcement learning based on Markov decision processes with discrete state spaces and discrete action spaces. The principle of maximum entropy also applies to probability distributions over trajectories in continuous state spaces. Hence, in this chapter, we aim to find the distribution $p^\star$ that satisfies Equation (7.3) and maximizes the differential entropy

$$h(p) = -\int p(\boldsymbol{\tau}) \log p(\boldsymbol{\tau})\, d\boldsymbol{\tau}, \tag{7.4}$$

which we define in Section 2.2. This leads to

$$p^\star = \operatorname*{argmax}_p h(p) \tag{7.5}$$

subject to the constraints

$$\forall i \; f_{i\mathcal{D}} = \mathbb{E}_p[f_i] = \int p(\boldsymbol{\tau}) f_i(\boldsymbol{\tau}) \, d\boldsymbol{\tau} \tag{7.6}$$

with

$$\int p(\boldsymbol{\tau}) \, d\boldsymbol{\tau} = 1. \tag{7.7}$$

The constraints require that for all features f_i the expectation match the empirical value $f_{i\mathcal{D}}$. The constraints furthermore require that the probability density be correctly normalized.

We apply the method of Lagrange multipliers to solve this constraint optimization problem. Introducing Lagrangian multipliers α and θ_i for these constraints yields the maximization problem

$$p^\star, \boldsymbol{\theta}^\star, \alpha^\star = \operatorname*{argmax}_{p, \boldsymbol{\theta}, \alpha} \Lambda(p, \boldsymbol{\theta}, \alpha), \tag{7.8}$$

where

$$\begin{aligned} \Lambda(p, \boldsymbol{\theta}, \alpha) := &- \int p(\boldsymbol{\tau}) \log p(\boldsymbol{\tau}) \, d\boldsymbol{\tau} \\ &- \sum_i \left(\theta_i \left(\int p(\boldsymbol{\tau}) f_i(\boldsymbol{\tau}) \, d\boldsymbol{\tau} - f_{i\mathcal{D}} \right) \right) - \alpha \left(\int p(\boldsymbol{\tau}) \, d\boldsymbol{\tau} - 1 \right). \end{aligned} \tag{7.9}$$

Applying the Euler-Lagrange equation [52] to Equation (7.8) implies that the probability distribution $p^\star$ is given by

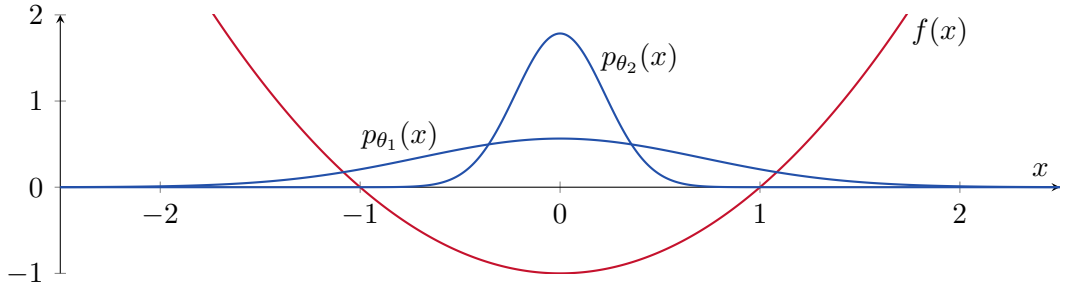
$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(\boldsymbol{\tau})\right) \tag{7.10}$$

with

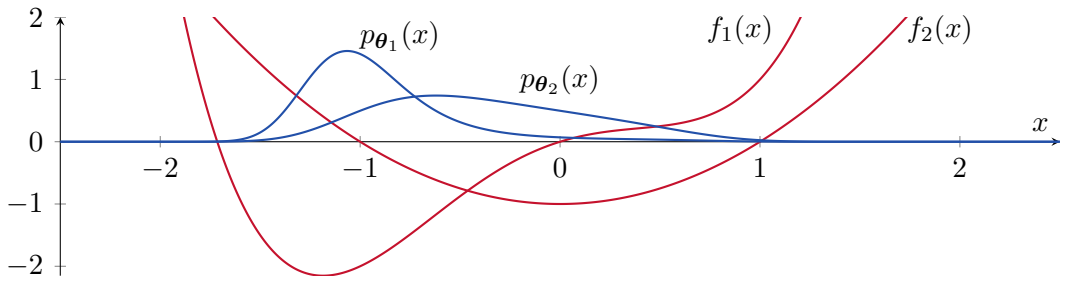
$$Z(\boldsymbol{\theta}) = \int \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}(\boldsymbol{\tau})\right) d\boldsymbol{\tau}, \tag{7.11}$$

where the partition function $Z(\boldsymbol{\theta})$ ensures that the probability density function $p_{\boldsymbol{\theta}}$ is correctly normalized. See Figure 7.2 for examples of maximum entropy distributions in one dimension. The figure illustrates how the parameter vector $\boldsymbol{\theta}$ affects the shape of the probability distribution. Specifically, the distributions become more peaked as the values of the parameters increase, which eventually allows our algorithm to replicate the demonstrated behavior.

The structure of the probability density function $p^\star$ that best explains the demonstrations is determined by the features $\mathbf{f}$. Their weights $\boldsymbol{\theta}^\star$, however, depend on the



(a) The maximum entropy distributions shown in blue are given by the probability densities $p_{\theta_i}(x) \propto \exp\left(-\theta_i f(x)\right)$ with $\theta_1 = 1$ and $\theta_2 = 10$, where the feature f depicted in red is given by $f(x) = x^2 - 1$.



(b) The maximum entropy distributions shown in blue are given by the probability densities $p_{\boldsymbol{\theta}_i}(x) \propto \exp\left(-\boldsymbol{\theta}_i^\top \mathbf{f}(x)\right)$ with $\boldsymbol{\theta}_1 = (2, 1)^\top$ and $\boldsymbol{\theta}_2 = (1, 2)^\top$, where the feature vector depicted in red is given by $\mathbf{f} = (f_1, f_2)^\top$ with $f_1(x) = x^4 - 0.5x^3 - 1.5x^2 + x$ and $f_2(x) = x^2 - 1$.

Figure 7.2: Example maximum entropy distributions in one dimension and the effect of the feature weights on the distributions. The maximum entropy distribution becomes more peaked as the feature weights increase.

demonstrations $\mathcal{D}$. It is not feasible to compute the feature weights $\boldsymbol{\theta}^\star$ analytically, yet we can apply gradient-based optimization techniques, where the gradient is given by

$$\nabla_{\boldsymbol{\theta}} \Lambda = \mathbb{E}_{p_{\boldsymbol{\theta}}}[\mathbf{f}] - \mathbf{f}_{\mathcal{D}}. \tag{7.12}$$

The problem of finding the maximum entropy distribution that matches the empirical feature values $\mathbf{f}_{\mathcal{D}}$ in expectation is equivalent to maximizing the log likelihood of the feature weights $\boldsymbol{\theta}$ given the training data $\mathcal{D}$ when assuming an exponential family distribution [76]. In this case, the natural logarithm of the likelihood of the feature weights $\boldsymbol{\theta}$ given the demonstrations $\mathcal{D}$ is given by

$$\ln \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \ln \left(\frac{1}{Z(\boldsymbol{\theta})} \exp\left(-\boldsymbol{\theta}^\top \mathbf{f}_{\mathcal{D}}\right) \right), \tag{7.13}$$

and its derivative with respect to $\boldsymbol{\theta}$ is given by

$$\nabla_{\boldsymbol{\theta}} \ln \mathcal{L} = \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) \mathbf{f}(\boldsymbol{\tau}) \, d\boldsymbol{\tau} - \mathbf{f}_{\mathcal{D}} = \mathbb{E}_{p_{\boldsymbol{\theta}}}[\mathbf{f}] - \mathbf{f}_{\mathcal{D}}.$$

In summary, we consider distributions whose expectations of the feature functions match the empirical values of the features given a set of demonstrations. We are interested in the distribution that maximizes entropy. This distribution is an exponential family distribution. Finding this distribution translates to computing the parameter vector $\boldsymbol{\theta}^\star$ that leads to feature matching or, alternatively, that maximizes the likelihood given the demonstrations. Even though computing $\boldsymbol{\theta}^\star$ analytically is not feasible, we can compute the gradient with respect to these parameters and apply gradient-based optimization.

7.3 Modeling Continuous Navigation Decisions

In this section, we learn a probabilistic model of the continuous navigation behavior of cooperatively navigating agents from a set $\mathcal{D}$ of demonstration trajectories. We are interested in the maximum entropy distribution over the trajectories with density $p_{\boldsymbol{\theta}_\mathbf{f}}$ that matches the demonstrations $\mathcal{D}$ in terms of a feature vector $\mathbf{f}$, leading to

$$\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}] = \mathbf{f}_\mathcal{D}. \tag{7.14}$$

As derived in Section 7.2, the distribution that maximizes the entropy subject to Equation (7.14) is then given by

$$p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}) = \frac{1}{Z(\boldsymbol{\theta}_\mathbf{f})} \exp\left(-{\boldsymbol{\theta}_\mathbf{f}}^\top \mathbf{f}(\boldsymbol{\tau})\right) \tag{7.15}$$

with

$$Z(\boldsymbol{\theta}_\mathbf{f}) = \int \exp\left(-{\boldsymbol{\theta}_\mathbf{f}}^\top \mathbf{f}(\boldsymbol{\tau})\right) d\boldsymbol{\tau}, \tag{7.16}$$

where $Z(\boldsymbol{\theta}_\mathbf{f})$ is a normalization factor, whose value we will not need to evaluate. The term

$${\boldsymbol{\theta}_\mathbf{f}}^\top \mathbf{f}(\boldsymbol{\tau}) = \sum_i \theta_{f_i} f_i(\boldsymbol{\tau}) \tag{7.17}$$

can be interpreted as a cost function that depends on a weighted sum of feature values. Hence, our model assumes that the agents are exponentially more likely to choose trajectories with lower cost.

Learning the navigation behavior in this context boils down to finding the feature weights $\boldsymbol{\theta}_\mathbf{f}$ that satisfy Equation (7.14). As described in the previous section, we can apply gradient-based optimization to compute these feature weights, where the gradient with respect to the parameter vector $\boldsymbol{\theta}_\mathbf{f}$ is given by Equation (7.12).

In the following, we present a continuous representation of the agents' trajectories. We furthermore introduce a set of features that capture physical properties of the trajectories such as velocities and accelerations. Last, we propose methods to approximate the feature expectations, which are necessary to compute the parameters of the model given training data.

7.3.1 Trajectory Representation

The representation of the agents' trajectories affects the inference process of the predictive model and the quality of the resulting predictions. We use the term trajectory to describe the position of a moving agent as a function of time. To obtain a smooth and differentiable representation of the trajectories, we use cubic splines to model the agents' planar movements in two-dimensional space. Specifically, each segment of these splines is given by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{a} + \mathbf{b}u + \mathbf{c}u^2 + \mathbf{d}u^3 \quad \text{with} \quad u \in [0, 1], \tag{7.18}$$

where the two-dimensional parameters $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{d}$ determine the shape of the segment. For each agent a_i, we specify the corresponding spline in Hermite form, i. e., in terms of a finite set of control points that determine the position and the velocity at the start points and the end points of the segments. Adjacent segments share control points in order to ensure continuity of the curve and its first derivative. We concatenate the segments and define an appropriate mapping from time t to the internal parameter u of the corresponding segment. The resulting spline-based representation allows us to efficiently compute the agent's position $\tau^{a_i}(t)$, the velocity $\dot{\tau}^{a_i}(t)$, and the acceleration $\ddot{\tau}^{a_i}(t)$ at every point t in time in closed form. The orientation of the agent is defined in terms of the first derivative of the spline.

We use the term composite trajectory to describe the positions of several agents as a function of time, capturing the agents' interactive navigation behavior in a given situation, i. e., how the agents evade each other. Specifically, a composite trajectory $\boldsymbol{\tau}$ that captures the positions of a set $\{a_i\}_{i=1}^n$ of agents at time t is given by

$$\boldsymbol{\tau}(t) = \left(\tau^{a_1}(t)^\top, \ldots, \tau^{a_n}(t)^\top\right)^\top, \tag{7.19}$$

Our probabilistic model predicts the trajectories of a set of agents by means of a probability distribution over composite trajectories.

7.3.2 Features

The goal of this work is to fit a navigation behavior model to observed trajectories. Specifically, to capture the behavior, the model relies on features

$$f : \mathcal{T} \to \mathbb{R} \tag{7.20}$$

that map composite trajectories $\boldsymbol{\tau} \in \mathcal{T}$ to real numbers $f(\boldsymbol{\tau})$. The features capture the behavior of groups of agents, which allows our model to learn a specific behavior for each of the groups.

In this section, we propose a set of such features, which are inspired by recent studies [71]. Among other things, these studies suggest that pedestrians consider features such as time of travel, velocities, accelerations, and clearances to other pedestrians. For specific applications, other features are possible.

7.3.2.1 Time

We propose to capture the time it takes the agents to reach their target positions by a feature

$$f^A_{\text{time}}(\boldsymbol{\tau}) := \sum_{a \in A} \int 1 \, dt, \tag{7.21}$$

where A refers to a group of agents. Similar to Mombaur et al. [131], this feature models the incentive to reach a certain target position as fast as possible.

7.3.2.2 Velocity

We propose to consider the velocity of the agents using a feature

$$f^A_{\text{velocity}}(\boldsymbol{\tau}) := \sum_{a \in A} \int \|\dot{\tau}^a(t)\|^2 \, dt, \tag{7.22}$$

which integrates the squared velocity over time for a group A of agents.

7.3.2.3 Acceleration

Pedestrians typically aim to walk efficiently, avoiding unnecessary accelerations [71, 131]. Integrating the squared acceleration over time yields the feature

$$f^A_{\text{acceleration}}(\boldsymbol{\tau}) := \sum_{a \in A} \int \|\ddot{\tau}^a(t)\|^2 \, dt, \tag{7.23}$$

where A refers to a group of agents.

7.3.2.4 Avoiding other agents

Pedestrians tend to evade other pedestrians. We assume that the evasive maneuvers depend on the distances between the agents. Hence, we use a feature

$$f^A_{\text{distance}}(\boldsymbol{\tau}) := \sum_{a \in A} \sum_{b \in A \setminus \{a\}} \int \frac{1}{\|\tau^a(t) - \tau^b(t)\|^2} \, dt, \tag{7.24}$$

where A refers to a group of agents.

7.3.2.5 Environment

We propose to capture the clearance to static obstacles, such as walls, as indicated by a map of the environment. We therefore use a feature

$$f^A_{\text{obstacle}}(\boldsymbol{\tau}) := \sum_{a \in A} \int \frac{1}{\|\tau^a(t) - o^a_{\text{closest}}(t)\|^2} \, dt, \tag{7.25}$$

where A refers to a group of agents, and o^a_{closest} is the position of the closest obstacle to agent a at time t.

7.3.3 Optimizing Trajectories

The composite trajectory $\boldsymbol{\tau}^\star$ that maximizes the probability density function $p_{\boldsymbol{\theta}_\mathbf{f}}$ is given by

$$\boldsymbol{\tau}^\star := \operatorname*{argmax}_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}). \tag{7.26}$$

The probability density function $p_{\boldsymbol{\theta}_\mathbf{f}}$ is differentiable with respect to the control points of the spline-based trajectories when using differentiable features, such as the features proposed in Section 7.3.2. In this case, we can efficiently compute the gradient of the density function $p_{\boldsymbol{\theta}_\mathbf{f}}$ by means of a series of analytical and numerical derivations. The experiments presented in Section 7.6.4 suggest that we can compute the trajectory $\boldsymbol{\tau}^\star$ by means of gradient-based optimization. Technically speaking, we use the RPROP algorithm [151] to optimize the trajectories with respect to the probability density function.

7.3.4 Computing Feature Expectations

As discussed in Section 7.2, learning the model parameters $\boldsymbol{\theta}_\mathbf{f}$ given training data requires computing the expectation of the features $\mathbf{f}$ with respect to the probability distribution $p_{\boldsymbol{\theta}_\mathbf{f}}$. Hence, we need to evaluate the feature expectations

$$\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}] = \int p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau})\mathbf{f}(\boldsymbol{\tau})\, d\boldsymbol{\tau}. \tag{7.27}$$

In general, however, computing these feature expectations with respect to distributions over continuous trajectories is not analytically tractable. Hence, we need to resort to some sort of approximation.

7.3.4.1 Dirac Approximation

As illustrated in Figure 7.3(a), a highly efficient way to approximate the feature expectations $\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}]$ is to approximate the distribution with density $p_{\boldsymbol{\theta}_\mathbf{f}}$ by a Dirac delta function at the mode

$$\boldsymbol{\tau}^\star = \operatorname*{argmax}_{\boldsymbol{\tau}} p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}) \tag{7.28}$$

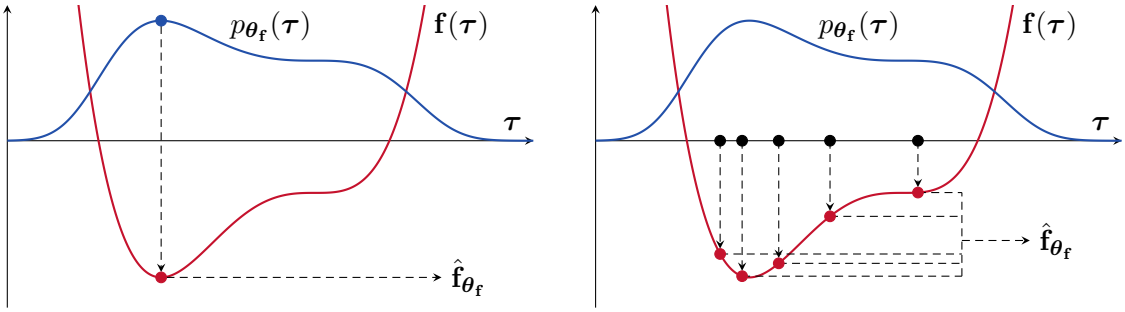
of the distribution $p_{\boldsymbol{\theta}_\mathbf{f}}$, where the Dirac delta function $\delta_{\boldsymbol{\tau}^\star}$ is given by

$$\delta_{\boldsymbol{\tau}^\star}(\boldsymbol{\tau}) := \begin{cases} +\infty & \text{if } \boldsymbol{\tau} = \boldsymbol{\tau}^\star \\ 0 & \text{if } \boldsymbol{\tau} \neq \boldsymbol{\tau}^\star \end{cases} \quad \text{with} \quad \int \delta_{\boldsymbol{\tau}^\star}(\boldsymbol{\tau})\, d\boldsymbol{\tau} = 1. \tag{7.29}$$

The approximation is then given by

$$\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}] \approx \hat{\mathbf{f}}_{\boldsymbol{\theta}_\mathbf{f}} := \int \delta_{\boldsymbol{\tau}^\star}(\boldsymbol{\tau})\mathbf{f}(\boldsymbol{\tau})\, d\boldsymbol{\tau} = \mathbf{f}(\boldsymbol{\tau}^\star), \tag{7.30}$$

which can be computed highly efficiently, as discussed in Section 7.3.3. However, this is a rather coarse approximation of the feature expectations. Specifically, agents that rely on



(a) Dirac approximation. **(b)** Sampling-based approximation.

Figure 7.3: Approximating the feature expectations $\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}]$. The figures illustrate the high-dimensional space of composite trajectories $\boldsymbol{\tau}$ and the space of feature values in one dimension, where the feature values of the composite trajectories are shown in red, and the density $p_{\boldsymbol{\theta}_\mathbf{f}}$ of the resulting maximum entropy distribution is depicted in blue.

this approximation reason about the distribution of trajectories solely in terms of the mode of the distribution and therefore neglect the variance in the trajectories in the process. As a consequence, models that are learned using this approximation do not capture imperfect human navigation behavior. Learning models by means of this approximation is known as inverse optimal control, originally posed by Kálmán [111].

7.3.4.2 Markov Chain Monte Carlo Sampling

A more principled way to approximate the feature expectations $\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}]$ that takes into account the distribution of trajectories is based on numerical sampling, even though it is computationally more involved than the Dirac approximation given in Equation (7.30). As illustrated in Figure 7.3(b), we can approximate the feature expectations based on a set $\{\boldsymbol{\tau}^{(k)}\}_{k=1}^{K}$ of sample composite trajectories drawn independently from the distribution with density $p_{\boldsymbol{\theta}_\mathbf{f}}$. The approximation is then given by

$$\mathbb{E}_{p_{\boldsymbol{\theta}_\mathbf{f}}}[\mathbf{f}] \approx \hat{\mathbf{f}}_{\boldsymbol{\theta}_\mathbf{f}} := \frac{1}{K} \sum_{k=1}^{K} \mathbf{f}\left(\boldsymbol{\tau}^{(k)}\right). \tag{7.31}$$

Directly sampling the composite trajectories from the distribution, however, is infeasible, yet we can obtain samples by means of Markov chain Monte Carlo (MCMC).

Markov chain Monte Carlo [8, 21] is a general framework to obtain samples from a high-dimensional target distribution by forming a Markov chain in the state space whose equilibrium equals the target distribution. A first-order Markov chain is a sequence of random variables $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(M)}$ such that the state $\mathbf{z}^{(m+1)}$ depends only on the state $\mathbf{z}^{(m)}$, which is formally given by

$$p(\mathbf{z}^{(m+1)} \mid \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}) = p(\mathbf{z}^{(m+1)} \mid \mathbf{z}^{(m)}). \tag{7.32}$$

The widely-used Metropolis-Hastings algorithm [21, 66] constructs the Markov chain by sampling a candidate $\mathbf{z}^\star$ for the next state from a proposal distribution with density $q(\mathbf{z} \mid \mathbf{z}^{(m)})$ that is conditioned on the current state $\mathbf{z}^{(m)}$. The algorithm then accepts the candidate state with probability $A(\mathbf{z}^\star, \mathbf{z}^{(m)})$, where

$$A(\mathbf{z}^\star, \mathbf{z}^{(m)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^\star) q(\mathbf{z}^{(m)} \mid \mathbf{z}^\star)}{\tilde{p}(\mathbf{z}^{(m)}) q(\mathbf{z}^\star \mid \mathbf{z}^{(m)})}\right) \tag{7.33}$$

with

$$p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}. \tag{7.34}$$

Hence, to evaluate the probability of acceptance given by Equation (7.33), we only need to be able to evaluate the density $p(\mathbf{z})$ for any given value of $\mathbf{z}$ up to some unknown normalization constant Z_p. If the candidate is accepted, then the algorithm sets $\mathbf{z}^{(m+1)}$ to $\mathbf{z}^\star$, otherwise the algorithm discards the candidate and sets $\mathbf{z}^{(m+1)}$ to $\mathbf{z}^{(m)}$. As long as the Markov chain is able to reach the entire state space according to the proposal distribution, the distribution of $\mathbf{z}^{(m)}$ converges to the target distribution with density $p(\mathbf{z})$ as m approaches infinity, irrespective of the choice of the initial state $\mathbf{z}^{(1)}$ of the Markov chain.

The proposal distribution and the resulting acceptance rate have a dramatic effect on the mixing time of the algorithm, e. g., the number of steps after which the distribution of the samples can be considered to be close to the target distribution. In general, it is difficult to design a proposal distribution that leads to satisfactory mixing. Furthermore, successive states in the Markov chain do not represent independent samples from the target distribution. They are rather highly correlated. We can obtain samples that are approximately independent by discarding all but every M^{th} sample if M is sufficiently large. For these reasons, efficient sampling from complex high-dimensional distributions is often not tractable in practice.

When sampling from continuous probability distributions, Gaussian distributions at the current state are often used as proposal distributions [21]. When sampling from the distribution over trajectories with density $p_{\boldsymbol{\theta}_\mathrm{f}}$, such a Gaussian proposal distribution, however, does not lead to satisfactory sampling. The distribution over trajectories tends to be highly peaked because of the physical laws that underly the trajectories. As a result, the induced Markov chain explores the space inefficiently owing to either a small step size or a high rejection rate. Moreover, the resulting random walk through the high-dimensional space of composite trajectories is prohibitory. The Hybrid Monte Carlo algorithm [21, 42] yet allows us to sample composite trajectories much more efficiently by avoiding random walk behavior. To this end, the algorithm exploits the gradient of the density of the target distribution to guide the sampling process towards regions of high probability. Owing to the spline-based representation of the trajectories described in Section 7.3.1, we can compute the gradient of the density efficiently when using the features presented in Section 7.3.2.

In essence, the Hybrid Monte Carlo algorithm samples from a target distribution with density $p(\mathbf{z})$ by simulating a fictitious, dynamical physical system that captures the

evolution of the state $\mathbf{z}$ under continuous time, where we denote time by t. The algorithm models the system according to Newton's second law of motion by using a momentum variable $\mathbf{r} = \{r_i\}$ that describes the rate of change of the position variable $\mathbf{z} = \{z_i\}$, where each component is given by

$$r_i = \frac{dz_i}{dt}. \tag{7.35}$$

The method casts the probability distribution with density $p(\mathbf{z})$ as

$$p(\mathbf{z}) = \frac{1}{Z_p} \exp\left(-E(\mathbf{z})\right), \tag{7.36}$$

where E can be regarded as the potential energy of the system when in state $\mathbf{z}$. The kinetic energy of the system, which depends on the current momentum, is given by

$$K(\mathbf{r}) = \frac{1}{2}\|\mathbf{r}\|^2 = \frac{1}{2}\sum_i r_i^2. \tag{7.37}$$

The total energy of the system is then given by

$$H(\mathbf{z}, \mathbf{r}) = E(\mathbf{z}) + K(\mathbf{r}), \tag{7.38}$$

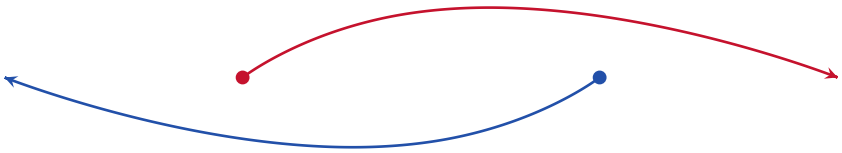
where H is called the Hamiltonian function. As the system evolves, its total energy is constant, which is why the density

$$p(\mathbf{z}, \mathbf{r}) = \frac{1}{Z_H} \exp\left(-H(\mathbf{z}, \mathbf{r})\right) \tag{7.39}$$

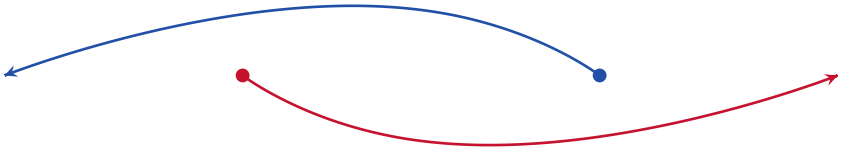
is also constant. Therefore, when sampling from the distribution with density $p(\mathbf{z}, \mathbf{r})$, integrating the Hamiltonian dynamics over time allows the algorithm to systematically make large steps with respect to $\mathbf{z}$ while avoiding both a low acceptance probability according to the Metropolis-Hastings criterion and random walk behavior. In order to enable the resulting Markov chain to explore the entire state space, the approach, however, needs to vary the value of the Hamiltonian function H while sampling. The Hybrid Monte Carlo algorithm therefore samples the value of $\mathbf{r}$ in each step from the conditional distribution $p(\mathbf{r} \mid \mathbf{z})$, which is a Gaussian. The method performs these stochastic updates of the momentum variable $\mathbf{r}$ and the Hamiltonian integration by means of the leapfrog algorithm [21, 42]. After performing a number of these "frog leaps", the method accepts the resulting candidate sample $(\mathbf{z}^\star, \mathbf{r}^\star)$ according to the Metropolis-Hastings criterion with probability $A(\mathbf{z}^\star, \mathbf{r}^\star, \mathbf{z}^{(m)}, \mathbf{r}^{(m)})$, where

$$A(\mathbf{z}^\star, \mathbf{r}^\star, \mathbf{z}^{(m)}, \mathbf{r}^{(m)}) = \min\Big(1, \exp\big(H(\mathbf{z}^{(m)}, \mathbf{r}^{(m)}) - H(\mathbf{z}^\star, \mathbf{r}^\star)\big)\Big), \tag{7.40}$$

to eliminate any bias introduced by numerical errors of the Hamiltonian integration.



(a) Two agents passing each other on the left side.



(b) Two agents passing each other on the right side.

Figure 7.4: Examples of two homotopically distinct composite trajectories that arise from the same initial situation comprising two agents.

7.4 Accounting for Discrete Navigation Decisions

In the previous section, we presented an approach to model continuous navigation behavior in terms of features, such as velocities, accelerations, and clearances when passing obstacles and other agents. In addition to that, however, the agents' decision process that ultimately leads to the observed trajectories in continuous spaces often explicitly comprises discrete decisions, such as the decision of passing another agent on the left or on the right side. These decisions partition the space of composite trajectories into homotopy classes. In this section, we extend the model to also account for these discrete navigation decisions. Specifically, we first review the concept of homotopy classes of the space of composite trajectories. We then cast the agents' discrete and continuous navigation decision process as a mixture distribution. We introduce a set of features to capture the discrete navigation decisions of the agents. Finally, we compute the expected feature values of the discrete probability distribution, which are needed for learning the model parameters given training data.

7.4.1 Homotopy Classes of Composite Trajectories

Figure 7.4 illustrates two composite trajectories of two agents each. Both composite trajectories lead the agents from the same positions to the same target positions. The composite trajectories depicted in (a) and (b), however, differ in the agents' decisions to bypass each other on different sides. Whereas the agents depicted in (a) pass each other on the left side, the agents depicted in (b) pass each other on the right side. Two composite trajectories are called homotopic if and only if they can be continuously transformed into each other without any collisions. Hence, the composite trajectories depicted in Figure 7.4 are non-homotopic. The agents' discrete decisions of bypassing each other on the left

side or on the right side partition the space $\mathcal{T}$ of composite trajectories into a set $\{\psi_i\}_{i=1}^{n}$ of homotopy classes, where all the composite trajectories of a given homotopy class are homotopic. For simplicity, we ignore composite trajectories that lead the agents around each other several times. We denote the space of homotopy classes by Ψ.

To ascertain whether two composite trajectories are homotopic, we compare their homology classes, which seems to be a suitable approximation in most practical situations. To describe the homology class of a composite trajectory $\boldsymbol{\tau} \in \mathcal{T}$, we capture the rotation of the angle $\alpha_a^b(t)$ between the vector $\tau^b(t) - \tau^a(t)$ and the vector $(1, 0)^\top$ over time for all agents a and b, where $\tau^a(t)$ and $\tau^b(t)$ are the positions of the agents a and b at time t, respectively. To do so, we integrate the time derivative of angle $\alpha_a^b(t)$ over time, which leads to

$$\kappa(a, b) := \int \dot{\alpha}_a^b(t)\, dt. \tag{7.41}$$

For example, since the two agents a and b depicted in Figure 7.4(a) pass each other on the left side, they yield

$$\kappa(a, b) > 0. \tag{7.42}$$

In contrast to that, since the two agents a and b depicted in Figure 7.4(b) pass each other on the right side, they yield

$$\kappa(a, b) < 0. \tag{7.43}$$

Most notably, however, this function is an invariant with respect to homology. This means that composite trajectories that belong to the same homology class leading the agents from the same start positions to the same target positions yield the same value, which allows us to compare the homology classes of composite trajectories. To efficiently compute $\kappa(a, b)$, we evaluate $\alpha_a^b(t)$ at discrete time steps t and sum up the angle differences. Since the step size does not affect the result as long as the steps still capture the direction of the angle, our implementation adapts the step size during the computation.

In Kuderer et al. [108], we extend this concept to homotopy classes with respect to static and dynamic obstacles. This work, however, is not part of this thesis.

7.4.2 Mixture Distribution

To explicitly account for the agents' discrete and continuous navigation decisions, we use a mixture distribution. See Figure 7.5 for an illustration. The density of the mixture distribution is given by a weighted sum of probability density functions that each model the agents' continuous navigation decisions for one of the homotopy classes of the space of composite trajectories. Specifically, the mixture density function $p_{\boldsymbol{\theta}_\mathbf{fg}}$ is given by

$$p_{\boldsymbol{\theta}_\mathbf{fg}}(\boldsymbol{\tau}) = \sum_{\psi \in \Psi} P_{\boldsymbol{\theta}_\mathbf{g}}(\psi) p_{\psi \boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}), \tag{7.44}$$

where Ψ refers to the space of homotopy classes. The probability distribution associated with a homotopy class $\psi \in \Psi$ is given by the density function

$$p_{\psi \boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}) = \begin{cases} p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}) & \text{if } \boldsymbol{\tau} \in \psi \\ 0 & \text{if } \boldsymbol{\tau} \notin \psi \end{cases} \tag{7.45}$$

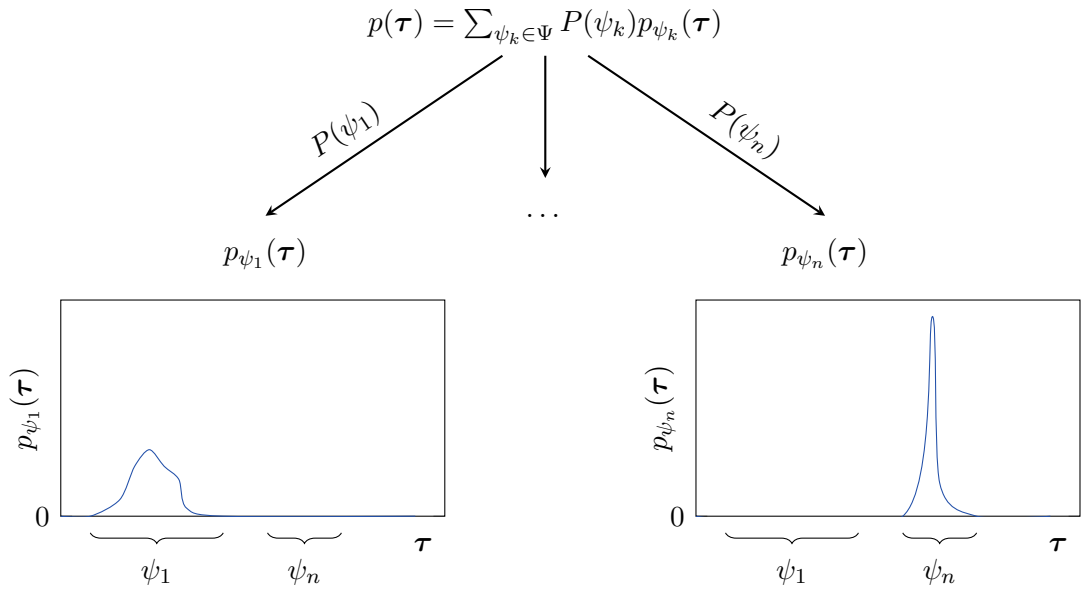


Figure 7.5: Mixture distribution to model the agents' discrete and continuous navigation behavior. The mixture density function $p(\boldsymbol{\tau})$ is given by a weighted sum of probability density functions $p_\psi(\boldsymbol{\tau})$ that capture the agents' continuous navigation decisions conditioned on the homotopy classes $\psi \in \Psi$. The weights assign a probability $P(\psi)$ to each homotopy class ψ of the space of composite trajectories. The figure illustrates the high-dimensional space $\mathcal{T}$ of composite trajectories in one dimension.

with

$$\int p_{\psi_{\boldsymbol{\theta}_\mathbf{f}}}(\boldsymbol{\tau})\, d\boldsymbol{\tau} = 1, \tag{7.46}$$

where the density $p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau})$ is given by Equation (7.15) in Section 7.3. To ascertain whether $\boldsymbol{\tau} \in \psi$, we compute $\kappa(a, b)$, as given in Equation (7.41), for all agents a and b.

When relying on such a mixture distribution, learning the agents' continuous navigation behavior requires computing the feature expectations with respect to probability distributions $p_{\psi_{\boldsymbol{\theta}_\mathbf{f}}}$ over composite trajectories of a given homotopy class ψ, as opposed to a probability distribution $p_{\boldsymbol{\theta}_\mathbf{f}}$ over the entire space $\mathcal{T}$ of composite trajectories. As a result, the mixture distribution enables us to efficiently approximate the feature expectations, since Markov chain Monte Carlo sampling is able to efficiently explore the space without the necessity of traversing regions of low probability that separate the homotopy classes, which would entail long mixing times.

Similarly to Section 7.2, to learn the discrete navigation behavior from demonstrations $\mathcal{D}$, our goal is to find the maximum entropy probability distribution $P_{\boldsymbol{\theta}_\mathbf{g}}(\psi)$ that induces the agents' discrete decisions in terms of features

$$g : \Psi \to \mathbb{R} \tag{7.47}$$

of the homotopy classes $\psi \in \Psi$. Hence, we want

$$\mathbb{E}_{P_{\boldsymbol{\theta}_\mathbf{g}}}[\mathbf{g}] = \mathbf{g}_\mathcal{D} = \sum_{\boldsymbol{\tau}_k \in \mathcal{D}} \frac{\mathbf{g}(\psi_{\boldsymbol{\tau}_k})}{|\mathcal{D}|}, \tag{7.48}$$

where $\mathbf{g}$ is a vector of such features. Furthermore, $\mathbf{g}_\mathcal{D}$ refers to the empirical values of the features given the training data $\mathcal{D}$, and $\psi_{\boldsymbol{\tau}_k}$ is the homotopy class of composite trajectory $\boldsymbol{\tau}_k$. The maximum entropy distribution that best explains the observed behavior in terms of the features without implying any further assumptions is then given by

$$P_{\boldsymbol{\theta}_\mathbf{g}}(\psi) = \frac{1}{Z(\boldsymbol{\theta}_\mathbf{g})} \exp\left(-{\boldsymbol{\theta}_\mathbf{g}}^\top \mathbf{g}(\psi)\right) \tag{7.49}$$

with

$$Z(\boldsymbol{\theta}_\mathbf{g}) = \sum_{\psi \in \Psi} \exp\left(-{\boldsymbol{\theta}_\mathbf{g}}^\top \mathbf{g}(\psi)\right). \tag{7.50}$$

The feature weights $\boldsymbol{\theta}_\mathbf{g}$ encode a model of the agents' discrete navigation behavior. We maximize the likelihood of the feature weights given the training data $\mathcal{D}$ by means of gradient-based optimization.

7.4.3 Features

We capture properties of the agents' navigation behavior in terms of homotopy classes by means of features. These features

$$g : \Psi \to \mathbb{R} \tag{7.51}$$

map homotopy classes $\psi \in \Psi$ to real values $g(\psi) \in \mathbb{R}$. In the following, we propose a set of features to capture some properties of natural decision making during navigation.

7.4.3.1 Passing Left vs. Passing Right

We propose to capture the decisions of avoiding other agents on the left or on the right side. We therefore use the feature

$$g_{\text{lr}}(\psi) := \sum_{a \in A} \sum_{b \in A \setminus \{a\}} \kappa(a, b), \tag{7.52}$$

where A is a group of agents, and $\kappa(a, b)$ is defined in Equation (7.41).

7.4.3.2 Group Behavior

In case we are able to recognize group memberships of agents, we propose to capture agents moving through groups of other agents. To this end, we propose to use a feature

$$g_{\text{group}}(\psi) := \sum_{a \in A} \left| \left\{ G \in \mathcal{G} \mid \exists b, c \in G \setminus \{a\} : \kappa(a, b)\kappa(a, c) < 0 \right\} \right|, \tag{7.53}$$

where A is the group of agents whose behavior the feature captures, $\mathcal{G}$ is a set of groups of agents, and $\kappa(a, b)$ is defined in Equation (7.41). Specifically, the feature counts the number of times agents of group A move through other groups. The feature allows our approach to learn to which extent the agents avoid to move through groups, accepting detours instead.

7.4.3.3 Cost of Trajectories

Furthermore, to model the agents' discrete navigation decisions, we propose to capture the cost of the composite trajectories of homotopy class ψ with respect to the learned continuous probability distributions $p_{\psi_{\boldsymbol{\theta}_\mathbf{f}}}$. Specifically, we propose to consider a feature

$$g_{\text{cost}}(\psi) := \mathbb{E}_{p_{\psi_{\boldsymbol{\theta}_\mathbf{f}}}}[c_{\boldsymbol{\theta}_\mathbf{f}}] \tag{7.54}$$

with

$$c_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}) := \boldsymbol{\theta}_\mathbf{f}{}^\top \mathbf{f}(\boldsymbol{\tau}) \tag{7.55}$$

that captures the expected cost of the composite trajectories of homotopy class ψ. Alternatively, the feature could capture the cost of the composite trajectory $\boldsymbol{\tau}^\star$ of homotopy class ψ, leading to

$$g_{\text{cost}}(\psi) := \boldsymbol{\theta}_\mathbf{f}{}^\top \mathbf{f}(\boldsymbol{\tau}^\star) \tag{7.56}$$

with

$$\boldsymbol{\tau}^\star := \operatorname*{argmax}_{\boldsymbol{\tau} \in \psi} p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau}). \tag{7.57}$$

These features allow the model to reason about the homotopy classes in terms of the cost of the composite trajectories according to the learned distribution $p_{\boldsymbol{\theta}_\mathbf{f}}(\boldsymbol{\tau})$.

7.4.4 Computing Feature Expectations

Maximizing the likelihood of the model parameters $\boldsymbol{\theta}_\mathbf{g}$ given the training data to learn a model of the agents' discrete navigation behavior requires computing the expected feature values

$$\mathbb{E}_{P_{\boldsymbol{\theta}_\mathbf{g}}}[\mathbf{g}] = \sum_{\psi \in \Psi} P_{\boldsymbol{\theta}_\mathbf{g}}(\psi)\mathbf{g}(\psi) \tag{7.58}$$

with respect to the discrete probability distribution $P_{\boldsymbol{\theta}_\mathbf{g}}(\psi)$. Summing over the entire space Ψ of homotopy classes, however, can be computationally prohibitory, since the number of homotopy classes scales exponentially with the number of agents and static obstacles.

Fortunately, only a few homotopy classes constitute the majority of the probability mass of the distribution $P_{\boldsymbol{\theta}_\mathbf{g}}(\psi)$ in most practical situations. For example, homotopy classes that entail a long detour around obstacles might be highly unlikely and for this reason might have only limited effect on the feature expectations. We therefore propose to

Algorithm 5 Compute the subset $\Psi' \subseteq \Psi$ of homotopy classes.

Input: set $A = \{a_i\}_{i=1}^{n}$ of agents
Output: set $R = \{\boldsymbol{\tau}_i\}_{i=1}^{|\Psi'|}$ with $\boldsymbol{\tau}_i \in \psi_i$ for each $\psi_i \in \Psi'$

for all agents $a_i \in A$ **do**
  $\tau^{a_i} \leftarrow$ trajectory from start position to target position of agent a_i
end for
$\boldsymbol{\tau} \leftarrow (\tau^{a_1\top}, \dots, \tau^{a_n\top})^\top$
$R \leftarrow \{\boldsymbol{\tau}\}$
while there are still unresolved potential maneuvers in R **do**
  for all $\boldsymbol{\tau} \in R$ **do**
    for all agents a, b with $a \neq b$ **do**
      if a and b get close according to $\boldsymbol{\tau}$ **and** $\boldsymbol{\tau}_{\pm 2\pi^{ab}} \notin R$ **then**
        $R \leftarrow R \cup \{\boldsymbol{\tau}_{\pm 2\pi^{ab}}\}$
      end if
    end for
  end for
end while
return R

approximate the feature expectations $\mathbb{E}_{P_{\boldsymbol{\theta}_\mathbf{g}}(\psi)}[\mathbf{g}]$ based on a subset $\Psi' \subseteq \Psi$ of homotopy classes. The approximation is then given by

$$\mathbb{E}_{P_{\boldsymbol{\theta}_\mathbf{g}}}[\mathbf{g}] \approx \hat{\mathbf{g}}_{\boldsymbol{\theta}_\mathbf{g}} := \sum_{\psi \in \Psi'} \hat{P}_{\boldsymbol{\theta}_\mathbf{g}}(\psi)\mathbf{g}(\psi), \tag{7.59}$$

where

$$\hat{P}_{\boldsymbol{\theta}_\mathbf{g}}(\psi) = \frac{1}{\hat{Z}(\boldsymbol{\theta}_\mathbf{g})} \exp\left(-\boldsymbol{\theta}_\mathbf{g}{}^\top \mathbf{g}(\psi)\right) \tag{7.60}$$

with

$$\hat{Z}(\boldsymbol{\theta}_\mathbf{g}) = \sum_{\psi \in \Psi'} \exp\left(-\boldsymbol{\theta}_\mathbf{g}{}^\top \mathbf{g}(\psi)\right). \tag{7.61}$$

Algorithm 5 sketches the heuristic to compute the subset $\Psi' \subseteq \Psi$ that our implementation uses to mitigate the computational burden. The algorithm computes the subset Ψ' in terms of a set R of composite trajectories that contains a representative composite trajectory for each homotopy class $\psi \in \Psi'$, which then can be used to evaluate the feature values $\mathbf{g}(\psi)$. The algorithm initializes the set R with a composite trajectory that comprises coarse trajectories of the agents from their start positions to their target positions. These coarse trajectories ignore collisions and are individually planned by means of a simple A*-planner. For each pair of agents a and b that get close to each other at any point in time according to the composite trajectory $\boldsymbol{\tau}$, the algorithm computes the composite trajectory $\boldsymbol{\tau}_{\pm 2\pi^{ab}}$ where the agents a and b pass each other on the other side. The algorithm then adds this composite trajectory to the set Ψ'.

In Kuderer et al. [108], we present an efficient approach to compute a set of trajectories of the homotopy classes induced by static obstacles in the environment, which, however, is not part of this thesis.

7.5 Applications to Mobile Robot Navigation

In order to behave in a socially compliant way, a mobile robot that navigates in an environment populated by humans needs to reason about the humans' intentions. The approach presented in this chapter enables the robot to learn a probabilistic model of the humans' navigation behavior. The robot can then use that model to predict the trajectories of nearby pedestrians and behave accordingly. Humans that navigate in the vicinity of the robot, however, react to the actions of the robot, which is why the robot might need to adapt its behavior, which in turn might affect the humans. To break up this infinite loop and to model cooperative behavior, we propose to jointly reason about the trajectories of the pedestrians and the robot, which means that the robot implicitly "predicts" its own trajectory in the context of the humans and then behaves according to its own predictions. Hence, we integrate the probabilistic navigation behavior model into a unified framework for mobile robot navigation.

Our model specifically predicts the trajectories of the pedestrians and the trajectory of the robot in terms of a joint probability distribution over the trajectories of the pedestrians and the robot. One approach to use the probabilistic model for mobile robot navigation is to have the robot behave according to a composite trajectory sampled from the probability distribution. This approach makes the robot imitate the learned behavior, including the observed stochasticity of the trajectories. During navigation, the robot needs to constantly update the prediction with the actions of the pedestrians. As a result, the robot needs to constantly sample new composite trajectories from an updated probability distribution that reflects the actions of the pedestrians and the robot. Sampling from the high-dimensional distribution over composite trajectories in real-time during navigation, however, does currently not seem to be feasible with a modern desktop computer. In our experiments, we therefore use a different approach to integrate the predictive model into the navigation system. Instead of sampling from the distribution, we have the robot behave according to the mode of the most likely homotopy class, which we can efficiently compute in real-time during navigation.

7.6 Experimental Evaluation

We carried out an extensive set of experiments to evaluate the approach presented in this chapter. To this end, we recorded several datasets of human trajectories using human motion capture. We then used our approach to learn a model of the navigation behavior of the humans from the recorded trajectories. To assess how well the models learned by our approach generalize to new situations, we carried out a cross-validation. We compare the effects of the two techniques that we propose to approximate the feature expectations,

namely the Dirac approximation (Dirac) and the sampling-based approximation (MCMC). We compare our approaches to the social forces method (SF) presented by Helbing and Molnár [69] and to the reciprocal velocity obstacles (RVO) method introduced by van den Berg et al. [167]. We furthermore conducted a Turing test with human subjects to evaluate whether the trajectories predicted by our approach are in fact perceived as human. Moreover, we present experiments with mobile robots that use our navigation model to navigate in populated environments. Specifically, we had a mobile robot learn a model of the navigation behavior of nearby pedestrians that the robot observed during navigation with its on-board laser range finders and then navigate using that model.

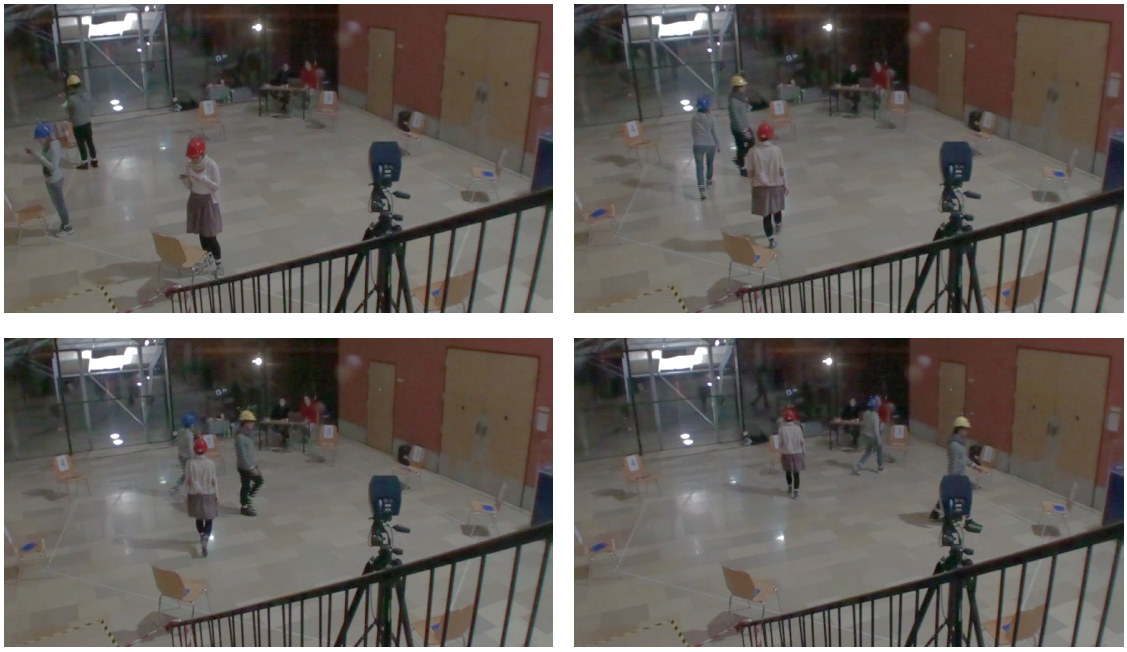
7.6.1 Acquiring Training Data

We recorded two datasets of trajectories of human subjects with an optical motion capture system [7] to fit our model to human navigation behavior. To prevent the subjects from focusing on their walking behavior, which may result in awkward trajectories, we told the them that we were conducting psychological research on the effects on attention when frequently changing locations. To further distract them, we made the subjects read and memorize newspaper articles at various locations. We set up eight differently numbered locations where each location was equipped with a different newspaper article. We repeatedly asked the three subjects to read the articles and simultaneously change their locations such that their paths crossed each other, which forced the subjects to avoid each other. The setup gave rise to eight different scenarios that repeatedly occurred after each cycle. During the experiments, we used a Motion Analysis motion capture system with eight digital Raptor-E cameras to track retroreflective markers that were attached to hats worn by the subjects. After the experiments, the subjects in fact reported that the cognitive load imposed by the memory task effectively prevented them from questioning the actual motivation of the experiment. Figure 7.6 shows the human subjects trying to memorize the newspaper articles during the first data acquisition. In addition to that, the figure shows the trajectories recorded while the subjects moved to the subsequent locations. Figure 7.7 gives an insight into the second data acquisition. Both datasets comprise one hour of human interaction each.

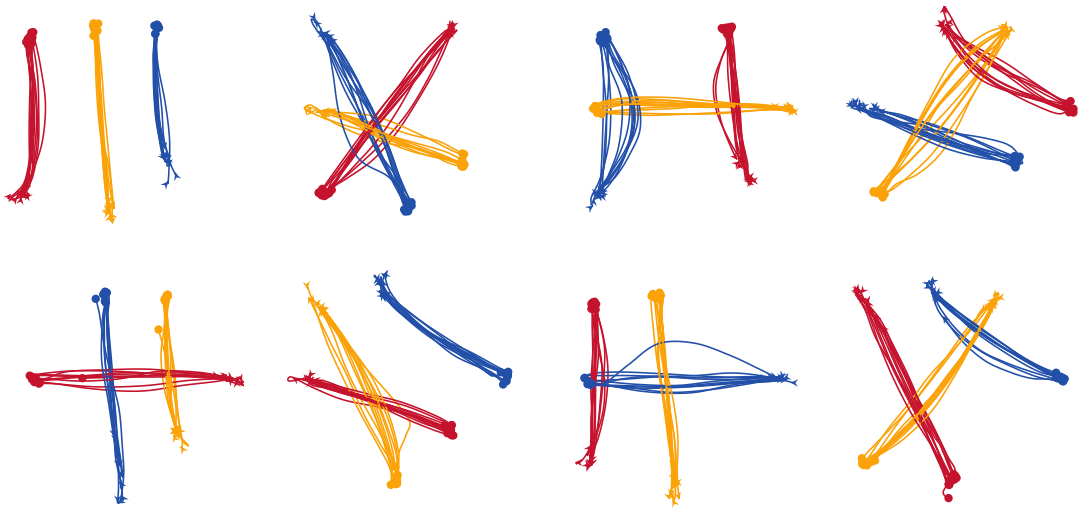
7.6.2 Cross-Validation

Cross-validation is a technique to evaluate how well a predictive model generalizes to new data. To this end, cross-validation partitions the available data into a training set and a testing set and evaluates how well a model that has been learned based on the training set performs on the test set. In k-fold cross-validation, the data is partitioned into k subsets of the same size. For each of these subsets, a cross-validation is performed, where the subset is used as the testing dataset, and the remaining $k - 1$ subsets are used to train the model.

To evaluate how well the models learned by our approach generalize to yet unseen situations, we conducted a five-fold cross-validation on the datasets described in Section 7.6.1.



(a) Snapshots of the human subjects avoiding each other on their way to their next target locations.



(b) Recorded trajectories grouped according to the eight scenarios.

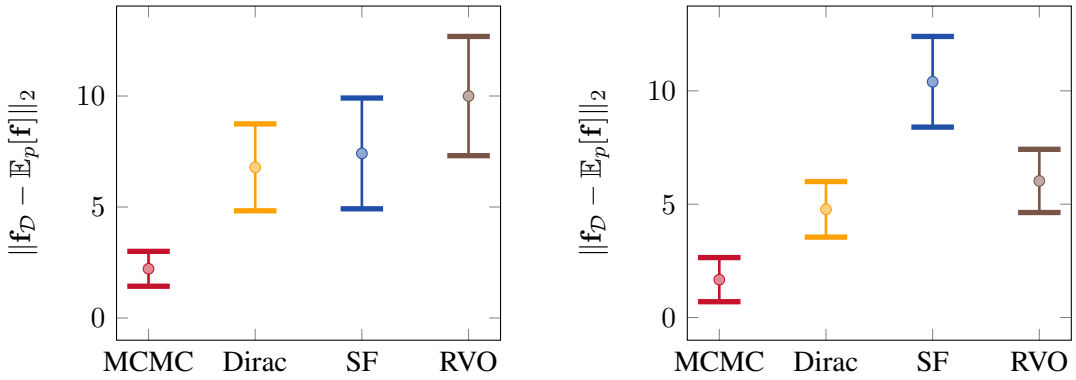
Figure 7.6: Recording real-world training data of three human subjects avoiding each other in order to proceed to their next target location. We recorded one hour of human interactions by tracking markers that were attached to hats worn by the subjects with a motion capture system. To prevent the subjects from focusing on their walking behavior, which may result in awkward trajectories, we made them read and memorize newspaper articles at various locations, saying that we were conducting psychological research on the effects on attention when frequently changing locations.

Figure 7.7: Recording real-world training data of four persons in a test environment with a motion capture system. The experimental setup was similar to the setup described in Figure 7.6.

We compared our approaches with the social forces algorithm and the reciprocal velocity obstacles method. To allow for a fair comparison, we used the same set of features for all the methods. To this end, we optimized the parameters of SF and RVO based on the training set of each fold. We applied stochastic gradient descend to minimize the norm of the discrepancy between the empirical feature values and the feature values induced by the methods. We additionally evaluated the parameters provided by Helbing and Molnár [69] and Guy et al. [63]. However, the learned parameters turned out to perform better. We assumed that the target positions of the agents are the positions last observed in the datasets.

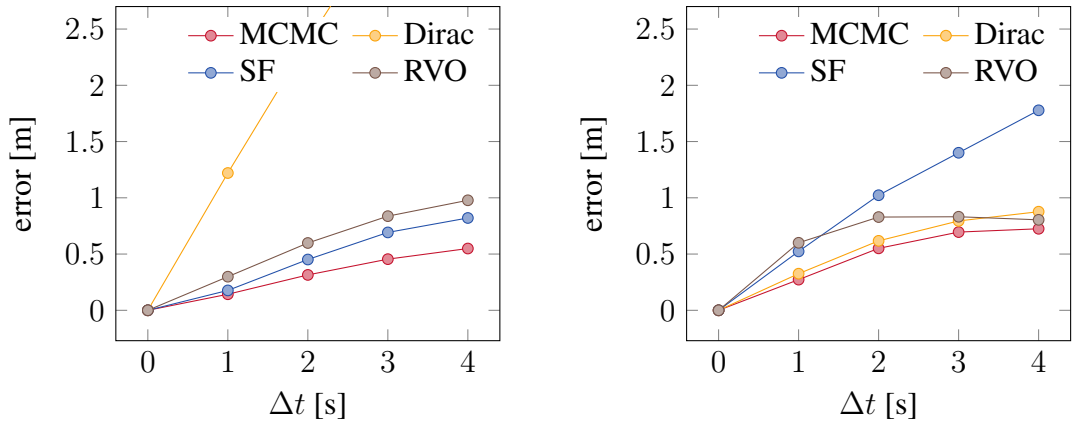
Figure 7.8 and Figure 7.9 depict the results of the five-fold cross-validation in terms of the features and in terms of the Euclidean error of the trajectories, respectively. The results suggest that models learned by our method based on sampling outperform the other methods in both aspects. The Dirac approximation leads to a higher error compared to the sampling-based method because of a bias of the model towards samples from highly unlikely homotopy classes, since the Dirac approximation under-estimates the feature expectations.

Our method explicitly accounts for the discrete navigation decisions of the agents by computing a probability distribution over the outcomes of these decisions. Figure 7.10 compares the distribution estimated by our approach with the frequencies observed in the motion capture data. The figure suggests that our approach is able to predict the distribution over the homotopy classes fairly accurately. In contrast to that, the estimates



(a) Dataset described in Section 7.6.1. **(b)** Dataset provided by Pellegrini et al. [143].

Figure 7.8: Feature discrepancy according to five-fold cross-validations on two datasets. MCMC refers to our approach based on MCMC sampling, Dirac refers to our approach based on a Dirac approximation, SF refers to the social forces model [69], and RVO refers to the reciprocal velocity obstacles method [167]. The results suggest that our approach based on MCMC sampling best captures pedestrian navigation behavior in terms of the features on both datasets.



(a) Dataset described in Section 7.6.1. **(b)** Dataset provided by Pellegrini et al. [143].

Figure 7.9: Position error in meters in the predictions according to five-fold cross-validations on two datasets. MCMC refers to our approach based on MCMC sampling, Dirac refers to our approach based on a Dirac approximation, SF refers to the social forces model [69], and RVO refers to the reciprocal velocity obstacles method [167]. The results suggest that our approach based on MCMC sampling best captures pedestrian navigation behavior in terms of prediction errors on both datasets.

of the social forces method are deterministic.

Figure 7.11 gives some insights into the learning phase of our algorithm. The figure evaluates the evolution of the discrepancy between the empirical feature values and the feature expectations while our learning algorithm refines the model parameters using RPROP. The figure averages the values over all folds of the k-fold cross-validation. Specifically, Figure 7.11(a) analyzes how our approach learns the agents' continuous

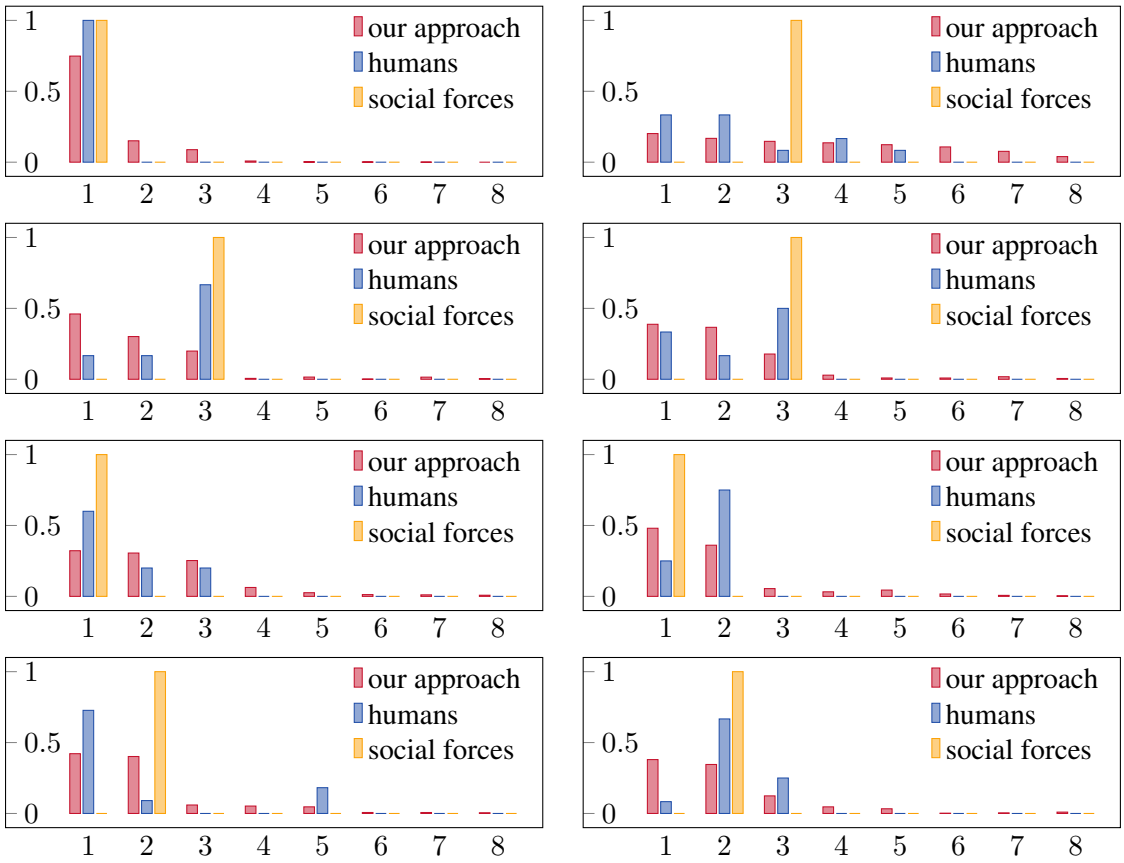


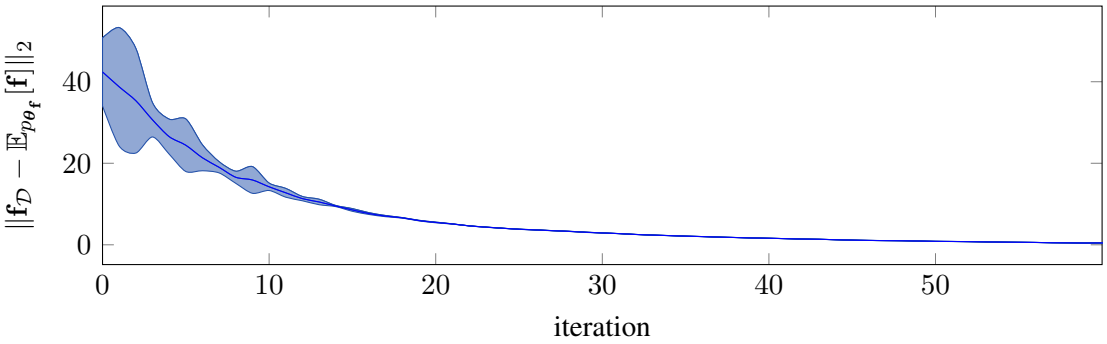
Figure 7.10: Probability distributions over the eight homotopy classes induced by the eight scenarios depicted in Figure 7.6 as estimated by our approach and the social forces method. The plots suggest that the distributions estimated by our approach are fairly close to the frequencies recorded in the motion capture system. The predictions of the social forces method are deterministic.

navigation decisions, and Figure 7.11(b) analyzes how our approach learns the agents' discrete navigation decisions. The results clearly suggest that our method is able to replicate the observed behavior in terms of the features.

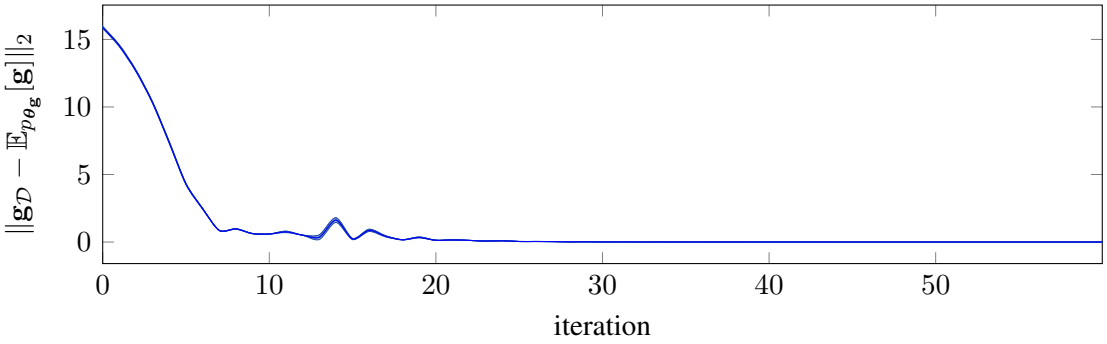
7.6.3 Turing Test

When using models of human navigation behavior to simulate agents in the context of computer graphics, it is often desirable that the generated trajectories are perceived as human. In other applications, such as autonomous driving, a robot is expected to navigate in a human-like way such that its actions match human expectations. To evaluate how human-like the trajectories generated by our approach are, we carried out a Turing test. Although the term Turing test stems from a more specific proposal by Turing [166], it is nowadays commonly referred to a test that evaluates the ability of a machine to exhibit intelligent behavior indistinguishable of that of a human.

We asked ten human subjects to distinguish recorded human behavior from behavior



(a) Learning a model of the agents' continuous decisions, which affect physical properties of their trajectories such as velocities and clearances to obstacles.

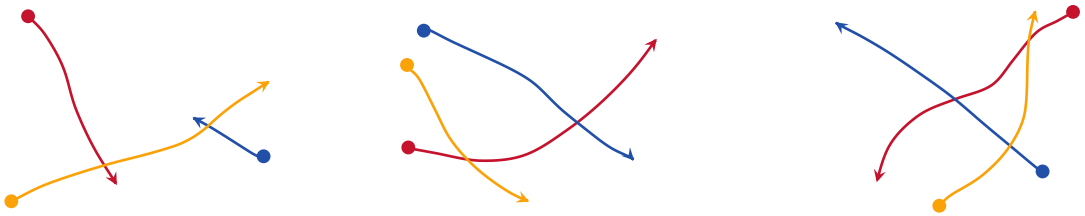


(b) Learning a model of the agents' discrete decisions, which determine the homotopy classes of the resulting composite trajectories.

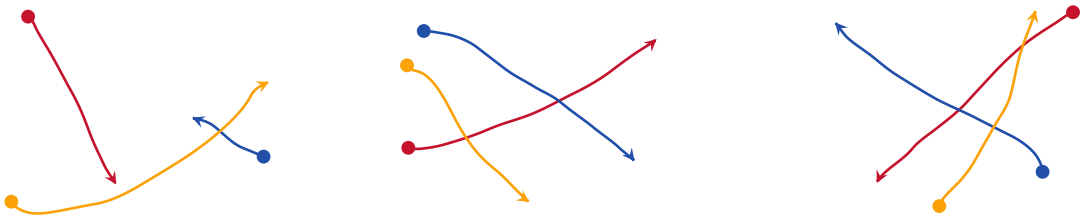
Figure 7.11: Evolution of the Euclidean norm and variance of the discrepancy between the empirical feature values and the feature expectations of the model while learning the model parameters $\boldsymbol{\theta}_\mathbf{f}$ and $\boldsymbol{\theta}_\mathbf{g}$, averaged over the folds of a five-fold cross-validation.

generated by several algorithms. We evaluated how well the subjects performed on a set of runs that was randomly drawn from recorded human demonstrations. We showed them animations of trajectories, such as the trajectories depicted in Figure 7.12, that were either recorded from the human demonstrations or from the prediction of one of the algorithms. Specifically, we presented 40 runs to each of the human subjects, where the trajectories were equally drawn from the human demonstrations, from the predictions computed by our approaches or from the predictions computed by the social forces method presented by Helbing and Molnár [69]. A comparison with the social forces method is interesting since many approaches to visualize human crowds, such as Sud et al. [159] and Pelechano et al. [142], rely on this algorithm.

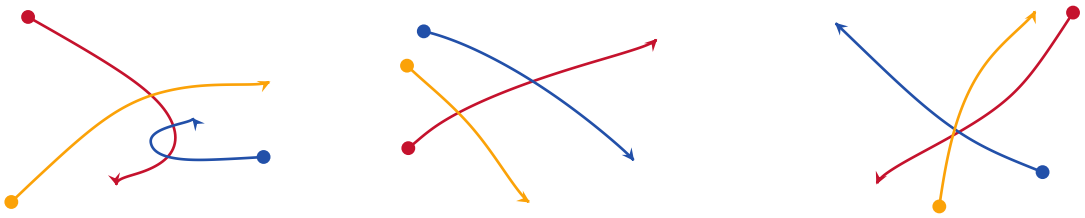
Figure 7.13 summarizes the results of the Turing test. The human subjects correctly identified 79 % of all the human demonstrations, but they mistook 68 % of the predictions of our MCMC sampling-based approach, 40 % of the predictions of our Dirac approximation-based approach, and 35 % of the predictions of the social forces algorithm



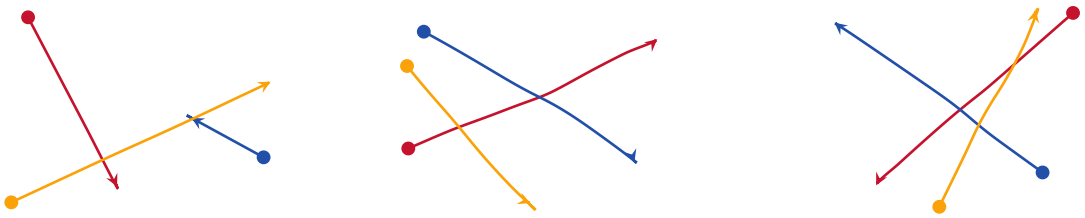
(a) Human trajectories as recorded with a motion capture system.



(b) Trajectories predicted by a model based on MCMC sampling.



(c) Trajectories predicted by a model based on the Dirac approximation.



(d) Trajectories predicted by the social forces method.

Figure 7.12: Example trajectories of three pedestrians as recorded with a motion capture system and as predicted by various models. (a): Trajectories recorded with a motion capture system in the test environment shown in Figure 7.6. (b): Samples drawn from our MCMC sampling-based model replicate the stochasticity of the observed trajectories. (c): Learning a model using the Dirac approximation can lead to a bias towards samples from highly unlikely homotopy classes. (d): The social forces model yields deterministic predictions.

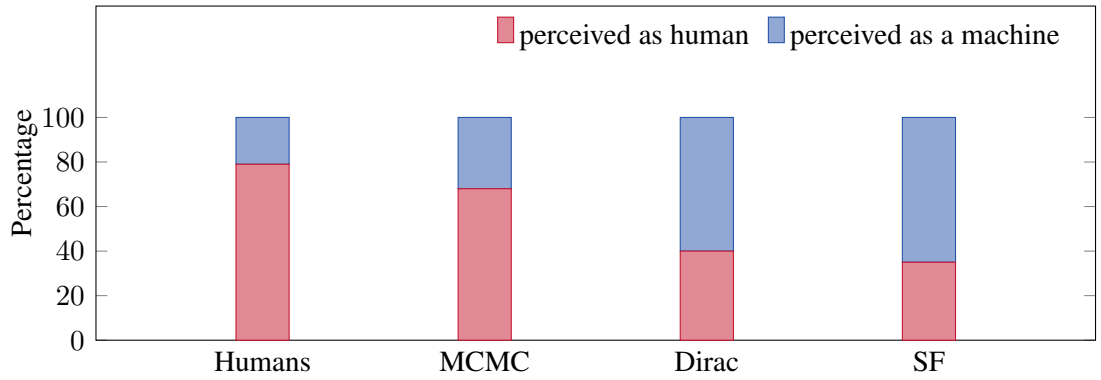


Figure 7.13: Results of a Turing test that we conducted to evaluate whether the trajectories induced by our two approaches (MCMC and Dirac) and by the social forces model (SF) presented by Helbing and Molnár [69] qualify as human. The results of the Turing test suggest that the behavior induced by our MCMC sampling-based approach most resembles human behavior.

for human behavior. In summary, the results of the Turing test indicate that the behavior induced by our MCMC sampling-based approach is significantly more human-like than the behavior induced by the other two methods according to a one-sided paired sample t-test at a 95 % confidence level.

7.6.4 Optimizing Composite Trajectories

As described in Section 7.3.3, we apply gradient-based optimization to optimize the composite trajectories with respect to the continuous maximum entropy distribution given in Equation (7.15). We empirically evaluated the performances of several gradient-based optimization techniques. Specifically, we compared gradient ascent, the Levenberg-Marquardt algorithm [127], and the RPROP algorithm [151]. Our experiments suggested that the latter performs best at this task. Hence, we used RPROP in all our experiments.

Our implementation optimizes the composite trajectories of several homotopy classes in parallel. One iteration to optimize 60 spline control points takes approximately 1 ms on a standard desktop computer. Figure 7.14 visualizes the evolution of the error in each iteration during trajectory optimization. Our experiments suggest that our approach allows a mobile robot to predict human navigation behavior online in situations similar to our experiments. Most of the computation time is spent to compute the gradients of the spline control points with respect to the density function, which is shaped by the features.

We require the gradient-based optimization to converge to a unique maximum within each homotopy class. To evaluate the convergence of the trajectory optimization, we considered one of the datasets that we recorded using a motion capture system. Based on a set of randomly sampled initial guesses, we optimized the composite trajectories. For each homotopy class, our algorithm found a unique solution irrespective of the initial guess with a maximal deviation of less than 2 cm. This suggests that our algorithm indeed finds the modes of the probability distribution corresponding to the homotopy classes.

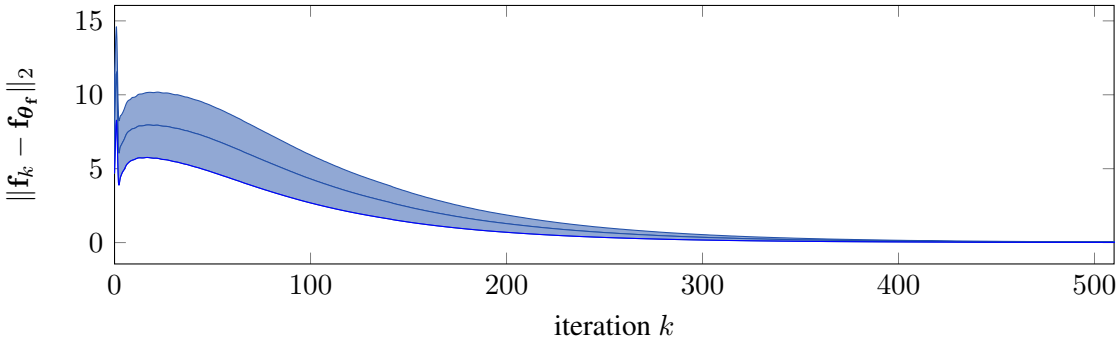


Figure 7.14: Optimizing composite trajectories given the feature weights $\boldsymbol{\theta}_\mathbf{f}$ using RPROP. The plot shows the evolution of the mean and the standard deviation of the Euclidean distance between the current feature values $\mathbf{f}_k$ and the resulting feature values $\mathbf{f}_{\boldsymbol{\theta}_\mathbf{f}}$ after convergence, averaged over 100 randomly generated scenarios. Our approach carries out each iteration to optimize the trajectories of three agents comprising 60 spline control points in approximately 1 ms on a standard desktop computer.

7.6.5 Mobile Robot Navigation

We demonstrate that our approach enables a mobile robot to autonomously navigate in a socially compliant way. We specifically assess the ability of our method to allow a mobile robot to learn a model of the navigation behavior of pedestrians by observing the pedestrians during navigation. We evaluated our approach using two distinct mobile robot setups. Whereas the first setup relies on an external optical motion capture system to localize the robot and the pedestrians, the second system solely relies on sensors that are mounted on the robot to localize the robot and to detect and track pedestrians.

The first setup is based on the custom-built wheeled robot depicted in Figure 7.15(a). The robot is controlled by a standard laptop computer that is mounted on the robot. A Motion Analysis optical motion capture system localizes the robot and the pedestrians. Specifically, eight digital Raptor-E cameras track retroreflective markers mounted on the robot and on hats worn by the pedestrians. Our predictive model runs on a standard desktop computer, which is connected to the laptop computer via a wireless local area network.

The second setup is based on an altered Permobil C500 wheelchair, which is depicted in Figure 7.15(b). The wheelchair is equipped with wheel encoders and two Hokuyo UTM-30LX laser range finders for localization and perception. A small desktop computer that is mounted on the robot carries out all the computations necessary for navigation in populated environments, including localization, perception, and behavior prediction. As opposed to the robot depicted in Figure 7.15(a), the robot depicted in Figure 7.15(b) is completely autonomous and does neither rely on any external computers nor external sensors.

Prior to each experiment, the robot used a graph-based approach to simultaneous localization and mapping to build a grid map of the experimental environment. In Chapter 3, we presented techniques to improve mapping, especially in settings where the

(a) Custom-built wheeled robot. **(b)** Robotic wheelchair.

Figure 7.15: The mobile robot platforms used in the experiments presented in this chapter.

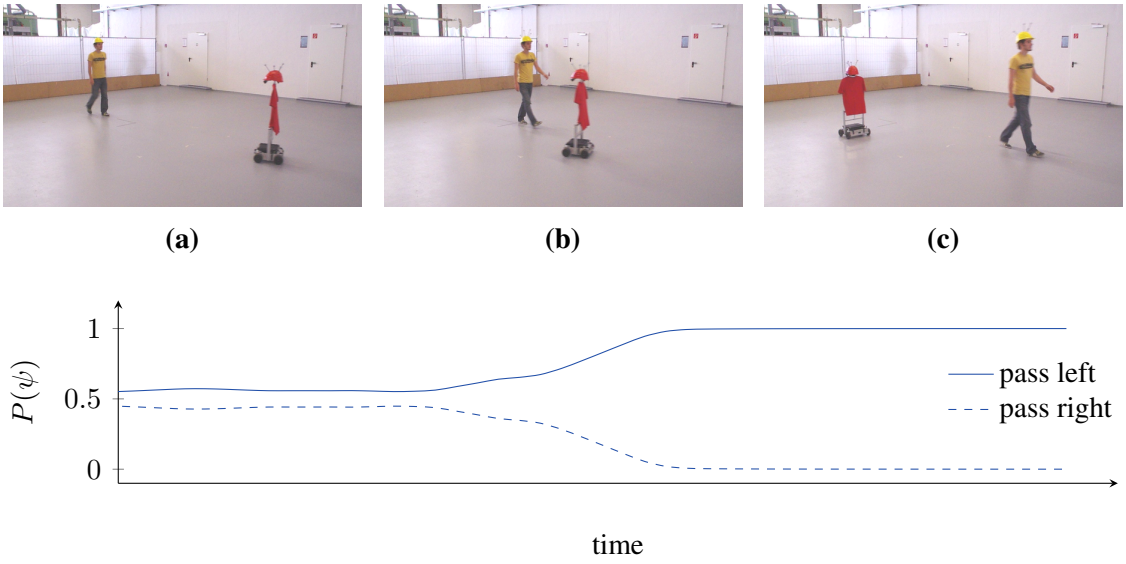
robot spends a lot of time revisiting the same areas. During the experiments, the robot relied on laser-based Monte Carlo Localization [163] using the computed maps of the environments.

We implemented a laser-based perception system that allows the robot to spot and track pedestrians in its vicinity. Our method extracts objects in the laser readings that resemble pedestrians and then relies on a nearest neighbor filter and gating to assign these observations to existing tracks of pedestrians, or to a new track if the observation cannot be assigned to any of the existing tracks. To prevent false positive detections, our method disregards laser observations that might belong to occupied space according to the static map. In Chapter 6, we present techniques to improve pedestrian tracking, especially in large environments, and to handle occlusions.

We implemented our navigation approach in C++. Running on standard desktop computers, our implementation predicts the trajectories of nearby pedestrians at 5 Hz, which allows the robots to quickly react to changing situations. Our implementation thereby optimizes the composite trajectories of various homotopy classes in parallel. As discussed in Section 7.3.1, our method represents the agents' trajectories using splines, where spline control points are placed approximately every $\Delta t = 1\,\mathrm{s}$, which seems to be a reasonable tradeoff between accuracy and computational efficiency.

7.6.5.1 Passing a Pedestrian

In a first experiment, the mobile robot depicted in Figure 7.15(a) and a pedestrian pass each other while moving to their target positions on the opposite sides. We tracked the



(d) Evolution of the probability distribution over the two homotopy classes during the encounter.

Figure 7.16: Example encounter of a robot and a human where the robot can pursue its initial intent to pass on the left side.

poses of the robot and the human using the optical motion capture system. Figure 7.16 and Figure 7.17 show two example encounters. The figures visualize the evolution of the estimated probability distribution of the homotopy classes during the encounters.

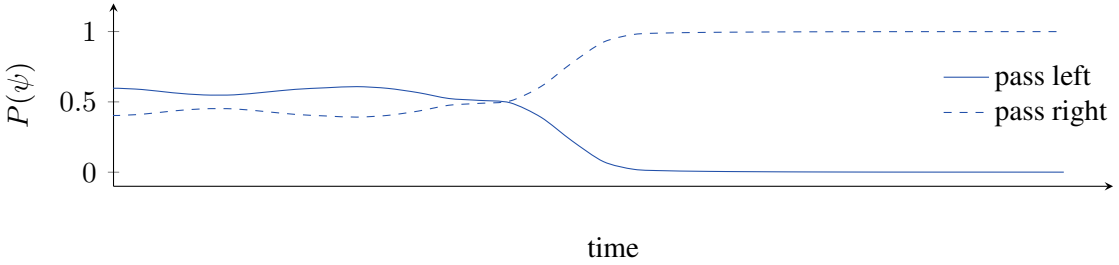
In the first encounter depicted in Figure 7.16, the belief of the robot reflects the symmetry of the situation. Both homotopy classes, i.e., passing left and passing right, seem more or less equally likely to the robot in terms of its probability distribution estimate over the corresponding composite trajectories. Nevertheless, a slight tendency towards the left side makes the robot commit to this homotopy class. As a consequence, the robot expects the human to evade to the opposite side. The human, however, refuses to evade to any side but instead walks on a straight line towards his target position. The robot adapts its expectations to the new situation but still decides to pass on the left side. After the situation has been resolved, the robot directly proceeds to its target position.

The second encounter depicted in Figure 7.17 illustrates a similar initial situation. In this situation, however, the human insists on passing on the right side regardless of the robot's visible intent to pass left. The robot thus re-evaluates the probability distribution over the topological variants and decides to pass on the other side.

7.6.5.2 Imitating Navigation Behavior

In a second experiment, we evaluated the ability of our approach to learn and then imitate navigation behavior by demonstration. We taught the robot depicted in Figure 7.15(b) two distinct navigation behaviors by teleoperation. The robot observed nearby pedestrians with its on-board laser range finders. As a result, their trajectories are only partially captured owing to occlusions and the limited field of view of the robot's laser range

(a) **(b)** **(c)**



(d) Evolution of the probability distribution over the two homotopy classes during the encounter.

Figure 7.17: The robot changes its initial intent to pass on the left side because the human's actions differ from the robot's initial predictions.

finders.

In a first setting, we joysticked the robot in a rather aggressive way. During a 5 minute training phase, we specifically taught the robot not to evade pedestrians. See Figure 7.18 for some sample demonstrations. The robot learned a model of the demonstrated navigation behavior and then used that model to navigate in a different building. Figure 7.19 depicts a sample run where the robot encounters a pedestrian in a corridor. The figure clearly shows that the robot has successfully learned to imitate the rather aggressive navigation behavior and does not evade the pedestrian.

In a second setting, we demonstrated a cautious, cooperative navigation behavior. We joysticked the robot in a way similar to the behavior that would be expected from a pedestrian. Specifically, the robot and the pedestrians mutually avoided each other during a 6 minute training phase. See Figure 7.20 for some sample demonstrations. The robot then learned a model of the demonstrated behavior and used it for navigation in a different building. See Figure 7.21 and Figure 7.22 for sample runs of encounters between the robot and pedestrians. The figures suggest that the robot has successfully learned the cautious, cooperative navigation behavior and is able to imitate the behavior, leading to natural, socially compliant mobile robot navigation.

7.7 Related Work

Over the past years, the robotics community has become increasingly interested in learning policies from demonstrations. Atkeson and Schaal [12] developed one of the pioneering approaches to infer a mapping from state features to actions to directly mimic

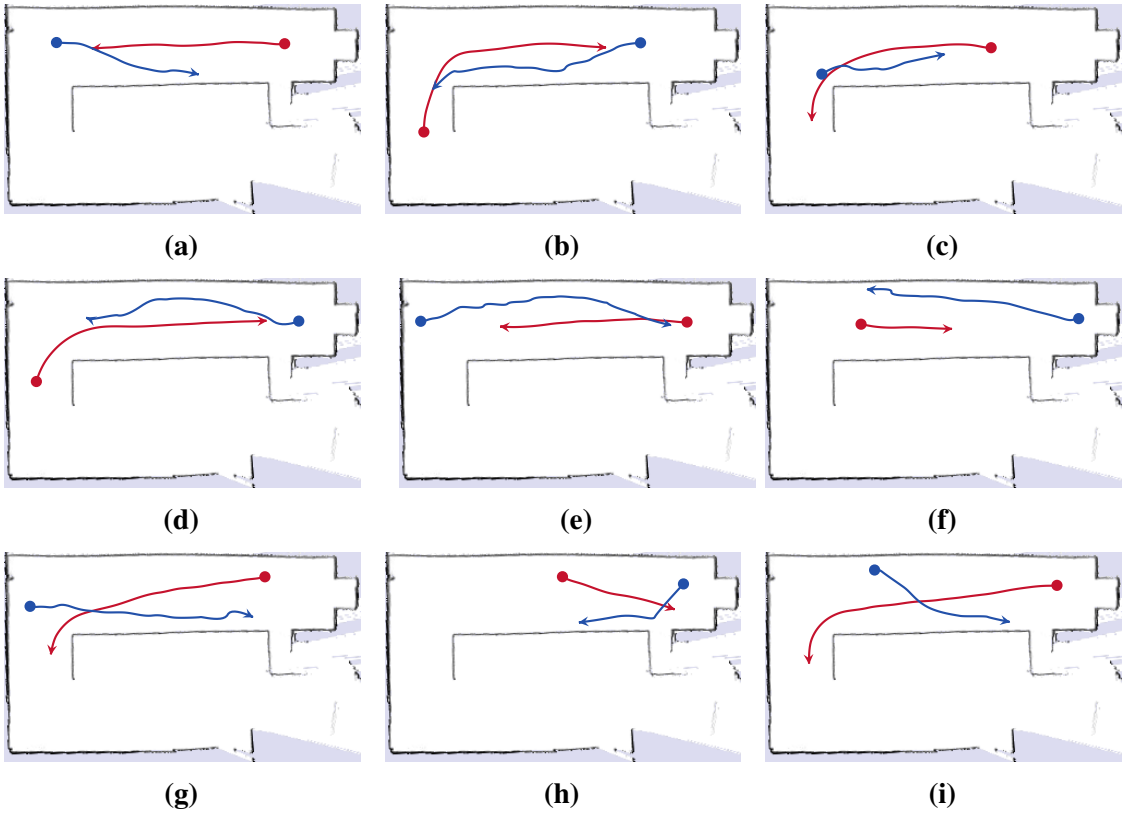
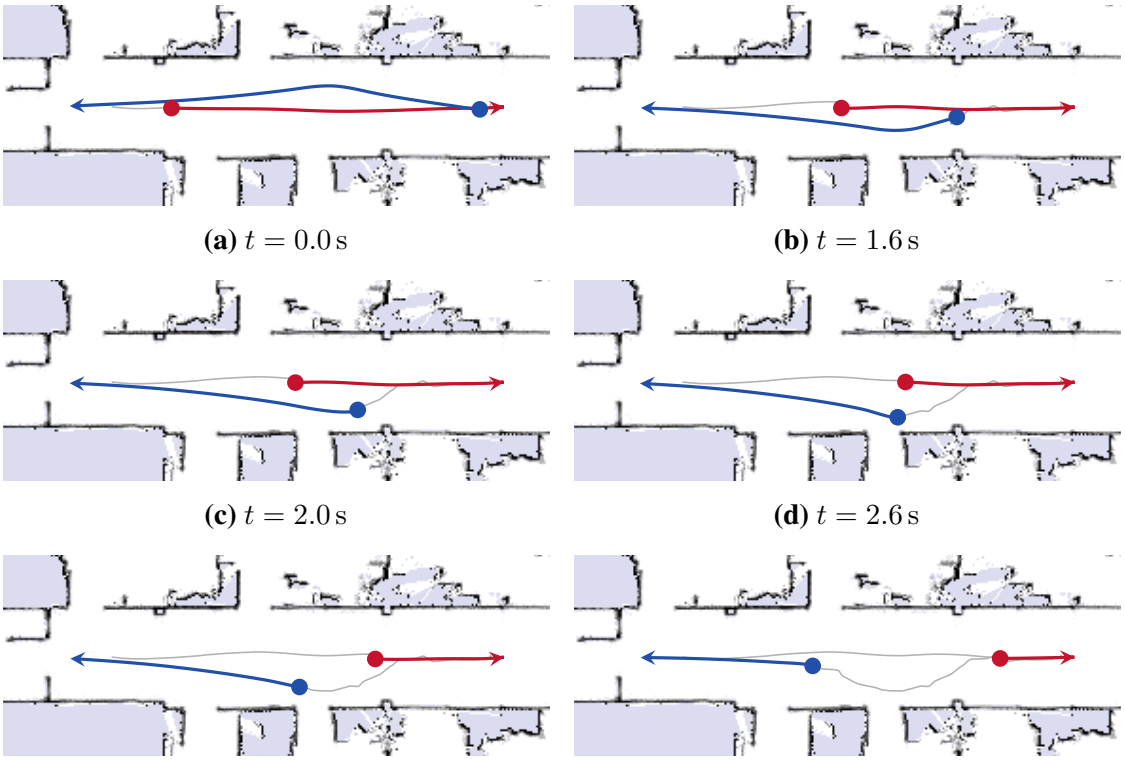


Figure 7.18: Teaching a robot an overly aggressive navigation behavior, where the robot depicted in red does not evade the pedestrians depicted in blue.

observed behavior. More recently, Ng and Russell [135] applied inverse reinforcement learning to recover a cost function that explains observed behavior. Abbeel and Ng [1] specifically suggest to match the values of features that capture relevant aspects of the behavior that is to be imitated. However, matching the feature values does not lead to a unique reward function. In contrast to that, maximum entropy inverse reinforcement learning, presented by Ziebart et al. [180], resolves this ambiguity by considering the policy with the highest entropy subject to the constraint that the feature values match in expectation.

Maximum entropy inverse reinforcement learning [180] works well in discrete state-action spaces of low dimensionality. However, discretizing the state-action spaces does not scale well to continuous trajectories, especially when taking into account higher-order dynamics such as velocities and accelerations. Amongst others, Ratliff et al. [147], Ziebart et al. [182], Henry et al. [70], and Vernaza and Bagnell [168] applied these indirect learning approaches to a variety of problems including route planning for outdoor mobile robots and learning pedestrian navigation behavior. For example, Kitani et al. [86] use maximum entropy inverse reinforcement learning to infer the preferences of pedestrians with respect to vision-based physical scene features such as sidewalks. Their model predicts the trajectory of a pedestrian taking into account these features using a discrete



(a) $t = 0.0\,\mathrm{s}$ **(b)** $t = 1.6\,\mathrm{s}$

(c) $t = 2.0\,\mathrm{s}$ **(d)** $t = 2.6\,\mathrm{s}$

(e) $t = 3.2\,\mathrm{s}$ **(f)** $t = 4.0\,\mathrm{s}$

Figure 7.19: A robot imitates an aggressive navigation behavior that it has observed earlier, where the robot shown in red does not evade pedestrians shown in blue.

Markov decision process. In contrast to that, Ziebart et al. [183] apply inverse reinforcement learning to predict the target of a partial pointing motion in graphical user interfaces. Their approach reasons about the trajectories of the pointer to the target in a continuous space in terms of velocities, accelerations, and jerks by formulating a linear dynamical system, which allows their method to perform efficient inference in closed-form.

Inspired by Abbeel and Ng [1] and Ziebart et al. [180], our approach aims to find maximum entropy distributions that match observed feature values in expectation. However, in contrast to the above-mentioned techniques, our approach reasons about joint probability distributions over trajectories of multiple agents to capture cooperative behavior. Moreover, our method reasons about the agents' trajectories in continuous state spaces including higher-order dynamics.

In many inverse reinforcement learning approaches, estimating the feature expectations is a challenging problem, especially in high-dimensional state spaces of continuous trajectories. To this end, Boularias et al. [24] apply importance sampling to compute the gradient of the model parameters while Vernaza and Bagnell [168] constrain the features to have a certain low-dimensional structure. Kalakrishnan et al. [82] propose to assume the demonstrations to be locally optimal and sample continuous trajectories by adding Gaussian noise to the model parameters. In Kuderer et al. [106], we approximate the

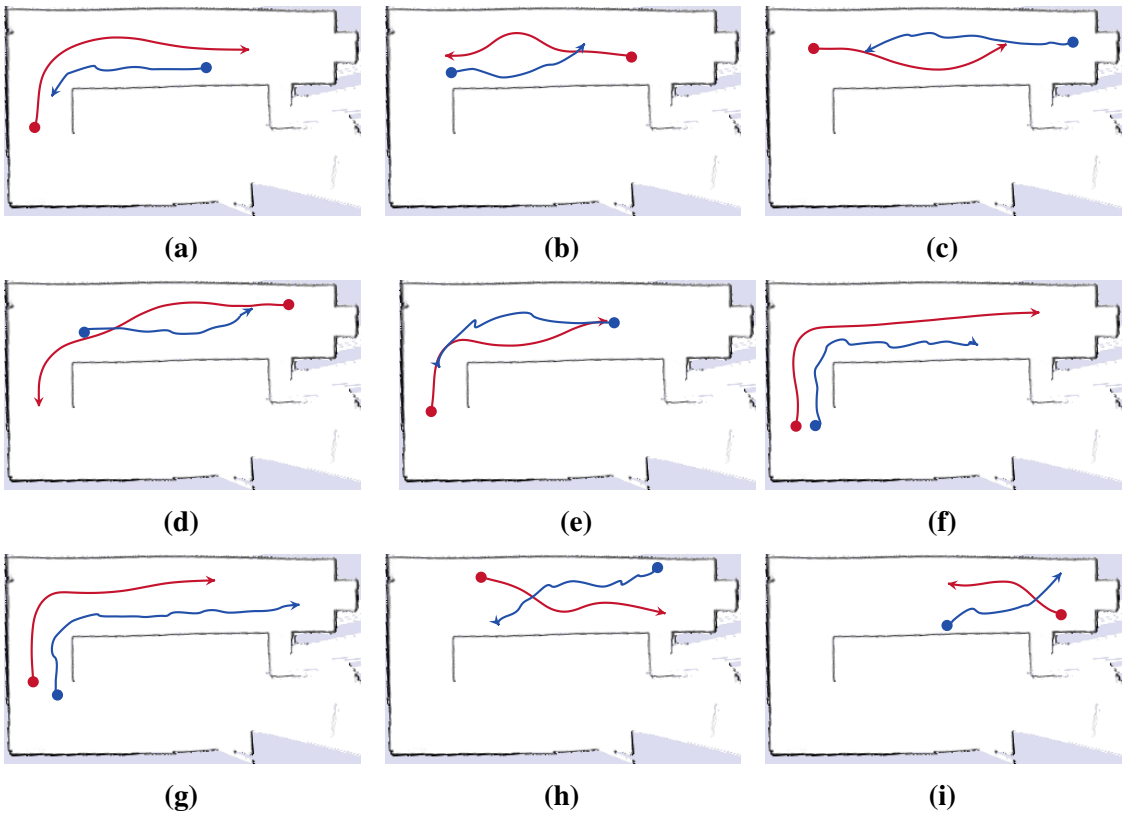
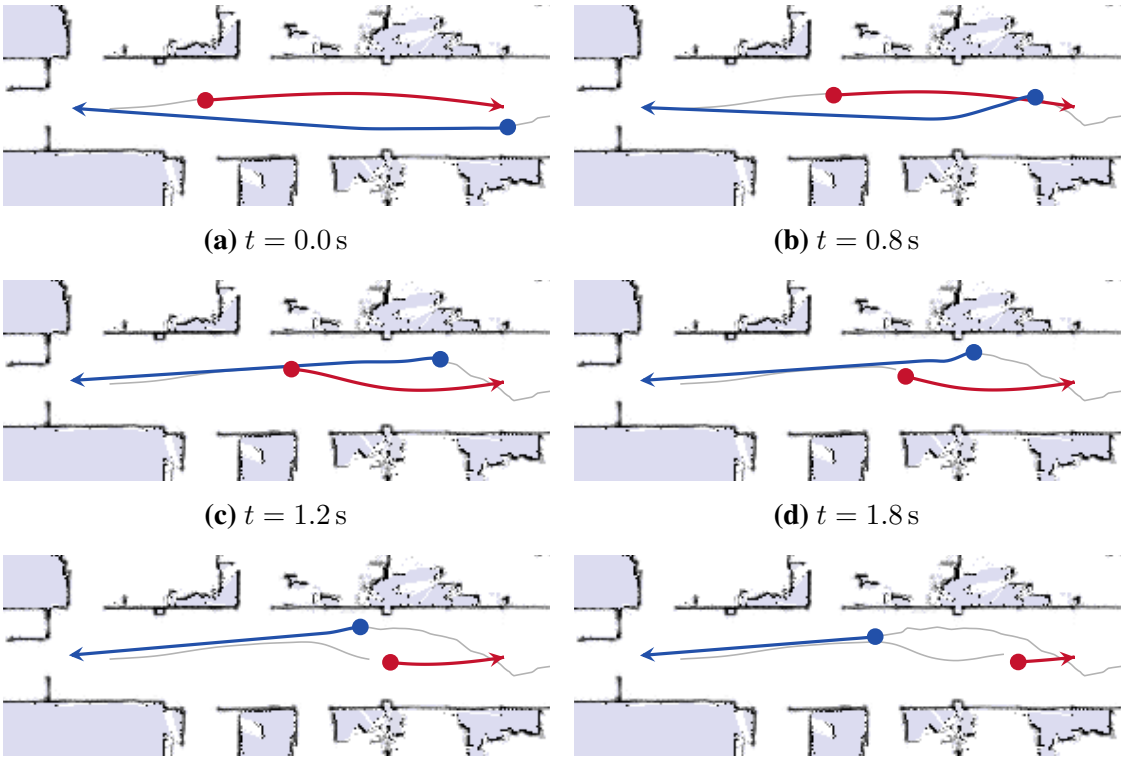


Figure 7.20: Teaching a robot a cautious, cooperative navigation behavior, where the robot depicted in red and the pedestrians depicted in blue mutually avoid each other.

feature expectations based on Dirac delta functions at the modes of the distribution. This approximation, however, leads to suboptimal models when learning from imperfect human navigation behavior since its stochasticity is not sufficiently captured. In practice, the method under-estimates the feature expectations and thus favors samples from homotopy classes that are highly unlikely according to the observed behavior. In contrast to that, in Kretzschmar et al. [105], we therefore model the navigation behavior in terms of a mixture distribution that allows us to estimate the feature expectations using Markov chain Monte Carlo sampling. Specifically, our representation of the probability distribution allows us to make use of the Hybrid Monte Carlo algorithm [42], which leads to highly efficient sampling applicable to arbitrary differentiable features. Hybrid Monte Carlo ideas are also used by Ratliff et al. [148] in the context of globally robust trajectory optimization.

As opposed to Abbeel and Ng [1] and Ziebart et al. [180], and, hence, in contrast to our approach, Dvijotham and Todorov [44] advocate addressing the problem of inverse reinforcement learning in a way that does not require solving the forward problem. They specifically estimate the value function instead of the reward function. Likewise, Levine and Koltun [116] circumvent the forward problem by relying on a local approximation of the reward function, which enables efficient learning in high-dimensional continuous



(a) $t = 0.0\,\mathrm{s}$ **(b)** $t = 0.8\,\mathrm{s}$

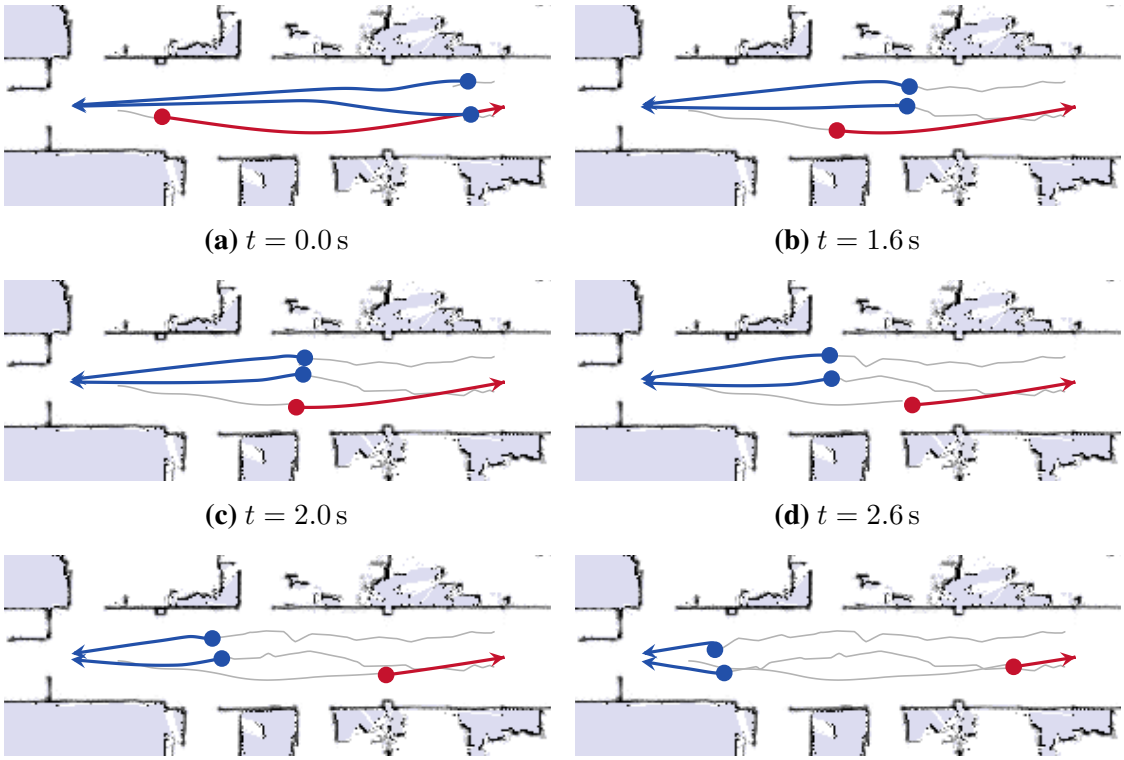
(c) $t = 1.2\,\mathrm{s}$ **(d)** $t = 1.8\,\mathrm{s}$

(e) $t = 2.4\,\mathrm{s}$ **(f)** $t = 3.2\,\mathrm{s}$

Figure 7.21: A robot imitates a cautious, cooperative navigation behavior that it has observed earlier. The robot shown in red and the pedestrians shown in blue mutually evade each other.

spaces. Their algorithm, however, assumes that the expert locally optimizes its actions instead of performing global optimization.

In contrast to the above-mentioned probabilistic learning approaches, many researchers have proposed models to capture the navigation behavior of pedestrians [22, 34, 64, 176, 177]. Zambrano et al. [178] divides these methods into steering models and optimization models. Steering models describe human navigation behavior as a dynamical system in which a set of rules determines the agent's immediate action given its current state in the environment [68, 69, 77, 115, 133, 172]. The widely-used social forces method presented by Helbing and Molnár [69] models pedestrian motion behavior in terms of forces that correspond to internal objectives of humans, such as the desire to reach a target and to avoid obstacles. Subsequently, several authors used parameter learning methods to fit the social forces model to observed crowd behavior [68, 77]. Although the social forces model seems to perform well at simulating crowds, we found that the model has shortcomings in predicting the movements of individual pedestrians, particularly during evasive maneuvers. In contrast to that, Pandey and Alami [141] and Kirby et al. [85] implement predefined social rules on a robot that operates in an environment populated by pedestrians. Similarly, Müller et al. [133] present a method that enables a mobile robot to detect and follow people that happen to walk in the same direction as the robot. Lerner



(a) $t = 0.0\,\mathrm{s}$ **(b)** $t = 1.6\,\mathrm{s}$

(c) $t = 2.0\,\mathrm{s}$ **(d)** $t = 2.6\,\mathrm{s}$

(e) $t = 3.2\,\mathrm{s}$ **(f)** $t = 4.0\,\mathrm{s}$

Figure 7.22: A robot encounters two pedestrians in a hall way. The robot imitates a cautious, cooperative navigation behavior that it has observed earlier. The robot shown in red and pedestrians shown in blue mutually evade each other.

et al. [115] present a data-driven approach to modeling the behavior of crowds, where agents behave according to examples that are stored in a database of previously recorded crowd movements.

In contrast to the above-mentioned steering models, optimization models cast pedestrians as utility-optimizing agents that minimize a cost function comprising relevant properties of human navigation [10, 71, 131, 144]. There are methods that minimize the walking discomfort in terms of accelerations and distances to other pedestrians [71], maximize smoothness of the trajectory [144], or minimize the derivative of the curvature [10]. Mombaur et al. [131] present an optimization model that allows a humanoid robot to imitate human-like trajectories. Some of the above-mentioned optimization approaches adapt the parameters of the models such that the resulting trajectories resemble training examples, which is known as inverse optimal control, a problem originally posed by Kálmán [111]. In contrast to these non-probabilistic methods, we capture the stochasticity of human navigation behavior by modeling the behavior in terms of a probability distribution over the agents' trajectories. This also allows our method to learn from non-optimal demonstrations.

Trautman and Krause [165] demonstrate that a mobile robot is prone to get stuck

in environments that are densely populated by pedestrians unless the robot engages in cooperative collision avoidance with the pedestrians. As opposed to modeling the agents independently, Trautman and Krause therefore propose to jointly reason about the trajectories of the robot and the pedestrians to account for interactions between the agents. Likewise, our approach reasons about composite trajectories of the agents to capture interactions between them. Similarly, van den Berg et al. [167] present an efficient approach to reciprocal collision avoidance that allows a set of mobile robots to navigate without collisions. The social forces model presented by Helbing and Molnár [69] as well as the data-driven approach presented by Lerner et al. [115] implicitly model cooperative navigation behavior. In contrast to that, we explicitly model cooperative navigation behavior in terms of a joint probability distribution over the trajectories of all the agents.

Some authors rely on cluster analysis to learn common motion patterns from a set of demonstrated trajectories. For instance, Bennewitz et al. [19] learn common motion patterns of pedestrians by means of EM clustering of demonstrated trajectories. Likewise, Bruce and Gordon [25] learn to predict the target locations of pedestrians by clustering demonstrations of trajectories.

Methods like the dynamic window approach suggested by Fox et al. [53], the velocity obstacles proposed by Fiorini and Shillert [49], and the reciprocal velocity obstacles presented by van den Berg et al. [167] have been applied to mobile robot navigation in crowded or dynamic environments. These methods enable a mobile robot to successfully avoid collisions with dynamic obstacles such as pedestrians. They, however, do not account for human predictive navigation abilities, which sometimes results in unnatural robot movements.

In the context of mobile robot navigation, popular algorithms such as rapidly-exploring random trees [113] and probabilistic roadmaps [83] rely on a set of random samples to construct a graph whose vertices, which correspond to poses in the environment, are typically connected by a local path planner [112]. Many authors optimize trajectories with respect to a given cost function. For instance, Sprunk et al. [155] use quintic Bézier splines to represent the trajectory of the robot and optimize control points in order to obtain time-optimized, curvature continuous trajectories. Ratliff et al. [148] present a general framework for trajectory optimization, which can be applied to high-dimensional mobile robot motion planning. These gradient-based optimization methods, however, are prone to get stuck in local minima. Kalakrishnan et al. [81] therefore propose to use stochastic trajectory optimization to overcome these local minima. However, large state spaces as a result of complex settings make it infeasible to efficiently find globally optimal solutions by uniformly sampling trajectories. In contrast to that, our model explores the state space by simultaneously searching regions of different homotopy classes, which each corresponds to a local minimum.

7.8 Conclusion

We presented a novel approach that allows a mobile robot to learn a model of the navigation behavior of cooperatively navigating agents such as pedestrians. Based on

observations of their continuous trajectories, our method infers a model of the underlying decision process. To cope with the discrete and continuous aspects of this process, our model uses a mixture distribution that captures the discrete decisions regarding the homotopy classes of the composite trajectories as well as continuous properties of the trajectories such as higher-order dynamics. To compute the feature expectations with respect to the continuous, high-dimensional probability distributions, our method uses Hamiltonian Markov chain Monte Carlo sampling. The learned models enable socially compliant mobile robot navigation by allowing the robot to predict the navigation behavior of pedestrians in terms of a probability distribution over their trajectories. A cross-validation demonstrates that our method generalizes to new situations and outperforms three state-of-the-art techniques. Furthermore, a Turing test suggests that the pedestrian trajectories induced by our approach appear highly human-like.

Chapter 8

Discussion

Robots need accurate models of the environment in order to successfully carry out their tasks autonomously. Such models comprise maps of the robot's environment for localization and path planning and models of the behavior of humans for socially compliant human-robot interaction. In many applications, however, these models cannot be provided by human experts in all necessary detail. In this thesis, we therefore presented several novel techniques that enable a mobile robot to learn such models from its own sensor measurements. The remainder of this chapter summarizes the main contributions of this thesis and identifies some interesting avenues of future work.

Maps of the environment are needed for a wide range of mobile robot applications including domestic service robots and delivery robots. State-of-the-art approaches to simultaneous localization and mapping store all the sensor measurements that the robot obtains in memory. This can exhaust the memory of the robot during long-term operation. We presented a novel approach that allows a mobile robot to discard laser measurements in order to free memory during navigation. Our probabilistic approach to mapping demands that we consider the mutual information between the laser measurements and the occupancy grid map in order to find measurements that provide little information about the environment. This allows us to discard laser measurements in a way that minimizes the loss of information about the environment. However, computing the expected entropy of the occupancy grid map requires solving an integral, which is infeasible in general. We demonstrated how to overcome this by expressing the entropy of the grid map in terms of the effects of the laser measurements on the grid cells. We demonstrated in empirical experiments that the memory consumption of our method does not grow without bound while the robot obtains new measurements of previously explored areas of the environment, which is in contrast to most state-of-the-art methods.

Solving the data association problem in simultaneous localization and mapping can be challenging, as the number of possible associations scales exponentially with the number of feature observations. As a result, especially in large, ambiguous environments, data association can become intractable, even when using state-of-the-art techniques. We discussed that mobile robots can resolve ambiguities in the environment by deploying uniquely identifiable artificial landmarks in the environment during operation. We then presented an approach that allows a mobile robot to learn a landmark deployment policy that maximizes the utility of the deployed landmarks regarding data association. To this end, our method uses actor-critic Monte Carlo reinforcement learning to compute a policy

that depends on features of the environment. We demonstrated that the policies learned by our approach outperform, in terms of data association performance, multiple baseline policies, such as a policy that deploys the landmarks equidistantly in time and a policy that deploys the landmarks at random points in time. Moreover, we demonstrated that our method outperforms a state-of-the-art technique to deploy landmarks.

The approaches to simultaneous localization and mapping presented in this thesis assume that the environment of the robot does not change during its mission, which, however, is typically not the case in practice. It would therefore be interesting to investigate methods for properly dealing with changing environments.

In the context of socially compliant mobile robots, we considered the problem of giving directions to people unfamiliar with the environment. We formulated the problem of describing routes in terms of a Markov decision process. We then used maximum entropy inverse reinforcement learning to compute a reward function from a set of descriptions given by humans. Since the state space and the action space of the Markov decision process are both discrete in this case, we can take advantage of a highly efficient dynamic programming technique to compute the reward function, which is commonly used in the context of maximum entropy inverse reinforcement learning. Computing the reward function is of interest because it encodes a particular style of giving directions. Since the reward function depends on features of the descriptions, it generalizes to new environments. We demonstrated that our method allows a mobile robot to imitate the style of a group of humans in its target environment. Participants of a user study rated the descriptions generated by our approach as significantly more human-like than those produced by a state-of-the-art method. Our current implementation, however, relies on manual work in the process. More specifically, formulating the Markov decision process requires a predefined set of locations in the environment. These states in the Markov decision process correspond to decision points along the route. In addition to that, a predefined set of instructions that later form the entire description needs to be provided manually. It would be interesting to leverage techniques from the field of natural language processing in order to automatically infer the decision points and common instructions from the corpus of directions given by humans.

We furthermore addressed the problem of human motion capture in order to obtain training data to learn models of the navigation behavior of pedestrians. We compensated for the drift in inertial motion capture systems by using a mobile robot that is equipped with a laser range finder. Our method allows the robot to anchor the pose estimate of a person in a map of the environment. We demonstrated how to robustly handle false-positive and false-negative laser observations of the person in the particle filter, which enables highly robust tracking even in populated environments. We then formulated a least squares optimization problem based on the inertial motion capture data and all the laser measurements that the robot obtained in order to further refine the trajectory estimate. As a result, our method leads to highly accurate estimates of the trajectories of pedestrians, which the robot can use to learn accurate models of their behavior. Our experiments, however, require that the tracked person wear a high-priced inertial motion capture suit. Given the prevalent use of modern mobile phones, it would be interesting

to explore the use of data from inertial sensors built into these devices. Developing an adequate motion model and a robust sensor model for such a device in the particle filter framework would be a first exciting step.

As a main contribution of this thesis, we presented a probabilistic framework to model the cooperative navigation behavior of pedestrians. To capture cooperative behavior, our model jointly reasons about the trajectories of pedestrians nearby the robot in terms of features of the trajectories. We use a mixture distribution to model discrete navigation decisions, such as whether to pass each other on the left or on the right side, as well as continuous properties of the trajectories, such as velocities and accelerations. A key challenge when learning the parameters of the model based on training data is to compute the expected values of the features that are induced by the current parameters. This requires solving an integral, which is infeasible in closed form. A common approach is to resort to approximations of these expectations by means of Markov chain Monte Carlo sampling. These sampling techniques, however, tend to be rather inefficient in practice. Our mixture representation of the probability distribution yet enables efficient sampling by exploiting the gradient of the density in order to guide the sampling process towards regions of high probability. We demonstrated that our approach outperforms two state-of-the-art techniques in a cross-validation, namely the social forces algorithm and the reciprocal velocity obstacles method. Participants of a Turing test mistook the trajectories computed by our method for being those of humans significantly more often than the trajectories computed by a state-of-the-art model. The number of homotopy classes, however, scales exponentially with the number of pedestrians that are to be considered. We discussed that our current implementation of the model relies on a set of heuristics to mitigate the computational burden that arises as the number of pedestrians increases. It would, however, be interesting to explore the space of homotopy classes in a more principled way. In this thesis, we used our probabilistic framework to model the behavior of pedestrians. Modeling other modes of transportation would lead to fascinating new research. For instance, it would be exciting to model the behavior of self-driving cars.

The contribution of this thesis is a set of novel techniques that allows a mobile robot to learn probabilistic models of its environment from its own sensor measurements. These models comprise maps of the robot's environment for navigation and models of the behavior of humans for socially compliant human-robot interaction. We implemented and evaluated all the presented methods, when applicable on real mobile robots, and demonstrated that they outperform the state of the art in robotics. The techniques presented in this thesis therefore provide useful insights into developing mobile robots that autonomously collaborate with humans in a socially compliant way even in previously unknown environments.

List of Figures

Bibliography

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, Banff, Canada, 2004.

[2] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research (IJRR)*, 29(13):1608–1639, 2010.

[3] Nichola Abdo, Henrik Kretzschmar, and Cyrill Stachniss. From low-level trajectory demonstrations to symbolic actions for planning. In *Proceedings of the ICAPS Workshop on Combining Task and Motion Planning for Real-World Applications (TAMPRA)*, Atibaia, São Paulo, Brazil, 2012.

[4] Nichola Abdo, Henrik Kretzschmar, Luciano Spinello, and Cyrill Stachniss. Learning manipulation actions from a few demonstrations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[5] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[6] Gary L. Allen. From knowledge to words to wayfinding: Issues in the production and comprehension of route directions. In *Spatial Information Theory A Theoretical Basis for GIS*, volume 1329 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[7] Motion Analysis. http://www.motionanalysis.com.

[8] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.

[9] AOL. MapQuest. http://www.mapquest.com/.

[10] Gustavo Arechavaleta, Jean-Paul Laumond, Halim Hicheur, and Alain Berthoz. An optimality principle governing human walking. *IEEE Transactions on Robotics (T-RO)*, 24(1):5–14, 2008.

[11] Kai O. Arras, Slawomir Grzonka, and Matthias Luber. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008.

[12] Christopher Atkeson and Stefan Schaal. Robot learning from demonstration. In *Proceedings of the International Conference on Machine Learning (ICML)*, Nashville, TN, USA, 1997.

[13] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. RANGE - robust autonomous navigation in GPS-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.

[14] Yaakov Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., 1987.

[15] Maxim A. Batalin and Gaurav S. Sukhatme. Efficient exploration without localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003.

[16] Maximilian Beinhofer, Henrik Kretzschmar, and Wolfram Burgard. Deploying artificial landmarks to foster data association in simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[17] Michael A. Bender, Antonio Fernãdez, Dana Ron, Amit Sahai, and Salil Vadhan. The power of a pebble: Exploring and mapping directed graphs. *Information and Computation*, 176(1):1 – 21, 2002.

[18] Maren Bennewitz. *Mobile Robot Navigation in Dynamic Environments*. PhD thesis, University of Freiburg, Department of Computer Science, 2004.

[19] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *International Journal of Robotics Research (IJRR)*, 24(1):31–48, 2005.

[20] Maren Bennewitz, Felix Faber, Dominik Joho, and Sven Behnke. Fritz – A humanoid communication robot. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Jeju Island, Korea, 2007.

[21] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[22] Stephen Bitgood and Stephany Dukes. Not another step! economy of movement and pedestrian choice point behavior in shopping malls. *Environment and Behavior*, 38(3):394–405, 2006.

[23] Michael Bosse, Paul Newman, John Leonard, Martin Soika, Wendelin Feiten, and Seth Teller. An Atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003.

[24] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. Lauderdale, FL, USA, 2011.

[25] Allison Bruce and Geoffrey Gordon. Better motion prediction for people-tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2004.

[26] Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti, Bastian Steder, Rainer Kümmerle, Christian Dornhege, Michael Ruhnke, Alexander Kleiner, and Juan D. Tardós. A comparison of SLAM algorithms based on a graph of relations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.

[27] Gary Burnett. 'Turn right at the traffic lights': The requirement for landmarks in vehicle navigation systems. *Journal of Navigation*, 53(03):499–510, 2000.

[28] Zachary Byers, Michael Dixon, Kevin Goodier, Cindy M. Grimm, and William D. Smart. An autonomous robot photographer. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2003.

[29] Nicholas Carlevaris-Bianco and Ryan M. Eustice. Generic factor-based node marginalization and edge sparsification for pose-graph slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[30] Nicholas Carlevaris-Bianco and Ryan M. Eustice. Long-term simultaneous localization and mapping with generic linear constraint node removal. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

[31] Nicholas Carlevaris-Bianco and Ryan M. Eustice. Conservative edge sparsification for graph slam node removal. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[32] Margarita Chli and Andrew J. Davison. Automatically and efficiently inferring the hierarchical structure of visual maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[33] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[34] Henrik Christensen and Elena Pacchierotti. Embodied social interaction for robots. In *Proceedings of the Symposium on Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, Hatfield, UK, 2005.

[35] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[36] Heriberto Cuayáhuitl, Nina Dethlefs, Lutz Frommberger, Kai-Florian Richter, and John Bateman. Generating adaptive route instructions using hierarchical reinforcement learning. In *Proceedings of Spacial Cognition*, Portland, OR, USA, 2010.

[37] Jinshi Cui, Hongbin Zha, Huijing Zhao, and Ryosuke Shibasaki. Tracking multiple people using laser and vision. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, 2005.

[38] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research (IJRR)*, 27(6):647–665, 2008.

[39] Robert Dale, Sabine Geldof, and Jean-Philippe Prost. Using natural language generation in automatic route description. *Journal of Research and Practice in Information Technology*, 36(3):23–39, 2005.

[40] Andrew J. Davison. Active search for real-time vision. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, Beijing, China, 2005.

[41] Michel Denis, Francesca Pazzaglia, Cesare Cornoldi, and Laura Bertolo. Spatial discourse and navigation: An analysis of route directions in the city of venice. *Applied Cognitive Psychology*, 13(2):145–174, 1999.

[42] Simon Duane, Anthony D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

[43] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes. Map validation and robot self-location in a graph-like world. *Robotics and Autonomous Systems (RAS)*, 22(2):159 – 178, 1997.

[44] Krishnamurthy Dvijotham and Emanuel Todorov. Inverse optimal control with linearly-solvable MDPs. In *Proceedings of the International Conference on Machine Learning (ICML)*, Haifa, Israel, 2010.

[45] Ethan Eade, Philip Fong, and Mario E. Munich. Monocular graph SLAM with complexity reduction. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

[46] Aandreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Moving obstacle detection in highly dynamic scenes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[47] Carlos Estrada, José Neira, and Juan D. Tardós. Hierachical SLAM: Real-time accurate mapping of large environments. *IEEE Transactions on Robotics (T-RO)*, 21(4):588–596, 2005.

[48] Ryan M. Eustice, Hanumant Singh, and John J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics (T-RO)*, 22(6): 1100–1114, 2006.

[49] Paolo Fiorini and Zvi Shillert. Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research (IJRR)*, 17:760–772, 1998.

[50] Ajo Fod, Andrew Howard, and Maja Matarić. Laser-based people tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, USA, 2002.

[51] John Folkesson, Patric Jensfelt, and Henrik I. Christensen. Vision SLAM in the measurement subspace. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 2005.

[52] Charles Fox. *An Introduction to the Calculus of Variations*. Courier Dover Publications, Mineola, NY, USA, 1987.

[53] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine (RAM)*, 4 (1):23–33, 1997.

[54] Stephen Friedman, Hanna Pasula, and Dieter Fox. Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In *Proceedings of the International Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

[55] Luis M. Fuentes and Sergio A. Velastin. People tracking in surveillance applications. *Image and Vision Computing*, pages 1165–1171, 2006.

[56] Robert Goeddel and Edwin Olson. DART: A particle-based method for generating easy-to-follow directions. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Algarve, Portugal, 2012.

[57] Barbara Gonsior, Christian Landsiedel, Antonia Glaser, Dirk Wollherr, and Martin Buss. Dialog strategies for handling miscommunication in task-related hri. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2011.

[58] Barbara Gonsior, Christian Landsiedel, Nicole Mirnig, Stefan Sosnowski, Ewald Strasser, Jakub Zlotowski, Martin Buss, Kolja Kühnlenz, Manfred Tscheligi, Astrid Weiss, et al. Impacts of multimodal feedback on efficiency of proactive information retrieval from task-related hri. *JACIII*, 16(2):313–326, 2012.

[59] Google Inc. Google Maps. https://maps.google.com.

[60] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2d and 3d mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, 2010.

[61] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics (T-RO)*, 23(1):34–46, 2007.

[62] Slawomir Grzonka, Frederic Dijoux, Andreas Karwath, and Wolfram Burgard. Mapping indoor environments based on human activity. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010.

[63] Stephen J. Guy, Jur van den Berg, Wenxi Liu, Rynson Lau, Ming C. Lin, and Dinesh Manocha. A statistical similarity measure for aggregate crowd dynamics. *ACM Transactions on Graphics (TOG)*, 31(6):190, 2012.

[64] Edward T. Hall. *The Hidden Dimension*. Doubleday, New York, 1966.

[65] Shazia Haque, Lars Kulik, and Alexander Klippel. Algorithms for reliable navigation and wayfinding. In *Proceedings of Spacial Cognition*, volume 4387, Bremen, Germany, 2007.

[66] W.K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[67] Ruijie He, Samuel Prentice, and Nicholas Roy. Planning in information space for a quadrotor helicopter in a gps-denied environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008.

[68] Dirk Helbing and Anders Johansson. Pedestrian, crowd and evacuation dynamics. In *Encyclopedia of Complexity and Systems Science*, pages 6476–6495. Springer New York, 2009.

[69] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Physical Review E (PRE)*, 51:4282–4286, 1995.

[70] Peter Henry, Christian Vollmer, Brian Ferris, and Dieter Fox. Learning to navigate through crowded environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010.

[71] Serge Hoogendoorn and Piet H.L. Bovy. Simulation of pedestrian flows by optimal control and differential games. *Optimal Control Applications and Methods*, 24(3): 153–172, 2003.

[72] Andrew Howard and Nicholas Roy. The robotics data set repository (Radish), 2003. http://radish.sourceforge.net/.

[73] Alycia M. Hund, Kimberly H. Haney, and Brian D. Seanor. The role of recipient perspective in giving and following wayfinding directions. *Applied Cognitive Psychology*, 22(7):896–916, 2008.

[74] Alycia M. Hund, Martin Schmettow, and Matthijs L. Noordzij. The impact of culture and recipient perspective on direction giving in the service of wayfinding. *Journal of Environmental Psychology*, 32(4):327–336, 2012.

[75] Viorela Ila, Josep M. Porta, and Juan Andrade-Cetto. Information-based compact pose slam. *IEEE Transactions on Robotics (T-RO)*, 26(1):78–93, 2010.

[76] Edwin T. Jaynes. Where do we stand on maximum entropy. *The Maximum Entropy Formalism*, pages 15–118, 1978.

[77] Anders Johansson, Dirk Helbing, and Pradyumn K. Shukla. Specification of the social force pedestrian model by evolutionary adjustment to video tracking data. *Advances in Complex Systems (ACS)*, 10:271–288, 2007.

[78] Michael Kaess and Frank Dellaert. Covariance recovery from a square root information matrix for data association. *Journal of Robotics and Autonomous Systems (RAS)*, 57:1198–1210, 2009.

[79] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics (T-RO)*, 24(6):1365–1378, 2008.

[80] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the bayes tree. *International Journal of Robotics Research (IJRR)*, 31(2):217–236, 2012.

[81] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[82] Mrinal Kalakrishnan, Peter Pastor, Ludovic Righetti, and Stefan Schaal. Learning objective functions for manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[83] Lydia E. Kavraki, Petr Švestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[84] Ayoung Kim and Ryan M. Eustice. Combined visually and geometrically informative link hypothesis for pose-graph visual SLAM using bag-of-words. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, 2011.

[85] Rachel Kirby, Reid Simmons, and Jodi Forlizzi. COMPANION: A constraint optimizing method for person-acceptable navigation. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Toyama, Japan, 2009.

[86] Kris M. Kitani, Brian D. Ziebart, J. Andrew Bagnell, and Martial Hebert. Activity forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Firenze, Italy, 2012.

[87] Marcus Kleinehagenbrock, Sebastian Lang, Jannik Fritsch, Frank Lömker, Gernot A. Fink, and Gerhard Sagerer. Person tracking with a mobile robot based on multi-modal anchoring. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Berlin, Germany, 2002.

[88] Alexander Kleiner, Johann Prediger, and Bernhard Nebel. RFID technology-based exploration and SLAM for search and rescue. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, 2006.

[89] Boris Kluge, Christian Kohler, and Erwin Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, 2001.

[90] Marin Kobilarov, Gaurav Sukhatme, Jeff Hyams, and Parag Batavia. People tracking and following with mobile robot using an omnidirectional camera and a laser. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, FL, USA, 2006.

[91] Thomas Kollar. *Learning to Understand Spatial Language for Robotic Navigation and Mobile Manipulation*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.

[92] Thomas Kollar and Nicholas Roy. Efficient optimization of information-theoretic exploration in SLAM. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Chicago, IL, USA, 2008.

[93] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Osaka, Japan, 2010.

[94] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Grounding verbs of motion in natural language commands to robots. *STAR Springer Tracts in Advanced Robotics*, 79:31–47, 2014.

[95] Kurt Konolige and Motilal Agrawal. FrameSLAM: From bundle adjustment to realtime visual mappping. *IEEE Transactions on Robotics (T-RO)*, 24(5):1066–1077, 2008.

[96] Kurt Konolige and James Bowman. Towards lifelong visual maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.

[97] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, Edinburgh, Scotland, 2005.

[98] Henrik Kretzschmar and Cyrill Stachniss. Information-theoretic compression of pose graphs for laser-based SLAM. *The International Journal of Robotics Research (IJRR)*, 31:1219–1230, 2012.

[99] Henrik Kretzschmar, Cyrill Stachniss, Christian Plagemann, and Wolfram Burgard. Estimating landmark locations from geo-referenced photographs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008.

[100] Henrik Kretzschmar, Giorgio Grisetti, and Cyrill Stachniss. Lifelong map learning for graph-based SLAM in static environments. *KI – Künstliche Intelligenz*, 24: 199–206, 2010.

[101] Henrik Kretzschmar, Cyrill Stachniss, and Giorgio Grisetti. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, 2011.

[102] Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Inferring navigation policies for mobile robots from demonstrations. In *Proceedings of the Autonomous Learning Workshop at the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[103] Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Predicting human navigation behavior via inverse reinforcement learning. In *The 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM)*, Princeton, NJ, USA, 2013.

[104] Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Learning navigation policies from human demonstrations. In *Proceedings of the Workshop on Inverse Optimal Control & Robotic Learning from Demonstration at Robotics: Science and Systems (RSS)*, Berlin, Germany, 2013.

[105] Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard. Learning to predict trajectories of cooperatively navigating agents. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[106] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *Proceedings of Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.

[107] Markus Kuderer, Henrik Kretzschmar, and Wolfram Burgard. Teaching mobile robots to cooperatively navigate in populated environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

[108] Markus Kuderer, Christoph Sprunk, Henrik Kretzschmar, and Wolfram Burgard. Online generation of homotopically distinct navigation paths. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.

[109] Barbara Kühnlenz, Stefan Sosnowski, Malte Buß, Dirk Wollherr, Kolja Kühnlenz, and Martin Buss. Increasing helpfulness towards a robot by emotional adaption to the user. *International Journal of Social Robotics*, 5(4):457–476, 2013.

[110] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[111] Rudolf E. Kálmán. When is a linear control system optimal? *Journal of Basic Engineering*, 86(1):51–60, 1964.

[112] Steven M. LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[113] Steven M. Lavalle and James J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, Hanover, NH, USA, 2000.

[114] Scott Lenser and Manuela Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, USA, 2000.

[115] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[116] Sergey Levine and Vladlen Koltun. Continuous inverse optimal control with locally optimal examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012.

[117] Jun S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[118] Gary Wai Keung Look. *Cognitively-inspired direction giving*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008.

[119] Kristin L. Lovelace, Mary Hegarty, and Daniel R. Montello. Elements of good route directions in familiar and unfamiliar environments. In *Proceedings of the International Conference on Spatial Information Theory (COSIT)*, London, UK, 1999.

[120] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[121] Matthias Luber, Johannes Andreas Stork, Gian Diego Tipaldi, and Kai Oliver Arras. People tracking with human motion predictions from social forces. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010.

[122] Matthias Luber, Gian Diego Tipaldi, and Kai O. Arras. Place-dependent people tracking. *International Journal of Robotics Research (IJRR)*, 2011.

[123] Jie Luo, Andrzej Pronobis, Barbara Caputo, and Patric Jensfelt. Incremental learning for place recognition in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.

[124] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[125] Matthew T. MacMahon. *Following Natural Language Route Instructions*. PhD thesis, Electrical and Computer Engineering Department, University of Texas at Austin, 2007.

[126] David M. Mark. Automated route selection for navigation. *Aerospace and Electronic Systems Magazine, IEEE*, 1(9):2–5, 1986.

[127] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2): 431–441, 1963.

[128] Óscar Martínez Mozos, Rudolph Triebel, Patric Jensfelt, Axel Rottmann, and Wolfram Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems (RAS)*, 55(5): 391–402, 2007.

[129] Mladen Mazuran, Tipaldi Gian Diego, Spinello Luciano, and Wolfram Burgard. Nonlinear graph sparsification for slam. In *Proceedings of Robotics: Science and Systems*, Berkeley, CA, USA, 2014.

[130] Microsoft. Bing Maps. http://www.bing.com/maps.

[131] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion – an inverse optimal control approach. *Autonomous Robots*, 28:369–383, 2010.

[132] Michael Montemerlo, Sebastian Thrun, and William Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, USA, 2002.

[133] Jörg Müller, Cyrill Stachniss, Kai O. Arras, and Wolfram Burgard. Socially inspired motion planning for mobile robots in populated environments. In *Proceedings of the International Conference on Cognitive Systems (COGSYS)*, Karlsruhe, Germany, 2008.

[134] José Neira and Juan D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6): 890–897, 2001.

[135] Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, Stanford, CA, USA, 2000.

[136] Kai Ni, Drew Steedly, and Frank Dellaert. Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007.

[137] Lauro Ojeda and Johann Borenstein. Non-GPS navigation for security personnel and first responders. *Journal of Navigation*, 60(3):391–407, 2007.

[138] Edwin Olson. Recognizing places using spectrally clustered local matches. 2009.

[139] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *International Journal of Robotics Research (IJRR)*, 32(7):826–840, 2013.

[140] Stefan Oßwald, Henrik Kretzschmar, Wolfram Burgard, and Cyrill Stachniss. Learning to give route directions from human demonstrations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014. **Best Cognitive Robotics Paper - Finalist**.

[141] Amit K. Pandey and Rachid Alami. A framework for adapting social conventions in a mobile robot motion in human-centered environment. In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, Munich, Germany, 2009.

[142] Nuria Pelechano, Jan Allbeck, and Norman I. Badler. Controlling individual agents in high-density crowd simulation. In *Proceedings of the Symposium on Computer Animation*, San Diego, CA, USA, 2007.

[143] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[144] Quang-Cuong Pham, Halim Hicheur, Gustavo Arechavaleta, Jean-Paul Laumond, and Alain Berthoz. The formation of trajectories during goal-oriented locomotion in humans. II. A maximum smoothness model. *European Journal of Neuroscience*, 26:2391–2403, 2007.

[145] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. Numerical recipes, 1990.

[146] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[147] Nathan Ratliff, J. Andrew Bagnell, and Martin Zinkevich. Maximum margin planning. In *Proceedings of the International Conference on Machine Learning (ICML)*, Pittsburgh, PA, USA, 2006.

[148] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[149] Kai-Florian Richter. *Context-specific route directions: generation of cognitively motivated wayfinding instructions*. Dissertationen zur künstlichen Intelligenz. Akademische Verlagsgesellschaft (Aka), 2007.

[150] Kai-Florian Richter and Matt Duckham. Simplest instructions: Finding easy-to-describe routes for navigation. In *Proceedings of the International Conference on Geographic Information Science (GIScience)*, Park City, UT, USA, 2008.

[151] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, San Francisco, CA, USA, 1993.

[152] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, Korea, 2001.

[153] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. *International Journal of Robotics Research (IJRR)*, 22(2):99–116, 2003.

[154] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Skeletal graphs for efficient structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, USA, 2008.

[155] Christoph Sprunk, Boris Lau, Patrick Pfaff, and Wolfram Burgard. Online generation of kinodynamic trajectories for non-circular omnidirectional robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[156] Cyrill Stachniss and Henrik Kretzschmar. Pose graph compression for laser-based SLAM. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, Flagstaff, AZ, USA, 2011. **Invited presentation**.

[157] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005.

[158] Hauke Strasdat, Cyrill Stachniss, and Wolfram Burgard. Which landmark is useful? Learning selection policies for navigation in unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[159] Avneesh Sud, Russell Gayle, Erik Andersen, Stephen Guy, Ming Lin, and Dinesh Manocha. Real-time navigation of independent agents using adaptive roadmaps. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, Newport Beach, CA, USA, 2007.

[160] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[161] Vicon Motion Systems. http://www.vicon.com.

[162] Sebastian Thrun. Finding landmarks for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998.

[163] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[164] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

[165] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 2010.

[166] Alan M. Turing. Computing machinery and intelligence. *Mind*, pages 433–460, 1950.

[167] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Proceedings of the International Symposium of Robotics Research (ISRR)*, Lucerne, Switzerland, 2009.

[168] Paul Vernaza and J. Andrew Bagnell. Efficient high dimensional maximum entropy modeling via symmetric partition functions. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 584–592. 2012.

[169] David Waller and Yvonne Lippa. Landmarks as beacons and associative cues: Their role in route learning. *Memory & Cognition*, 35(5), 2007.

[170] Hui Wang, Michael Jenkin, and Patrick Dymond. The relative power of immovable markers in topological mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[171] Shawn L. Ward, Nora Newcombe, and Willis F. Overton. Turn left at the church, or three miles north: A study of direction giving and sex differences. *Environment and Behavior*, 18(2):192–213, 1986.

[172] William H. Warren. The dynamics of perception and action. *Psychological Review*, 113:358–389, 2006.

[173] Torsten Wilhelm, Hans-Joachim Böhme, and Horst-Michael Gross. Sensor fusion for vision and sonar based people tracking on a mobile service robot. In *Proceedings of the International Workshop on Dynamic Perception*, Bochum, Germany, 2002.

[174] Kai M. Wurm, Henrik Kretzschmar, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. Identifying vegetation from laser data in structured outdoor environments. *Robotics and Autonomous Systems*, 62:675–684, 2014.

[175] Xsens. MVN - Inertial Motion Capture. http://www.xsens.com.

[176] Mitsumasa Yoda and Yasuhito Shiota. Analysis of human avoidance motion for application to robot. In *Proceedings of the 5th IEEE International Workshop on Robot and Human Communication*, 1996.

[177] Mitsumasa Yoda and Yasuhito Shiota. The mobile robot which passes a man. In *Proceedings of the 6th IEEE International Workshop on Robot and Human Communication*, 1997.

[178] Davide Zambrano, Delphine Bernardin, Daniel Bennequin, Cecilia Laschi, and Alain Berthoz. A comparison of human trajectory planning models for implementation on humanoid robot. In *In Proceedings of the 4th IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012.

[179] Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Machine Learning Department, Carnegie Mellon University, 2010.

[180] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Chicago, IL, USA, 2008.

[181] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Deyvivien. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp)*, Seoul, Korea, 2008.

[182] Brian D. Ziebart, Nathan Ratliff, Garrat Gallagher, Christoph Mertz, Kevin Peterson, J. Andrew Bagnell, Martial Hebert, Anind K. Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.

[183] Brian D. Ziebart, Anind K. Dey, and J. Andrew Bagnell. Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of the ACM International conference on Intelligent User Interfaces (IUI)*, Lisbon, Portugal, 2012.

[184] Jakob Ziegler, Henrik Kretzschmar, Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Accurate human motion capture in large areas by combining IMU- and laser-based people tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, 2011.

[185] Zoran Zivkovic and Ben Kröse. Part based people detection using 2D range data and images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.